**RĪGAS TEHNISKĀ UNIVERSITĀTE**

Datorzinātnes un informācijas tehnoloģijas fakultāte

Lietišķo datorsistēmu institūts

**Domeniks Delvers**

bakalaura akadēmisko, pirmā līmeņa studiju programmas

students, stud. apl. nr. 201RDB298

**VAIRĀKU AĢENTU**

**VIRZĪTA APMĀCĪBA**

**PĀRKLĀJUMA CEĻA MEKLĒŠANAI**

**BAKALAURA DARBS**

Zinātniskais vadītājs Doktors, Profesors

Agris Ņikitenko

RĪGA 2023

**RĪGAS TEHNISKĀ UNIVERSITĀTE**

**DATORZINĀTNES UN INFORMĀCIJAS TEHNOLOĢIJAS FAKULTĀTE**

<institūta nosaukums>

<struktūrvienības nosaukums>

**<Noslēguma darba no saraksta: bakalaura darbs, maģistra darbs, diplomprojekts, kvalifikācijas darbs> izpildes lapa**

Noslēguma darba autors:

students(-e) <vārds, uzvārds>\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(paraksts, datums)

Noslēguma darbs ieteikts aizstāvēšanai:

Zinātniskais vadītājs:

<zinātniskais grāds, amats, vārds, uzvārds> \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(paraksts, datums)

Anotācija

Pārklājuma ceļa plānošana, virzīta apmācība, vairāku aģentu sistēmas, procedurāla ģenerācija,

Mūsdienās mājas tīrīšanas roboti paliek arvien izplatītāki. Pašlaik tirgū esošie risinājumi nepiedāvā vairāku aģentu funkcionalitāti, tomēr tādas sistēmas ir vieglāk mērogojamas, izturīgākas kā arī ļauj ātrāk pārklāt lielākas telpas. Viena no pieejām kā īstenot vairāku aģentu pārklājuma ceļa plānošanas algoritmu ir virzītas apmācības metode. Vairāku aģentu virzītā apmācība pārklājuma ceļa meklēšanai ir apskatīta šajā bakalaura darbā.

Darbā ir X lappušu, Y attēlu, Z tabulu, W pielikumu un Q izmantotie avoti.

Abstract

Coverage path planning, reinforcement learning, multi – agent systems, procedural generation.

Nowadays, home cleaning robots are becoming more and more popular. Solutions that are currently available in the market are lacking multi – agent functionality, however such systems are easier to scale, more robust, as well as they allow for faster coverage of huge environments. One of the methods of achieving multi – agent coverage path planning is by using reinforcement learning. Multi – agent reinforcement learning for coverage path planning is overlooked in this thesis paper.

Paper contains X pages, Y images, Z tables, W additions and Q references.

Satura rādītājs

[Anotācija 2](#_Toc136105783)

[Abstract 3](#_Toc136105784)

[Satura rādītājs 4](#_Toc136105785)

[Ievads 5](#_Toc136105786)

[1. ProBlēmas Nostādne 7](#_Toc136105787)

[1.1. Vairāku aģentu pārklājuma ceļa plānošanas formalizēšana iekštelpu vidēs 7](#_Toc136105788)

[1.1.1. Aģenta un vides modelis 7](#_Toc136105789)

[1.1.2. Mērķi un ierobežojumi 7](#_Toc136105790)

[1.2. Izaicinājumi vairāku aģentu pārklājuma ceļa plānošanā 8](#_Toc136105791)

[2. SAISTĪTIE PĒTīJUMI 9](#_Toc136105792)

[2.1. Tradicionālas pārklājuma meklēšanas tehnikas 9](#_Toc136105793)

[2.2. Virzītā apmācība pārklājuma ceļa plānošanai 10](#_Toc136105794)

[3. Proksimālās Politikas Optimizācijas pielietojums pārklājuma ceļa meklēšanā vairāku aģentu scenārijos 13](#_Toc136105795)

[3.1. PPO pielietojamība vairāku aģentu pārklājuma ceļa meklēšanai 13](#_Toc136105796)

[3.2. PPO pārskats 14](#_Toc136105797)

[3.3. PPO vairāku aģentu sistēmā 15](#_Toc136105798)

[4. Iekštelpas vides procedurāla ģenerēšana 17](#_Toc136105799)

[4.1. Vides skeleta veidošana 17](#_Toc136105800)

[4.2. Durvju izvietojums 18](#_Toc136105801)

[4.3. Mēbeļu un šķēršļu ģenerēšanā 19](#_Toc136105802)

[5. Aģents 21](#_Toc136105803)

[Atsauksmes 23](#_Toc136105804)

Ievads

Pēdējos gados ir palielinājusies interese pret pilna pārklājuma ceļa meklēšanas algoritmu attīstīšanu robotizēto sistēmu un mākslīgā intelekta nozarēs. Iespēja efektīvi izbraukt un pārklāt dažādas kompleksas, dinamiskas vides ir ļoti svarīga vairākiem uzdevumiem: glābšanas operācijām, drošības uzraudzībai, kartēšanai, vides monitoringam un īpaši tīrīšanai. Mājas tīrīšanas robotu tirgū jau no 2014 gada ir novērojama stabila izaugsme. Palielinoties interesei pēc gudrās mājas tehnoloģijas un automatizācijas, pieprasījums pēc mājas tīrīšanas robotiem arvien aug. Īpaša tirgus izaugsme tika novērota COVID – 19 pandēmijas laikā. Sakarā ar pandēmiju daudzi cilvēki bija spiesti pavadīt vairāk laika mājās, tāpēc palielinājās interese pēc produktiem, kas ļauj saturēt dzīvokļus tīrākus. Saskaņā ar “*Markets and Markets*” pētījumu, starp 2021. gadu un 2027. gadu, var sagaidīt 16% salikto gada pieauguma tempu (*Compound Annual Growth Rate*)[[1]](#footnote-1). Zinātniskais raksts pamato tik strauju izaugsmi ar palielinātu pieprasījumu pēc inovatīviem mājas tīrīšanas veidiem, kā arī gudrās mājas tehnoloģijas izplatību. Mājas tīrīšanas robotu tirgū ir daudz konkurējošo ražotāju, piemēram: “*iRobot*”, “*Dyson*”, “*LG Electronics*”, “*Ecovacs* *Robotics*” un “*Samsung* *Electronics*”. Augot pieprasījumam pēc mājas tīrīšanas robotiem, ražotāji cenšas uzlabot savu robotu efektivitāti, lai piesaistītu vairāk klientu.

Īpaši būtiski robotu efektivitāti ietekmē tā programmatūra: navigācijas algoritms, kartēšanas algoritms un ceļa plānošanas algoritms. Lielākā daļā tieši no ceļa plānošanas algoritma ir atkarīgs, cik ātri un kvalitatīvi robots paveiks savu uzdevumu. Algoritmam ir jānodrošina pilns telpas pārklājums, šķēršļu apbraukšana, kā arī jāatrod ceļš ar minimālo garumu un mazāku pagriezienu skaitu. Lai veicinātu pilnu pārklājumu ražotāji izmanto dažādas metodes, kuras savukārt ir atkarīgas no vienlaicīgas lokalizācijas un kartēšanas (SLAM) sistēmas precizitātes. Mūsdienīgie tīrīšanas roboti izmanto nepilnvērtīgas metodes pārklājuma ceļa meklēšanai vairāku iemeslu dēļ. SLAM sistēmu neprecizitātes dēļ nav iespējams precīzi izbraukt iepriekš ieplānoto trajektoriju, kā arī nav iespējams pietiekami precīzi nokartēt vidi.

