

VEČAGENTNO NAČRTOVANJE POTI ZA SKUPINO AVTOMATSKO VODENIH VOZIČKOV V INDUSTRIJSKEM OKOLJU S POLNILNICAMI

Andrej Zdešar, Matevž Bošnak, Viktor Zaletelj, Rok Vrabič,
Gregor Klančar

Izvleček:

V članku je predstavljen pristop, ki omogoča učinkovito načrtovanje poti za več agentov. Okolje je opisano z grafom, ki predstavlja omrežje cest in križišč za vozila različnih oblik in hitrosti. Načrtovanje temelji na algoritmu za načrtovanje poti z varnimi intervali, prioritetami in varnimi mesti, kar omogoča vseživljenjsko delovanje. Algoritem je kompleten in daje skoraj optimalne poti, ki so časovno in prostorsko zvezne. Algoritem smo dopolnili z ocenjevanjem energije baterije, kar omogoča iskanje le izvedljivih planov. Z dopolnitvijo algoritma za implicitno iskanje plana do optimalnega cilja iz nabora ciljev pa lahko dosežemo izbiro varnega mesta ali polnilnice že med načrtovanjem poti. Predlagani pristop je praktično uporaben v raznovrstnih realnih industrijskih scenarijih pri nalogah razvažanja materiala znotraj industrijskega okolja, pri čemer upoštevamo, da morajo agenti med izvajanjem nalog po potrebi tudi v polnilnice. Analizirali smo tudi računsko učinkovitost algoritma glede na število vozil in velikost zemljevida. Rezultati kažejo, da je predstavljeni pristop računsko učinkovit tudi pri velikem številu vozil in v primeru velikih zemljevidov.

Ključne besede:

načrtovanje poti, večagentni sistem, avtomatsko vodeni vozički, industrijska logistika, grafi, varni intervali, varna mesta, polnilnice, prioritete

1 Uvod

Sodobni industrijski proizvodni obrati in skladišča vse pogostejše uporabljajo skupino avtomatsko vodenih vozičkov (agentov) za razvoz materiala med delovnimi postajami. Za ustrezen in nemoten transport so potrebne napredne rešitve večagentnega načrtovanja poti, ki zagotavljajo (čim bolj) optimalne poti brez trkov in zastojev. Med načrtovanjem poti moramo upoštevati tudi trenutno energijo vsakega posameznega vozila, tako da vozilo nalogo lahko opravi, sicer pa moramo poskrbeti za polnje-

nje vozila, če to nima več dovolj energije za opravljanje nalog. To torej zahteva še načrtovanje poti do najbolj primerne polnilnice.

Računska zahtevnost optimalnih algoritmov za večagentno načrtovanje poti, ki temeljijo na različicah algoritma A^* [1, 2], narašča eksponentno s številom agentov, zato so ti primerni le za majhne skupine agentov. Računska zahtevnost običajno omilimo z uvedbo suboptimalnih pristopov, ki načrtujejo pot vsakega agenta ločeno, in nato rešujejo morebitne konflikte z uvedbo prometnih pravil, lokalnega večagentnega usklajevanja, pogajanja ali prioritete [1, 2, 3]. Dodaten izziv predstavlja t. i. vseživljenjsko načrtovanje poti, kjer bodoče naloge niso znane vnaprej, in moramo zagotoviti neprekinjeno delovanje, tako da ne pride do nerešljivih situacij tudi v prihodnosti [4, 5].