Šodien neviens ražotājs nepiedāvā efektīvu risinājumu klientiem, kuri vēlās izmantot savās mājās vairākus tīrīšanas robotus vienlaicīgi. Mūsdienas tīrīšanas robotiem nav sistēmu, kas nodrošinātu efektīvu vairāku aģentu sadarbību kopīgas problēmas risināšanai. Vairāku aģentu pārklājuma ceļa meklēšanas algoritma izstrāde sniedz vairākus ieguvumus:

* Uzlabota efektivitāte: Pētījumi par vairāku aģentu ceļu plānošanu telpu pārklājumam var novest pie efektīvākiem tīrīšanas algoritmiem, ļaujot mājas tīrīšanas robotiem pārklāt visu iekštelpu ar minimālu lieko darbību skaitu un samazinot tīrīšanas laiku. Šāda paaugstināta efektivitāte var būt pievilcīgs pārdošanas arguments patērētājiem, veicinot šo robotu tirgus pieprasījumu.
* Koordinācija starp vairākiem robotiem: Lielākās mājsaimniecībās vai komerciālās telpās vairāku tīrīšanas robotu izvietošana var nodrošināt ātrāku un efektīvāku tīrīšanu. Tomēr vairāku robotu darbības koordinēšana var būt sarežģīta. Vairāku aģentu virzītās apmācības pieejas var palīdzēt izstrādāt algoritmus, kas ļauj vairākiem tīrīšanas robotiem efektīvi koordinēt savas darbības, izvairoties no sadursmēm un nodrošinot vides pilnīgu pārklājumu.
* Pielāgošanās dinamiskām vidēm: Iekštelpu vide var būt sarežģīta un dinamiska, laika gaitā pārvietojot mēbeles vai citus šķēršļus. Vairāku aģentu virzītā apmācība var ļaut tīrīšanas robotiem pielāgot savas ceļu plānošanas stratēģijas pārklājumam atbilstoši vides izmaiņām, nodrošinot efektīvu tīrīšanu pat tad, ja vide mainās .

Darba mērķis ir paveikt pilna pārklājuma ceļa meklēšanu vairākiem aģentiem virtuālajās vidēs izmantojot virzītās apmācības metodi. Lai veiksmīgi sasniegtu šo mērķi tika izvirzīti sekojoši darba uzdevumi:

1. Sagatavot virtuālo vidi aģentu apmācībai. Šīm nolūkam tika izmantota “Unity 3D”, kur procedurāli tiek ģenerētas dažādas vides.
2. Izmantojot virzīto apmācību uztrenēt modeli vairāku aģentu sistēmas pārklājuma ceļa meklēšanai.
3. Novērtēt modeļa efektivitāti dažādos scenārijos un veikt secinājumus.

Darba pirmajā nodaļā ir aprakstīts X, otrajā Y, trešajā …..

1. Problēmas Nostādne
   1. Vairāku aģentu pārklājuma ceļa plānošanas formalizēšana iekštelpu vidēs
      1. Aģenta un vides modelis

Vairāku aģentu pārklājuma ceļa meklēšanas kontekstā vide parasti tiek reprezentēta diskretizēta režģa vai grafa formā, kur katra šūna vai mezgls atbilst noteiktai lokācijai iekštelpu vidē[1]. Šķēršļi, piemēram sienas vai mēbeles, tiek attēlotas kā bloķētas vai nepieejamas šūnas vai mezgli.

Katrs sistēmas aģents ir mobilais robots, aprīkots ar uztveres un vadības spējām, kas ļauj tam uztvert un pārvietoties vidē. Aģenta stāvoklis var ietvert tā pašreizējo pozīciju, orientāciju un citu svarīgu informāciju, piemēram vietējo karti vai citu aģentu stāvokļus. Pieņem, ka aģenti ir homogēni attiecībā uz savām spējām, un to darbības var ietvert kustību uz kaimiņu šūnām vai mezgliem, pagriešanos vai saziņu ar citiem aģentiem. Pārejas modelis, kas apraksta aģenta stāvokļa izmaiņas atkarībā no vides rakstura un robotu dinamikas[2].

* + 1. Mērķi un ierobežojumi

Vairāku aģentu pārklājuma ceļa plānošanā galvenais mērķis ir minimizēt zaudējuma funkciju, piemēram, kopējo laiku vai attālumu, nodrošinot tajā pašā laikā vides pārklājumu[3]. Šis mērķis var tikt formalizēts kā vairāku aģentu optimizācijas problēma, ņemot vērā dažādus ierobežojumus, piemēram, aģentu ierobežotas uztveres un vadības spējas, saziņas ierobežojumus un sadursmju novēršanas prasības[4].

Papildus galvenajam mērķim var ņemt vērā arī sekundāros mērķus, piemēram, enerģijas patēriņa minimizācija vai darba slodzes līdzsvarošana starp aģentiem. Šie sekundārie mērķi var tikt integrēti zudumu funkcijā, vai ar vairāku mērķu optimizācijas tehnikām[5].

* 1. Izaicinājumi vairāku aģentu pārklājuma ceļa plānošanā

Jo aģentu skaits un vides lielums palielinās, jo palielinās vairāku aģentu pārklājuma ceļa plānošanas problēmas sarežģītība, kas padara algoritmu izstrādi, kas efektīvi spētu pārvaldīt lielus apjomus, grūtāku[6]. Mērogojamības problēmas var rasties sakarā ar iespējamo aģentu darbību un stāvokļu kombināciju skaita strauju palielināšanos, kā arī palielinātu mijiedarbību skaitu starp aģentiem. Mērogojami algoritmi, kas spēj apstrādāt lielu aģentu skaitu un sarežģītas vides, ir būtiska vairāku aģentu pārklājuma ceļa plānošanas pētniecības daļa.

Vairāku aģentu darbību koordinēšana ir liels izaicinājums vairāku aģentu pārklājuma ceļa plānošanā. Aģentiem ir jāiemācās sadarboties un izvairīties no konfliktiem, piemēram, sadursmēm vai liekiem pārklājumiem, strādājot kopā, lai sasniegtu pilnīgu un efektīvu vides pārklājumu[7]. Koordināciju var panākt, izmantojot dažādas pieejas, tostarp centralizētu lēmumu pieņemšanu, sadalītu lēmumu pieņemšanu vai abu kombināciju. Tomēr efektīvas koordinācijas stratēģijas izstrādāšana, kas līdzsvaro optimālu, skaitļošanas sarežģītību un izturību, joprojām ir nepārtraukta pētniecības problēma.

Iekštelpu vidēs mēdz būt neskaidrībās un dinamiskās izmaiņās, piemēram, kustīgie šķēršļi, gaismas apstākļu variācijas vai sensoru trokšņi. Šie faktori var ietekmēt vairāku aģentu pārklājuma ceļa plānošanas algoritmu veiktspēju, tāpēc ir grūti izstrādāt izturīgas un pielāgojamas stratēģijas[8]. Domājot par neskaidrībām un dinamiskām izmaiņām, ir nepieciešams izstrādāt algoritmu, kas spētu efektīvi apstrādāt nepilnīgu vai trokšņainu informāciju un pielāgot savu uzvedību reāllaika novērojumu pamatā.