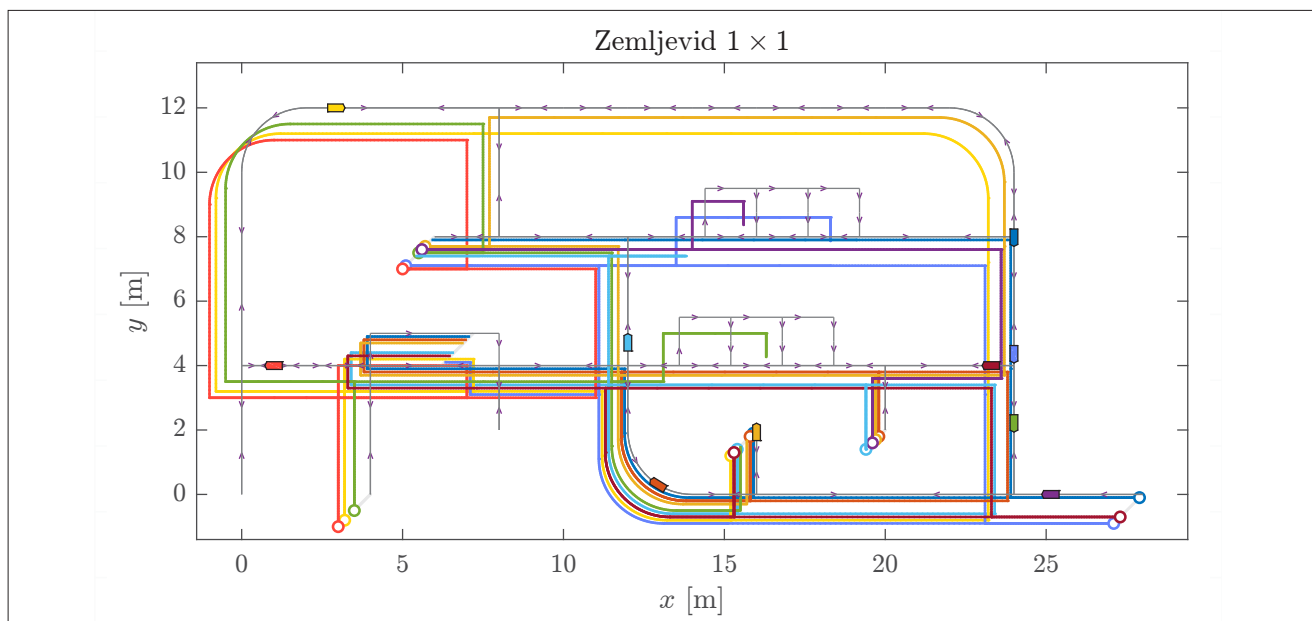
V članku je predstavljen računsko učinkovit algoritem za načrtovanje poti večagentnega sistema, ki temelji na algoritmu načrtovanja poti z varnimi intervali [6]. Algoritem je prirejen za okolje, opisano z grafom, ki predstavlja omrežje križišč in cest (*slika 1*). Načrtovanje poti omogoča iska-

Doc. dr. Andrej Zdešar, univ. dipl. inž., **dr. Matevž Bošnak**, univ. dipl. inž., oba Univerza v Ljubljani, Fakulteta za elektrotehniko; **Viktor Zaletelj**, Epi-log, d. o. o., Ljubljana; **doc. dr. Rok Vrabič**, univ. dipl. inž., Univerza v Ljubljani, Fakulteta za strojništvo; **prof. dr. Gregor Klančar**, univ. dipl. inž., Univerza v Ljubljani, Fakulteta za elektrotehniko



© The Authors 2025. CC-BY 4.0

<https://doi.org/10.5545/Ventil-31-2025-4.05>



Slika 1: Primer okolja, opisanega z grafom, v katerem se giblje 10 agentov

nje časovno in prostorsko zveznih planov, ki so blizu optimalnim rešitvam. Z vpeljavo prioritet in varnih mest smo dosegli, da je načrtovanje poti kompletno tudi v vseživljenjskem načinu delovanja. Vpeljana varna mesta lahko v praksi funkcionirajo kot polnilnice za polnjenje baterij, servisne postaje, mesta za umik idr. Predstavljeni algoritem načrtovanja poti ima vgrajen mehanizem za ocenjevanje porabe energije in omogoča implicitno izbiro optimalne polnilnice že med samim načrtovanjem poti.