Vairāku aģentu pārklājuma ceļa plānošanā komunikācija starp aģentiem spēlē svarīgu lomu, lai nodrošinātu koordināciju un informācijas apmaiņu. Tomēr iekštelpu vides var būt pakļautas komunikācijas ierobežojumiem, piemēram, ierobežotam joslas platumam, aizkavēm vai savienojamībai[9]. Šie ierobežojumi var ietekmēt vairāku aģentu pārklājuma ceļa plānošanas algoritmu veiktspēju, padarot grūtu efektīvu koordinācijas un informācijas apmaiņas uzturēšanu starp aģentiem. Algoritmu izstrāde, kas spēj pārvaldīt komunikācijas ierobežojumus un pielāgot savu uzvedību atkarībā no pieejamajiem komunikācijas resursiem, ir būtiska vairāku aģentu pārklājuma ceļa plānošanas pētniecības daļa.

1. SAISTĪTIE PĒTīJUMI
   1. Tradicionālas pārklājuma meklēšanas tehnikas

Pārklājuma ceļa plānošana (CPP – *coverage path planning*) ir labi izpētīta problēma robotikā. CPP galvenais mērķis ir atrast optimālu ceļu, kas ļauj robotam apmeklēt visas punktus dotā vidē, kas ir noderīgi tādos uzdevumos kā tīrīšana, uzraudzība un izpēte. Gadu gaitā ir izstrādāti daudzi algoritmi, lai atrisinātu CPP problēmu dažādos scenārijos un pieņēmumos.

Tradicionālās CPP metodes, piemēram, Bustrofedona dekompozīcija [10] un aptveroša koka pārklājums [11], darbojas pieņemot pilnīgu vides zināšanu. Piemēram, Bustrofedona dekompozīcijas metode sadala zināmu vidi šūnās, izmantojot kritiskos punktus, kas tiek identificēti gar skrāpēšanas līniju. Šis pieejas veids rada pilnīgu pārklājuma ceļu, kas minimizē pagriezienu skaitu, nodrošinot efektīvu enerģijas patēriņu un samazinot misijas laiku vidēs, kurām izkārtojums ir zināms iepriekš. Līdzīgi aptveroša koka pārklājuma metode veido brīvās telpas aptverošo koku vidē un rada pārklājuma ceļu, pilnīgi izmeklējot koku. Šī pieeja var sniegt optimālus risinājumus zināmās vidēs un ir īpaši efektīva vidēs ar sarežģītām formām un maziem šķēršļiem.

Tomēr šīs tradicionālās metodes galvenokārt ir piemērotas statiskām, zināmām vidēm. Tās rada optimālus ceļus, balstoties uz pilnīgu vides karti, kas parasti nav pieejama reālās pasaules scenārijos. Turklāt šīm metodēm trūkst elastības un pielāgojamības, lai tikt galā ar dinamiskām izmaiņām vidē. Piemēram, ja vidē parādās jauns šķērslis vai esošais pārvietojas, šīm metodēm būtu jāpārrēķina visu pārklājuma ceļš, kas var būt skaitliski dārgs un laikietilpīgs.

Turklāt tradicionālās CPP metodes nedarbojas nezināmās vidēs, kur vides karte nav pieejama iepriekš un to ir jāatklāj, izpētot. Šīs metodes parasti prasa pilnīgu karti, lai izveidotu pārklājuma ceļu, un bez šīs informācijas tās nevar nodrošināt pilnīgu vai optimālu pārklājumu. Piemēram, lai gan bustrofedona dekompozīcijas un aptverošā koka pārklājuma metodes var radīt optimālus pārklājuma ceļus zināmās vidēs, to sniegums ievērojami samazinās nezināmās vidēs. Tās var atstāt lielas neapklātas teritorijas vai radīt neefektīvus ceļus, kas palielina misijas laiku un enerģijas patēriņu.

Heiristiskas metodes, piemēram, viļņa frontes izplatīšana [12] un potenciālā lauka balstītas metodes [13], ir piedāvātas, lai tikt galā ar nezināmām vidēm. Viļņa frontes izplatīšanas metode uztver vidi kā labirintu, kas ir jāizpēta, un izplata viļņa fronti no sākumpunkta, kamēr visa vide netiek pārklāta. Šī pieeja var tikt galā ar nezināmām vidēm, bet tā neņem vērā pārklājuma ceļa kopējo efektivitāti, radot nepilnvērtīgus risinājumus. No otras puses, potenciālā lauka balstītas tehnoloģijas rada pārklājuma ceļu, sekojot potenciālā lauka gradientam, kas ir izveidots, balstoties uz zināmajām un atklātajām vides daļām. Šī pieeja var apiet nezināmus šķēršļus, bet tai ir jutība pret lokāliem minimumiem, kas var izraisīt nepilnīgu pārklājumu vai neefektīvus ceļus.

Kopsavilkumā, lai gan tradicionālās CPP metodes ir pierādījušas sevi zināmās, statiskās vidēs, tām ir ievērojamas problēmas nezināmās un dinamiskās vidēs. Tās prasa pilnīgu vides zināšanu, bez tā šīs metodes nevar nodrošināt pilnīgu un optimālu pārklājumu. Tāpēc pastāv vajadzība pēc CPP metodēm, kas var pielāgoties nezināmām vidēm un joprojām nodrošina efektīvu un pilnīgu pārklājumu.

* 1. Virzītā apmācība pārklājuma ceļa plānošanai

Atbildot uz tradicionālo CPP metožu ierobežojumiem, pētnieki ir pievērsušies elastīgākiem un pielāgojamākiem risinājumiem. Viens no virzieniem ir stimulēta mācīšanās (RL – *reinforcement learning*), kas ir mašīnmācīšanās veids, kurā aģents mācās pieņemt lēmumus, mijiedarbojoties ar savu vidi [14]. RL algoritmi ir panākusi lieliskus panākumus dažādos sarežģītos lēmumu pieņemšanas uzdevumos, sākot no spēļu spēlēšanas un kontroles uzdevumiem līdz navigācijai un ceļa plānošanai.

CPP kontekstā RL balstītās metodes piedāvā vairākas priekšrocības salīdzinājumā ar tradicionālajām metodēm. Pirmkārt, tām nav nepieciešama pilnīga vides karte iepriekš. Tā vietā tās mācās optimālu politiku, mijiedarbojoties ar vidi ar izmēģinājumu un kļūdu metodi, kas padara tās labi piemērotas nezināmām vidēm. Otrkārt, RL balstītās metodes ir elastīgākas un pielāgojamākas. Tās var tikt galā ar dinamiskām izmaiņām vidē un pielāgot savu politiku atbilstoši, atšķirībā no tradicionālajām metodēm, kas prasa pārrēķināt visu pārklājuma ceļu, kad notiek izmaiņas.

Ir piedāvātas vairākas RL balstītas CPP metodes, demonstrējot rezultātus viena aģenta scenārijos. Piemēram, Q-mācīšanās, kas ir klasiska RL algoritma, ir izmantota, lai apgūtu optimālu pārklājuma politiku nezināmās vidēs[15]. Šī pieeja vidi attēlo kā režģi, kur katra šūna atbilst stāvoklim, un aģents apgūst Q-vērtību katram stāvokļa-darbības pārim, kas atspoguļo gaidāmo rezultātu, veicot darbību stāvoklī. Atkārtoti mijiedarbojoties ar vidi, aģents atjaunina savas Q-vērtības un pakāpeniski tiecas uz optimālu politiku.

Politikas gradientu metodes, kas ir cits RL algoritma veids, arī ir piemērotas CPP. Šīs metodes tieši optimizē politiku, sekojot gaidāmā rezultāta gradientam [16]. Tās var tikt galā ar nepārtrauktām stāvokļa un darbības telpām, kas padara tās piemērotas CPP uzdevumiem, kur aģenta kustība nav ierobežota ar režģi. Tomēr, tāpat kā Q-mācīšanās, šīs metodes ir paredzētas viena aģenta scenārijos un nemērogojās līdz vairāku aģentu scenārijam sakarā ar problēmām, piemēram nestacionaritāti.

Starp RL algoritmiem Proksimālās Politikas Optimizācija (PPO – *proximal policy optimization)* ir izrādījusies kā viena no spējīgākām metodēm CPP, īpaši vairāku aģentu scenārijos. PPO, ko izstrādāja “*OpenAI*” pētnieki [17], risina dažas no izplatītākajām problēmām politikas optimizācijā, piemēram, paraugu neefektivitāti un politikas degradāciju. PPO saglabā līdzsvaru starp izpēti un izmantošanu un nodrošina, ka atjauninātā politika pārāk neatšķiras no pašreizējās politikas, tādējādi uzlabojot stabilitāti apmācības laikā.

PPO izturība, īstenošanas vienkāršība un spēja tikt galā ar lielām un nepārtrauktām darbības telpām padara to par pievilcīgu izvēli vairāku aģentu pārklājuma ceļa plānošanai. Tās spēja ierobežot politikas izmaiņas katrā atjauninājumā padara to mazāk pakļautu zemas kvalitātes atjauninājumiem, kas varētu būt īpaši noderīgi CPP, kur aģenti bieži sastopas ar grūtām navigācijas problēmām. Turklāt PPO efektivitāti, apstrādājot lielas un sarežģītas darbības telpas, varētu izmantot, lai optimizētu vairāku aģentu veiktspēju kopīgā vidē. Ar atbilstošu apbalvojuma struktūru un stāvokļu reprezentāciju, PPO varētu veicināt koordinētu un efektīvu vairāku aģentu pārklājuma stratēģiju izstrādi.

Vairāku aģentu pārklājuma ceļa plānošanas problēma (MAPP – *multi agent path planning)* rada unikālas problēmas, kas nepastāv viena aģenta scenārijos. Jo lielāks ir aģentu skaits, jo eksponenciāli pieaug stāvokļu un darbību telpu izmēri, izraisot parādību, ko sauc par dimensiju lāstu (*curse of dimensionality*)[18]. Turklāt no katra aģenta perspektīvas vide kļūst nestacionāra, jo stāvokļa pārejas ir atkarīgas ne tikai no aģenta darbībām, bet arī no citu aģentu darbībām. Šie izaicinājumi padara MAPP par daudz sarežģītāku problēmu nekā viena aģenta CPP.

Ir izstrādātas vairākas vairāku aģentu RL (MARL – *multi agent reinforcement learning)* metodes, lai risinātu šos izaicinājumus. Neatkarīga Q-mācīšanās (IQL – *independent Q-learning*) [19] ir vienkārša Q-mācīšanās paplašināšana vairāku aģentu scenārijos. IQL katrs aģents apgūst savas Q-vērtības neatkarīgi, uzskatot citus aģentus par vides daļu. Šī pieeja labi mērogojās ar aģentu skaitu, bet tas bieži noved pie neoptimāliem risinājumiem sakarā ar koordinācijas trūkumu starp aģentiem.

Lai uzlabotu koordināciju, kopējās darbības mācīšana (JAL – *joint action learning*) [20] ļauj aģentiem apgūt kopējo darbības vērtības funkciju. Šī pieeja var sasniegt labāku veiktspēju nekā IQL, bet tā cieš no dimensiju lāsta, jo kopējā darbības telpa pieaug eksponenciāli ar aģentu skaitu. Citas metodes, piemēram, koordinēta stimulēta mācīšanas (CRL – *coordinated reinforcement learning*) [21] un vairāku aģentu dziļi determinēts politikas gradients (MADDPG – *multi agent deep deterministic policy gradient)* [22], mēģina atrast līdzsvaru starp mērogojamību un koordināciju, ļaujot aģentiem apgūt individuālas politikas, koordinējot savas darbības caur centralizētu kritiķi vai komunikāciju.

Kopsavilkumā, RL balstītās metodes, īpaši MARL metodes, piedāvā risinājumu MAPP problēmai nezināmās vidēs. Tās nodrošina nepieciešamo elastību un pielāgojamību, lai tikt galā ar nezināmām un dinamiskām vidēm, un tās var mērogoties līdz vairāku aģentu scenārijam ar atbilstošiem koordinācijas mehānismiem. Tomēr ir jāpārvar vēl daudzi izaicinājumi, piemēram, līdzsvara panākšana starp mērogojamību un koordināciju, kā arī nestacionaritātes risināšana vairāku aģentu iestatījumos.

1. Proksimālās Politikas Optimizācijas pielietojums pārklājuma ceļa meklēšanā vairāku aģentu scenārijos
   1. PPO pielietojamība vairāku aģentu pārklājuma ceļa meklēšanai

Proksimālās Politikas Optimizācijas piemērotība vairāku aģentu pārklājuma ceļa plānošanai izriet no vairākiem šī spēcīgā RL algoritma aspektiem. Pirmkārt, PPO pieeja politikas optimizācijai nodrošina izturību, risinot dinamiskas un sarežģītas vides, kas kļūst īpaši vērtīgas iestatījumos, kuros tiek izmantota vides procedūru ģenerācija [17].

Tradicionālie ceļa plānošanas algoritmi parasti prasa zināmu vides karti iepriekš un balstās uz deterministiskiem noteikumiem, lai atrastu optimālus ceļus. Tomēr, procedurāla vides ģenerācija ievieš nezināmībās un sarežģītības elementus, kas padara tradicionālās metodes neefektīvas. Šeit izceļas PPO. Tā spēcīgās mācību spējas padara to īpaši piemērotu navigācijai cauri nezināmām un mainīgām struktūrām, demonstrējot tā pielāgojamību un elastību pret neparedzētām vides izmaiņām [23].

Otrkārt, PPO ir politikas balstīta metode, kas nodrošina tai spēju efektīvāk apstrādāt augstdimensiju vai nepārtrauktas darbības telpas efektīvāk, nekā tās vērtību balstītajiem analogiem. Vairāku aģentu scenārijā kopējās darbības telpas izmērs eksponenciāli pieaug ar aģentu skaitu, tādējādi efektīva šīs telpas apstrāde kļūst par kritisku prasību [24]. PPO spēja relatīvi viegli pārvaldīt augstdimensiju darbības telpas ļauj efektīvāk koordinēt un darboties vairākiem aģentiem.