2 Vseživljenjsko večagentno načrtovanje poti

V nadaljevanju je na kratko predstavljeno prioriteto načrtovanje poti z varnimi intervali za okolje, ki je opisano z grafom. Algoritem načrtovanja poti je bil nadgrajen še z vmesnimi cilji in z varnimi mesti, da smo dosegli vseživljenjsko delovanje [7]. Predlagani pristop je računsko učinkovit in je praktično uporaben za različne naloge, kot so zaporedne dostave, prevzemi in dostave v predpisanih vrstnih redih [8] idr. Omogočena je tudi izvedba načrtovanja poti v polnilnice [9], ki jih definiramo v varnih mestih. V ta namen smo algoritem še dodatno dopolnili s sprotnim ocenjevanjem porabe energije in modelom polnjenja baterij v polnilnicah. Algoritem za načrtovanje poti smo razširili tudi z implicitnim iskanjem optimalne rešitve do kateregakoli izmed nabora ciljev. Predstavljene rešitve so uporabljene za izvedbo strategije izbire varnih mest in polnilnic. Z uporabo polnilnic lahko s predstavljenim pristopom zagotovimo, da agentom nikoli ne zmanjka energije za opravljanje nalog, tako da nikoli ne pride do zastojev.

2.1 Opis okolja in predstavitev problema

Okolje opišemo z grafom $G = \langle E, V \rangle$, kjer so V vozlišča (križišča povezav) in E povezave (ceste med vozlišči). Povezave so usmerjene in utežene s trajanjem vožnje. Za vsako usmerjeno povezavo $e \in E$ predpišemo neposredno dosegljive povezave iz končnega vozlišča $v_k \in V$ povezave e oz. povezave, ki vodijo do usmerjene povezave e čez njeno začetno vozlišče $v_z \in V$. V splošnem tako lahko niso možni prehodi med vsemi povezavami, ki se stikajo v določenem vozlišču. Agenti se lahko gibljejo le v predpisanih smereh povezav.

Predpostavljamo še, da se trki med agenti lahko zgodijo le v okolici vozlišč, sicer pa ne more priti do trka med vozili na različnih povezavah, razen če gre za par usmerjenih povezav, ki predstavljata dvosmerno cesto. V primeru sekajočih se povezav lahko graf nekoliko preoblikujemo z vpeljavo dodatnih vozlišč na sečiščih povezav. Graf opremimo z informacijo o t. i. bližnjih vozliščih, ki so tako skupaj, da lahko pride do trkov med agenti, ki so v okolici kateregakoli izmed bližnjih vozlišč. S takšno predstavitevjo zemljevida lahko zapišemo skoraj poljubno okolje, v katerem se po urejenih progah vozi skupina vozil.

V članku se ukvarjamo z načrtovanjem nabora nekonfliktnih planov $\{\pi_1, \dots, \pi_k, \dots\}$ v G za skupino agentov. Vsak plan π_k se začne v vozlišču $v_{start,k} \in V$ ali na lokaciji $I_{start,k} \in [0, 1]$ znotraj ceste $e_{start,k} \in E$, kjer se nahaja agent k , in gre čez zahtevane vmesne cilje ($v_{cilj,k}(1), v_{cilj,k}(2), \dots, v_{cilj,k}(N-1)$), če so ti podani, do končnega ciljnega vozlišča $v_{cilj,k}(N)$, $v_{cilj,k} \in V$, ali končne lokacije $I_{cilj,k}$ na cesti $e_{cilj,k} \in E$. Običajna naloga, ki jo obravnavamo v tem delu, zahteva prevoz med prevzemnim mestom P in dostavnim mestom

D , pri čemer se agent na začetku v splošnem ne nahaja na prevzemnem mestu P . Med načrtovanjem poti moramo upoštevati še, da imajo agenti omejeno količino energije, ki je shranjena v bateriji. Za baterijo z nivojem polnosti $c(t) \in [0\%, 100\%]$ predpostavljamo preprost model: baterija se prazni proporcionalno s prepotovano razdaljo ($\Delta c = -\zeta_p \Delta d$, ζ_p je faktor praznjenja baterije) in se v polnilnici polni s konstantno močjo ($\Delta c = \zeta_p \Delta t$, ζ_p je faktor polnjenja baterije). V okolju imamo na voljo vsaj eno ali več polnilnic, ki se nahajajo v vozliščih ali na cestah. Med načrtovanjem poti moramo tako po potrebi oz. glede na nivo baterije $c(t)$ poiskati tudi pot do najprimernejše proste polnilnice, ki jo lahko dosežemo.