Būtiska PPO iezīme ir tā pieeja, kas ierobežo politikas izmaiņas katrā atjauninājumā, nodrošinot stabilu un kontrolētu mācīšanās procesu. Tas tiek panākts, izvairoties no straujiem politikas atjauninājumiem, kas varētu novest pie nestabilas mācīšanās vai politikas degradācijas. Ņemot vērā sarežģītās mijiedarbības starp aģentiem vairāku aģentu sistēmā un rezultātā radušās nestacionaritātes, šī PPO iezīme kļūst īpaši izdevīga MAPP kontekstā.

Visbeidzot, PPO algoritms tika izstrādāts, ņemot vērā paralēlas vides. Tas labi saskanās ar *Unity ML-Agents* platformas spējām, kas efektīvi izmanto vairākas paralēlas vides [25]. Šī iezīme ļauj paātrināt apmācības procesu, un ņemot vērā vairāku āģentu sistēmu apmācības procesa būtiski ilgo dabu, PPO un *Unity ML-Agents* kombinācija var nodrošināt ievērojamas ātruma un efektivitātes priekšrocības.

* 1. PPO pārskats

Proksimālā Politikas Optimizācija (PPO), ko izstrādāja “*OpenAI*” pētnieki, ir politikas optimizācijas metožu evolūcija, kas tapa risinot vairākas problēmas, kas saistītas ar iepriekšējām politikas gradientu metodēm, tostarp paraugu neefektivitāti un grūtības ar pielāgošanu [17]. Metodes centrālais princips ir vērsts uz pārmērīgi lielu politikas atjauninājumu novēršanu, kas tieši veicina stabilitāti un izturību mācīšanās procesā, kas ir divas būtiskas īpašības efektīvai RL lietošanai [23].

Plašākā RL kontekstā termins "aģents" tiek izmantots, lai apzīmētu autonomu būtni, kas mijiedarbojas ar apkārtējo vidi. Konkrētā gadījumā, kas attiecas uz iekštelpu navigāciju, aģents varētu būt tīrīšanas robots, kas pārvietojas pa istabu vai ēku. Aģenta mijiedarbība ar vidi ietver noteiktu "darbību" veikšanu, piemēram, kustēšanos vai pagriešanos, balstoties uz tā pašreizējo "stāvokli", kas var ietvert parametrus, piemēram, tā pašreizējo pozīciju un orientāciju[14].

Aģenta darbību sekas tiek novērtētas, pamatojoties uz "apbalvojuma" sistēmu, kas skaitliski izsaka katras darbības efektivitāti vai vērtību. Piemēram, tīrīšanas robota scenārijā apbalvojums varētu būt saistīts ar pārklātās teritorijas lielumu. Visu šo darbību kopums veido aģenta "politiku", kas ir būtisks elements RL apmācības laikā. Politiku var uztvert kā projekciju no pašreizējā stāvokļa līdz nākamajai veicamajai darbībai, kas vada aģenta uzvedību vidē [26].

RL mērķis ir apgūt optimālu politiku, kas maksimizē gaidāmo kumulatīvo apbalvojumu. Tas ir iteratīvs process, kurā politika tiek parādīta kā funkcija, kas parametrizēta ar noteiktiem regulējamiem parametriem. Algoritms tad cenšas iteratīvi atjaunināt šos parametrus virzienā, kas palielina gaidāmo apbalvojumu. Tomēr šis process var būt nestabils lielu politikas atjauninājumu dēļ, kas noved pie neregulāras apmācības vai pat gaidāmā apbalvojuma samazināšanās, kas ir problēma, ko bieži sauc par politikas degradāciju[27].

Šīs problēmas risināšana ir viens no nozīmīgākajiem PPO ieguldījumiem. Algoritma dizains ļauj tam ierobežot politikas atjauninājumu katrā solī, nodrošinot, ka tas paliek noteiktā tuvumā no pašreizējās politikas. Tieši no šīs īpašības izriet nosaukums "*proximal policy optimization*", jo tas novērš straujas izmaiņas politikā, tādējādi veicinot stabilitāti un pakāpenisku uzlabošanos mācīšanās procesā. Šādas mērķtiecīgas pieejas priekšrocība ir tā, ka tā nodrošina līdzsvarotu kompromisu starp jaunu darbību izpēti un jau apgūtu uzvedību izmantošanu, saglabājot efektīvu mācīšanās tempu[24].

PPO inovatīvā pieeja nodrošina izturīgu risinājumu, kas ne tikai pārvalda RL sarežģīto dinamiku, bet arī veicina efektīvu savākto paraugu izmantošanu. Šīs sasniegums veicina PPO plašu izmantošanu, ieskaitot vairāku aģentu ceļa plānošanu iekštelpās.

* 1. PPO vairāku aģentu sistēmā

Pielietojot Proksimālās Politikas Optimizāciju vairāku aģentu ceļa plānošanas (MAPP) jomā katram individuālam aģentam sistēmā būtu savs politikas modelis, kas pārstāvētu tā uzvedības karti, ņemot vērā dažādos stāvokļus un darbības, kas saistītas ar konkrēto uzdevumu. Šo politiku dinamiskais raksturs atspoguļotu unikālas situācijas, ar kurām saskaras katrs aģents, un pielāgotos vides variācijām.

Stāvokļa reprezentācija ir svarīgs aspekts, kas jāņem vērā vairāku aģentu scenārijā. Tas pārsniedz tikai individuālā aģenta stāvokli un ietver informāciju par citiem aģentiem, ņemot vērā faktorus, piemēram, to pašreizējas pozīcijas, virzienus un sasniegto pārklājumu. To var uzskatīt par visaptverošāku vides izpratni, kas ņem vērā ne tikai aģenta tuvējo apkārtni, bet arī visu sistēmas zināšanas, sekmējot sistēmas vispārējo efektivitāti [22].

Turklāt šajā kontekstā darbības var ietvert elementāras navigācijas darbības, piemēram, pārvietošanos uz priekšu vai rotācijas regulēšanu. Apbalvojums, kas ir svarīga RL daļa, tiktu strukturētas tā, lai veicinātu vides efektīvu un visaptverošu pārklāšanu. Piemēram, varētu piešķirt augstāku atlīdzību, ja aģenti izvairās no pārklājošiem ceļiem, tādējādi veicinot sadarbību un samazinot lieku pārklāšanu [3].

Apmācības posmā katrs aģents neatkarīgi mijiedarbotos ar vidi, krājot pieredzi stāvokļa-darbības-apbalvojuma secībā. Šīs secības veido aģenta apmācības pamatu, sniedzot vērtīgu ieskatu par noteiktu darbību ietekmi konkrētajos stāvokļos. Izmantojot PPO pieeju, aģenti iteratīvi pielāgotu savas politikas, izmantojot iegūto pieredzi, kas noved pie darbību koordinācijas uzlabošanas un efektivitātes pārklājumā palielināšanas [28].