Rezultat načrtovanja poti je časovni načrt poti oz. plan $\pi = (a_1, a_2, \dots, a_n)$, ki vsebuje zaporedje akcij a_i . Te opišejo pot s podanim hitrostnim profilom od starta do cilja z zelenimi lokacijami v ustreznih časovnih trenutkih. Akcijo opišemo z $a_i = (t_i, e_i, l_i)$, kjer t_i podaja želen čas prihoda na lokacijo, ki je podana s položajem l_i znotraj povezave $e_i \in E$. Akcija a_i opisuje premik, če se lokacija glede na a_{i-1} spremeni, tj. $e_i \neq e_{i-1} \vee l_i \neq l_{i-1}$. Sicer akcija a_i predstavlja čakanje za čas $t_i - t_{i-1}$ na mestu $l_i = l_{i-1}$ znotraj povezave $e_i = e_{i-1}$.

2.2 Prioritetno načrtovanje poti z varnimi intervali

Za določitev planov za več agentov smo izhajali iz algoritma za *načrtovanje poti z varnimi intervali* (NPVI) [6]. Varni interval definira maksimalni neprekinjen časovni interval, v katerem je neko vozlišče dostopno. Posamezno vozlišče ima lahko enega ali več varnih intervalov, ki so ločeni z intervali zasedenosti. Interval zasedenosti definira časovni interval, ko vozlišče zaseda nek agent. Unija varnih in zasedenih intervalov predstavlja celotno časovno os. NPVI pri raziskovanju grafa razlikuje stanja $s = (v, VI)$, ki so definirana kot skupek vozlišča $v \in V$ in enega od njegovih varnih intervalov VI . V posamezno vozlišče lahko vstopimo večkrat (v različnih varnih intervalih).

V prvem koraku algoritma postavimo začetno stanje $s_{start} = (v_{start}, VI_{start})$ z začetnim vozliščem v_{start} na odprti seznam. Za vsako stanje s vodimo evidenco (g, h, s_p, t_p, c) , ki vsebuje najmanjšo ceno g (potovalni čas) od začetnega stanja s_{start} do trenutnega stanja s , stanje starša s_p , heuristično oceno cene h od trenutnega stanja s do cilja v_{cilj} (ocenjen čas potovanja) in najkrajši možen čas prihoda t_p v stanje s znotraj varnega intervala VI ter oceno nivoja baterije c glede na izbrano pot od začetnega stanja s_{start} . V vsaki iteraciji z odprtega seznama vzamemo stanje $s_i = (v_i, VI_i)$ z najnižjo skupno ceno $f_i = g_i + h_i$ in zanj poiščemo možne naslednike, ki jih dodamo na odprti seznam. Pri načrtovanju poti privzeto optimiziramo potovalni čas, lahko pa bi (dodatno) upoštevali tudi potrebno energijo za izvedbo poti in/ali tudi še dolžino poti.

Nivo baterije c_i v stanju s_i ocenimo glede na začetno stanje baterije c_{start} in predvideno porabo energije po

prepotovani poti. V tem delu predpostavljamo, da se baterija prazni le glede na prepotovano razdaljo, lahko pa bi upoštevali tudi bolj kompleksen model praznjenja baterije. V primeru čakanja v vozlišču ali na cesti, ki je označena kot polnilnica, upoštevamo, da se baterija polni po izbranem modelu polnjenja. Čas čakanja v polnilnici lahko predpišemo vnaprej glede na oceno potrebne energije ali pa ga določimo implicitno glede na minimalni želeni nivo baterije C_p po končanju polnjenja. Pri tem upoštevamo, da omejitev $c \in [0\%, 100\%]$ ni nikoli kršena.