Nozīmīgs apsvērums vairāku aģentu scenārijos ir aģentu mijiedarbības vadība. *Unity ML-Agents* rīkkopa, kas izstrādāta, lai atbalstītu vairāku aģentu apmācību, piedāvā šim jautājumam pieņemamu risinājumu. Tā ļauj katru aģentu uzskatīt par individuālu būtni, uzskatot citus par vides daļu - pieeju, ko sauc par decentralizētu izpildi. Šis ietvars ļauj katru aģentu apmācīt, izmantojot savu PPO algoritma instanci, samazinot apmācības problēmas sarežģītību un nodrošinot mērogojamību [25].

Neskatoties uz iepriekš minētajām priekšrocībām, ir svarīgi atzīmēt, ka apmācības process un aģentu uzvedība lielā mērā ir atkarīga no stāvokļa reprezentācijas un apbalvojuma funkcijas izvēles. Stāvokļa reprezentācijai jāatspoguļo precīzi vides apstākļi un aģentu statuss, savukārt atlīdzības funkcijai jāveicina aģentu virzīšanās uz vēlamo uzvedību. Tāpēc šo komponentu rūpīga izstrāde un regulēšana ir būtiska PPO veiksmīgai pielietošanai MAPP procedurāli ģenerētās simulētās vidēs [29].

1. Iekštelpas vides procedurāla ģenerēšana
   1. Procedurālas ģenerācijas aktualitāte pētītajā problēmā

Procedurāla ģenerācija ir koncepts, kas cieši saistīts ar daudzām interaktīvām lietojumprogrammām, piemēram, videospēlēm, un tas ieņem kritisku un nenomaināmu lomu vairāku aģentu ceļa plānošanas (MAPP) kontekstā. Būtībā procedurāla ģenerācija attiecas uz vides algoritmisko radīšanu dinamiski, kamēr simulācija vai spēle progresē. Tas ir ievērojami kontrastē ar tradicionālo pieeju: izmantot iepriekš definētu, statisku izkārtojumu. Šīs metodes iedzimtā elastība un pielāgojamība to padara par īpaši izdevīgu metodi dažādu un sarežģītu vižu veidošanai, kas nepārtraukti izaicina un pilnveido iesaistīto aģentu pielāgošanās un vispārināšanas spējas [30].

MAPP specifiskajā kontekstā no aģentiem - vai tie ir virtuālās būtnes simulācijā vai fiziski roboti reālās pasaules scenārijā - tiek prasīts, lai viņi efektīvi un rūpīgi pārklātu doto vidi. Šīs vides var ievērojami atšķirties pēc to izmēra, sarežģītības, struktūras un citām būtiskām īpašībām. Kad šīs vides tiek veidotas, izmantojot procedurālo ģenerēšanu, tas nozīmē, ka aģentiem ir jābūt pielāgojamiem un daudzpusīgiem, lai spētu tikt galā ar plašu situāciju un izaicinājumu spektru. Tie var svārstīties no navigācijas pa dažādiem telpu izkārtojumiem un manevrēšanu ap dažādām mēbeļu izvietojumiem, līdz neplānotu šķēršļu izvairīšanu un savu ceļu optimizāciju efektivitātes nolūkos [31].

Paša vides neprognozējamība un dinamika atspoguļo reālas pasaules scenārijus precīzāk. Tāpat kā tīrīšanas vai uzraudzības robotiem var būt nepieciešams darboties dažādās ēkās vai apgabalos, katram ar savām unikālām problēmām un prasībām, MAPP aģentiem procedurāli ģenerētās vidēs ir jāmācās pielāgot savas stratēģijas vides specifikācijām. Šī vides adaptācija veido efektīvas vairāku aģentu sistēmas būtību.

Turklāt, izmantojot procedurālo ģenerēšanu, ir iespējams simulēt ievērojami lielāku skaitu unikālu scenāriju, salīdzinot ar statiskiem vides dizainiem. Tas tieši sekmē izturīgāku mācīšanās modeļu izstrādi aģentiem, jo tie tiek pakļauti daudzveidīgām problēmām, kuras tiem ir jāmācās pārvarēt. Kad aģenti nevar paļauties tikai uz konkrētas, atkārtojošās vides atmiņā esošo risinājumu, tie tiek spiesti mācīties un pieņemt fundamentālākas, universālas stratēģijas maršruta plānošanā[32]. Aģenti ir spiesti saprast efektīvas pārklāšanas pamatprincipus, nevis pārmērīgi pielāgoties konkrētam statisku vides kopumam.

Kopsavilkumā, procedurāla ģenerācija rada bagātu, daudzveidīgu un izaicinājumiem pilnu apmācības iespēju MAPP aģentiem, veicinot izturīgu, pielāgojamu un efektīvu pārklājuma stratēģiju izstrādi, kas var vispārināties plašā scenāriju klāstā.

* 1. Vides skeleta veidošana

Pirmā fāze procedurālas iekštelpu vides radīšanā ietver vides pamata skeleta radīšanu. Šis “skelets” būtībā kalpo kā pamats, uz kura tiks uzklāti nākamie vides elementi, piemēram, durvis un mēbeles. Skelets galvenokārt sastāv no istabu pamata izkārtojuma vidē, ievērojot noteiktus kritērijus un ierobežojumus, lai nodrošinātu efektīvu un realizējamu izkārtojumu [31].

Visvienkāršākais veids, kā radīt šo pamata skeletu, ir izvietot taisnstūrveida istabas nejauši pieejamā vides telpā. Istabu nejauša izvietošana piedāvā iedzimtu priekšrocību, radot bagātīgu istabu izvietošanas variāciju, nodrošinot, ka katrs procedurāli ģenerētais vides elements ir unikāls un rada savu izaicinājumu kopumu aģentiem. Tomēr, lai nodrošinātu, ka vide joprojām ir saprātīga un pārvietojama, istabu izvietošanas procesā ir jāiekļauj noteikti ierobežojumi.

Viens ierobežojums, kas var tika izmantots, ir tas, ka katrai istabai jābūt kaut vienai kopīgai sienai ar kādu citu. Tas tiek panākts, izvēloties nejaušu punktu vides telpā, to nosakot par jaunas istabas augšējo kreiso stūri. Tālāk nejauši tiek noteikts istabas platums un augstums, radot taisnstūrveida istabu ar patvaļīgu izmēru. Nākamais solis ir pārbaudīt, vai šī jaunā istaba pārklājas ar kādu no esošajām istabām. Ja tiek konstatēts pārklājums, jaunās istabas pozīcija un izmēri ir jākoriģē, lai tā dalītos ar sienu ar pārklājušo istabu, nodrošinot savienojamību, saglabājot istabu izvietošanas nejaušību. Šis process tiek atkārtots, līdz vidē ir izvietots vēlamais istabu skaits [32].

Šī metode noved pie dažādu istabu izkārtojumu radīšanas, katram ar savu unikālo konfigurāciju. Istabas var ievērojami atšķirties pēc izmēra un formas, un kopīgās sienas var radīt interesantas konfigurācijas, kas padara vidi sarežģītāku un reālistiskāku. Tomēr ir svarīgi atzīmēt, ka šajā sākotnējā stadijā šīs istabas ir izolētas telpas bez jebkādas savienojamības, izņemot kopīgās sienas - nav durvju, kas ļautu pāreju starp istabām, nav arī mēbeļu vai citu elementu, kas varētu ietekmēt aģentu darbību.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  |  | |  |  | |
| 1.att. Procedurāli ģenerēto vižu skeletu piemēri |

* 1. Durvju izvietojums

Pēc istabu izvietošanas un konfigurācijas vides iekšienē, nākamais solis procedurālas ģenerācijas procesā ietver savienojamības nodrošināšanu starp šīm istabām, stratēģiski izvietojot durvis. Šis ir kritiski svarīgs vides ģenerēšanas aspekts, jo tas nodrošina piekļuvi visām istabām, tādējādi ļaujot aģentiem efektīvi apklāt visas teritorijas. Lai to sasniegtu, tika izmantota metode no grafu teorijas, konkrēti minimālā aptverošā koka koncepts [33].