Ključno pri implementaciji algoritma NPVI je določitev naslednjih stanj za širitev iskanja. Za trenutno stanje s_i določimo, v katera nova vozlišča lahko pridemo in v kateri varni interval lahko vstopimo. Pri tem preverimo, da na povezavi do naslednjega vozlišča ni konflikta (npr. trka z drugim vozilom) oz. v primeru konflikta preverimo, če ga je možno preprečiti z ustreznim čakanjem v vozlišču v_i . Upoštevamo torej le tista sosednja vozlišča, ki jih agent (po povezavah z upoštevanimi hitrostnimi omejitvami in brez trkov z drugimi agenti) lahko doseže v varnih intervalih. Zavržemo pa tudi tista vozlišča, za dosego katerih pade ocenjeni nivo baterije c od začetka plana do trenutnega vozlišča pod minimalno dovoljeno vrednost C_d . Dodatno predpišemo tudi minimalni želeni nivo baterije $C_k \geq C_d$ v končnem cilju, s čimer lahko dosežemo, da ima agent v končnem stanju dovolj energije za še neznane naslednje naloge – npr. glede na minimalno potrebno energijo, da agent iz končne lokacije doseže vsaj eno izmed polnilnic. Nivo baterije v območju $C_d < c(t) < C_k$ je možen le, če je del plana tudi polnjenje oz. čakanje v polnilnici.

Če z odprtega seznama vzamemo stanje s ciljnim vozliščem $v_{cilj} = v_i$, končamo preiskovanje grafa, saj obstaja pot do cilja. V tem primeru najdemo najboljši plan π s pregledom prednikov, ki vodijo od ciljnega stanja s_{cilj} do začetnega stanja s_{start} . Če izčrpamo odprti seznam, preden najdemo ciljno vozlišče, iskana pot ne obstaja ali ni dosegljiva (agent nima dovolj energije).

Ko vpeljemo še prioritete, dosežemo razklopljenost večagentnega načrtovanja poti, kar bistveno zmanjša računsko kompleksnost [1, 2]. Prioritete obenem ponujajo tudi možnost, da agenti opravljajo bolj ali manj pomembne naloge z različnimi prioritetami. Problem načrtovanja poti tako lahko rešujemo za vsakega agenta ločeno ob upoštevanju gibanja ostalih višjeprioritetnih agentov. Torej: ko za agenta najdemo primeren plan, nastavimo ustrezne intervale zasedenosti vseh vozlišč, ki so povezana s stanji plana. Pri naslednjem nižjeprioritetnem načrtovanju poti nato upoštevamo posodobljene intervale zasedenosti vozlišč.

Z rekurzivno uporabo algoritma za načrtovanje poti lahko dosežemo tudi iskanje poti od starta v_{start} do več zaporednih ciljev $(v_{cilj}(1), v_{cilj}(2), \dots, v_{cilj}(N))$.

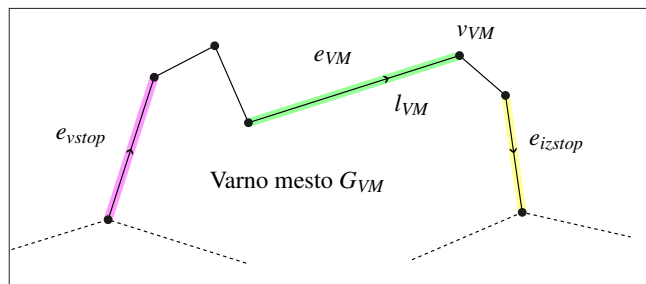
Ko dosežemo prvi cilj v_{cilj} (1) in dobimo načrt poti π_{cilj} (1), nadaljujemo z iskanjem plana π_{cilj} (2) od trenutnega cilja v_{cilj} (1) do naslednjega cilja v_{cilj} (2) itd. Če najdemo pot do končnega cilja v_{cilj} (N), združimo posamezne plane v celoten plan $\pi' = (\pi_{cilj}(1), \pi_{cilj}(2), \dots, \pi_{cilj}(N))$. Algoritem večagentnega načrtovanja poti smo priredili tudi tako, da lahko iščemo optimalno pot do kateregakoli izmed nabora ciljev. To enostavno dosežemo tako, da med načrtovanjem poti onemogočimo hevristiko ($h = 0$), iskanje pa prekinemo, ko dosežemo prvega izmed ciljev v podanem naboru. Ker algoritem iskanja v tem primeru ne uporablja hevristike, je iskanje malce bolj dolgotrajno, a običajno vseeno precej hitreje, kot če bi poiskali poti do vsakega cilja ločeno.