Sākotnēji istabu izkārtojums tiek reprezentēts kā grafs, kur katrai istabai atbilst mezgls un potenciālām durvīm, kas varētu savienot divas istabas, atbilst loki. Šādas grafiskas reprezentācijas radīšana kalpo dubultu mērķi: pirmkārt, tas pārveido durvju izvietošanas problēmu uz matemātiski risināmāku formu, un, otrkārt, tas nodrošina veidu, kā kvantitāti novērtēt un minimizēt 'izmaksas' vai 'svaru', kas saistīts ar katru potenciālo durvju izvietošanu, balstoties uz izvēlētajām metrikām, piemēram, attālumu starp istabām.

Pēc tam no šī grafa tiek izveidots minimāls aptverošs koks. Minimāls aptverošs koks ir apakšgrafs, kas savieno visus grafa mezglus, nodrošinot ceļu starp jebkuru mezglu pāri, vienlaikus saglabājot kopējo malu svaru minimumā. Durvju izvietošanas kontekstā šis malu svars atbilst attālumam starp istabām. Tādējādi minimālais aptverošais koks dod priekšroku durvju izvietošanai starp tuvākajām istabām, minimizējot kopējās 'izmaksas', kas saistītas ar durvju izvietošanu.

|  |  |  |  |
| --- | --- | --- | --- |
| A picture containing rectangle, square, line, diagram  Description automatically generated | A picture containing line, circle  Description automatically generated | A picture containing line, circle  Description automatically generated |  |
| 2.att. durvju izvietošanas algoritma darbības princips pa soļiem | | | |

Kad minimāls aptverošais koks ir izveidots, durvis var izvietot vidē. Minimālā aptverošā koka malas atbilstu kopīgajām sienām starp istabām, kur jāizveido durvju atveres. Pamatojot durvju izvietošanu ar minimālo aptverošo koku, tiek nodrošināti divi galvenie rezultāti: pirmkārt, visas istabas ir savienotas un pieejamas, tādējādi padarot visu vidi sasniedzamu aģentiem; un, otrkārt, vides sarežģītība ir minimizēta, saglabājot durvju skaitu (un tādējādi potenciālo ceļu skaitu) minimumā [34].

Nobeigumā, izmantojot grafu metodes un minimālā aptverošā koka konceptu, tiek nodrošināta strukturēta un matemātiski pamatota pieeja durvju izvietošanai.

|  |  |
| --- | --- |
|  |  |
| 3.att. Procedurāli ģenerēto vižu piemēri ar durvīm | |

* 1. Mēbeļu un šķēršļu ģenerēšanā

Pēdējā fāze iekštelpu vides procedurālas ģenerācijas procesā ir mēbeļu vai papildu šķēršļu izvietošana telpu konstrukcijās. Šajā posmā ir būtiska loma reālas pasaules iekštelpu vides sarežģītību un neparedzamo elementu atdarināšanā, tādējādi izaicinot MAPP aģentu pielāgošanās spējas un efektivitāti dažādos scenārijos.

Mēbeļu izvietošanas tika panākta, algoritmiski izvēloties pozīcijas katrā telpā un izvietojot atsevišķus mēbeļu objektus šajās vietās. Lai arī izvietojums var šķist nejaušs, tika izveidoti vairāki ierobežojumi un noteikumi, lai saglabātu praktisku un reālistisku telpu izkārtojumu.

Piemēram, tika definēti ierobežojumi, lai izvairītos no pārklājumiem starp mēbeļu objektiem, sienām un durvīm. Šis ierobežojums atspoguļo reālas pasaules situācijas, kur mēbeles parasti nebloķē ieejas vai izejas un netiek saliktas viena uz otras. Papildus ir svarīgi saglabāt pietiekami daudz brīvas vietas, lai aģenti spētu pabraukt ap mēbelēm. Tas nodrošina, ka aģentiem ir pietiekami daudz manevrēšanas telpas, lai efektīvi izpildītu savus uzdevumus, vienlaikus risinot mēbeļu apiešanas izaicinājumu.

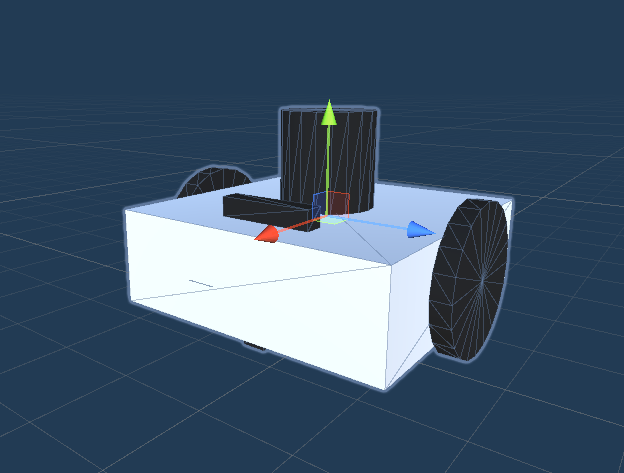
Procedurāla ģenerācija attiecas arī uz mēbeļu objektu veidiem, kas tiek ievietoti telpās. Simulācijā tika iekļauti dažādi mēbeļu objekti, kuri raksturojas ar atšķirīgu izmēru, formu un funkcionalitāti. Mainot mēbeļu objektus un to daudzumu katrā telpā, tika ieviesta lielu mainīguma un sarežģītības pakāpe.

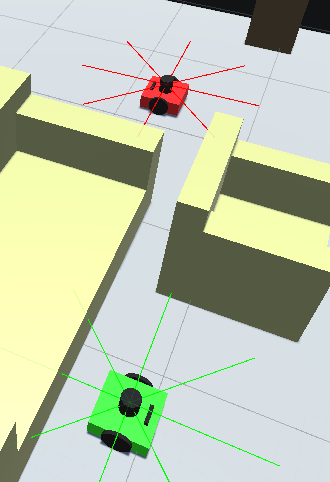
Rezultātā simulētā iekštelpu vide aģentiem rada dažādus izaicinājumus. Sākot ar navigāciju ap lielo ēdamistabas galdu un beidzot ar šaurās atstarpes starp grāmatplauktu un sienu šķērsošanu, aģentiem ir jāpielāgo savas stratēģijas, lai tiktu galā ar plašu situāciju spektru.

Kopsavilkumā, procedurāla ģenerācija darbojas kā ietekmīgs rīks dažādu un izaicinājumiem pilnu iekštelpu vides simulēšanai MAPP. Radot dinamiski mainīgus izkārtojumus, kas ir pilni ar reālistiskiem šķēršļiem, šī pieeja sekmē spējīgu, pielāgojamu aģentu izveidi, kuri spēj efektīvi tikt galā ar daudzām situācijām.

|  |  |
| --- | --- |
|  |  |
|  |  |
| 4.att. Procedurāli ģenerēto vižu piemēri ar durvīm un mēbelēm | |

1. Aģents





Diagram

Description automatically generated

Atsauksmes

[1] E. Galceran and M. Carreras, “A survey on coverage path planning for robotics,” *Rob Auton Syst*, vol. 61, no. 12, pp. 1258–1276, 2013, doi: https://doi.org/10.1016/j.robot.2013.09.004.

[2] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[3] H. Choset, “Coverage for robotics - A survey of recent results,” *Ann Math Artif Intell*, vol. 31, no. 1–4, 2001, doi: 10.1023/A:1016639210559.

[4] L. M. Rekleitis, G. Dudek, and E. E. Milios, “Multi-robot exploration of an unknown environment, efficiently reducing the odometry error,” in *IJCAI International Joint Conference on Artificial Intelligence*, 1997.

[5] M. J. Matarić, “Reinforcement Learning in the Multi-Robot Domain,” *Auton Robots*, vol. 4, no. 1, 1997, doi: 10.1023/A:1008819414322.

[6] L. E. Parker, “ALLIANCE: an architecture for fault tolerant multirobot cooperation,” *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, pp. 220–240, 1998, doi: 10.1109/70.681242.

[7] R. Zlot, A. T. Stentz, M. B. Dias, and S. Thayer, “Multi-robot exploration controlled by a market economy,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2002. doi: 10.1109/robot.2002.1013690.

[8] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, “Collaborative multi-robot exploration. In: Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065).Vol 1. IEEE; 2000,” *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 1, no. February, 2000.

[9] M. A. Batalin and G. S. Sukhatme, “Spreading Out: A Local Approach to Multi-robot Coverage,” in *Distributed Autonomous Robotic Systems 5*, 2002. doi: 10.1007/978-4-431-65941-9\_37.

[10] H. Choset and P. Pignon, “Coverage Path Planning: The Boustrophedon Cellular Decomposition,” in *Field and Service Robotics*, London: Springer London, 1998, pp. 216–222. doi: 10.1007/978-1-4471-1273-0\_32.

[11] Y. Gabriely and E. Rimon, “Spanning-tree based coverage of continuous areas by a mobile robot,” *Ann Math Artif Intell*, vol. 31, no. 1, pp. 77–98, 2001, doi: 10.1023/A:1016610507833.

[12] A. Zelinsky, R. A. Jarvis, J. Byrne, and S. Yuta, “Planning Paths of Complete Coverage of an Unstructured Environment by a Mobile Robot,” 2007.

[13] Y. Koren and J. Borenstein, “Potential field methods and their inherent limitations for mobile robot navigation,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 1991. doi: 10.1109/robot.1991.131810.

[14] R. S. Sutton and A. G. Barto, “Reinforcement Learning: An Introduction. Cambridge: MIT Press.,” *MA: MIT Press.[Google Scholar]*, 1998.

[15] C. J. C. H. Watkins and P. Dayan, “Q-learning,” *Mach Learn*, vol. 8, no. 3, pp. 279–292, 1992, doi: 10.1007/BF00992698.

[16] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, “Policy Gradient Methods for Reinforcement Learning with Function Approximation,” in *Advances in Neural Information Processing Systems*, S. Solla, T. Leen, and K. Müller, Eds., MIT Press, 1999. [Online]. Available: https://proceedings.neurips.cc/paper\_files/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf

[17] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal Policy Optimization Algorithms,” Jul. 2017.

[18] L. Busoniu, R. Babuska, and B. De Schutter, “A Comprehensive Survey of Multiagent Reinforcement Learning,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, 2008, doi: 10.1109/TSMCC.2007.913919.

[19] M. Tan, “Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents,” in *Machine Learning Proceedings 1993*, 1993. doi: 10.1016/b978-1-55860-307-3.50049-6.

[20] J. Hu and M. P. Wellman, “Multiagent reinforcement learning: Theoretical framework and an algorithm,” *Proceedings of the fifteenth international conference on machine learning*, vol. 242, 1998.

[21] C. Guestrin, M. G. Lagoudakis, and R. E. Parr, “Coordinated Reinforcement Learning,” in *International Conference on Machine Learning*, 2002.

[22] R. Lowe, Y. I. WU, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, “Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper\_files/paper/2017/file/68a9750337a418a86fe06c1991a1d64c-Paper.pdf

[23] H. Mao, S. B. Venkatakrishnan, M. Schwarzkopf, and M. Alizadeh, “Variance reduction for reinforcement learning in input-driven environments,” in *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[24] V. Mnih *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, 2015, doi: 10.1038/nature14236.

[25] A. Juliani *et al.*, “Unity: A General Platform for Intelligent Agents.” May 2018.

[26] K. Doya, “Reinforcement learning in continuous time and space,” *Neural Comput*, vol. 12, no. 1, 2000, doi: 10.1162/089976600300015961.

[27] H. Van Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double Q-Learning,” in *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, 2016. doi: 10.1609/aaai.v30i1.10295.

[28] Ö. Şimşek, A. P. Wolfe, and A. G. Barto, “Identifying useful subgoals in reinforcement learning by local graph partitioning,” in *ICML 2005 - Proceedings of the 22nd International Conference on Machine Learning*, 2005. doi: 10.1145/1102351.1102454.

[29] P. Hernandez-Leal, B. Kartal, and M. E. Taylor, “Is multiagent deep reinforcement learning the answer or the question? A brief survey,” *Auton Agent Multi Agent Syst*, vol. 33, no. 6, 2019.

[30] A. Zafar, H. Mujtaba, and M. O. Beg, “Search-based procedural content generation for GVG-LG,” *Applied Soft Computing Journal*, vol. 86, 2020, doi: 10.1016/j.asoc.2019.105909.

[31] J. Togelius, N. Shaker, and M. J. Nelson, “Procedural Content Generation in Games, 1. Introduction,” *IEEE Trans Comput Intell AI Games*, vol. 10, no. 3, 2015.

[32] A. Summerville *et al.*, “Procedural content generation via machine learning (PCGML),” *IEEE Trans Games*, vol. 10, no. 3, 2018, doi: 10.1109/TG.2018.2846639.

[33] A. Habib, M. Akram, and C. Kahraman, “Minimum spanning tree hierarchical clustering algorithm: A new Pythagorean fuzzy similarity measure for the analysis of functional brain networks,” *Expert Syst Appl*, vol. 201, p. 117016, 2022, doi: https://doi.org/10.1016/j.eswa.2022.117016.

[34] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, “Search-based procedural content generation: A taxonomy and survey,” in *IEEE Transactions on Computational Intelligence and AI in Games*, 2011. doi: 10.1109/TCIAIG.2011.2148116.

1. Markets and Markets “Household Robots Market Size, Share & Trends Analysis Report By Offering, By Type (Domestic, Entertainment & Leisure), By Application (Vacuuming, Lawn Mowing), By Region, And Segment Forecasts, 2020 - 2027.” Atvērts 09.03.2023

   https://www.marketsandmarkets.com/Market-Reports/household-robot-market-253781130.html [↑](#footnote-ref-1)