Ključno za računsko učinkovito izvedbo algoritma je tudi, da se vsi trki preverjajo implicitno le s preverjanjem časovnih intervalov. Glede na oblike, dimenzije in hitrosti agentov lahko vnaprej določimo minimalni varnostni čas (s katerim razširimo intervale zasedenosti), ki zagotovi, da se trki v okolici vozlišč nikoli ne zgodijo. Poudariti velja, da je predstavljeni pristop večagentnega načrtovanja poti uporaben tudi za heterogeno skupino vozil (različne dimenzije in hitrosti vozil), kar tudi ne poveča računske zahtevnosti.

2.3 Varna mesta in polnilnice

Varno mesto (VM) definiramo kot področje v prostoru (in grafu), kamor se agent po končani nalogi lahko varno umakne, če je potrebno, in neovirajoče v nedogled čaka na naslednjo nalogo. V varno mesto lahko vstopijo (in to samo enkrat na svoji poti) le agenti, ki načrtujejo pot s ciljem v njem, sicer ga ne smejo prečkati. Splošna struktura varnega mesta je prikazana na sliki 2. Varno mesto predstavimo z grafom $G_{VM} = \{V_{VM}, E_{VM}\}$, ki ni del osnovnega grafa ($E \cap E_{VM} = \emptyset$), a ga lahko iz grafa G dosežemo le po usmerjenih vstopnih cestah $e_{vstop} \in E_{VM}$ ter zapustimo le po usmerjenih izstopnih cestah $e_{izstop} \in E_{VM}$, ki vodijo nazaj v graf G . Določimo lahko tudi več vhodnih in izhodnih cest. Znotraj G_{VM} izberemo eno varno lokacijo bodisi na eni izmed cest $e_{VM} \in E_{VM}$ bodisi v vozlišču $v_{VM} \in V_{VM}$, ki omogoča neovirajoče čakanje. Med načrtovanjem poti do varnega mesta dovolimo iskanje tudi čez vstopne povezave $e_{vstop} \in E_{VM}$, ki vodijo v graf G_{VM} . Če začnemo iskanje znotraj G_{VM} , pa dovolimo iskanje tudi čez izstopne povezave $e_{izstop} \in E_{VM}$, ki vodijo do osnovnega grafa G . Da preprečimo večkratne vožnje čez varno mesto, iskanja ne nadaljujemo pri stanjih, ki se končajo v varni lokaciji. Definiramo lahko tudi več varnih mest.

Pri načrtovanju posameznega plana vedno poiščemo še plan π_{VM} od zadnjega cilja v_{cilj} (N) do enega izmed varnih mest. Končni rezultat je tako plan $\pi = (\pi', \pi_{VM})$. Ta pristop omogoča, da znotraj G ne pride do blokad agentov (npr. v slepih ulicah). S tem postane načrtovaje poti kompletno, torej iskanje vedno najde pot, če ta obstaja oz. je dosegljiva. Me-



Slika 2 : Splošna struktura varnega mesta

nimo, da je agent v času izvajanja plana π_{VM} prost, če še ni znotraj varnega mesta, zato se lahko izvajanje tega dela plana kadarkoli prekine, če je možno najti plan do novih zelenih ciljev. Če to ni možno, se agent še vedno lahko umakne v VM in od tam nadaljuje kasneje.

Vsakemu varnemu mestu predpišemo tudi kapaciteto agentov, ki jih lahko sprejme, tako da neovirajoče čakajo na cesti e_{VM} oz. na cesti pred v_{VM} v varnem mestu. Za izvedljivost mora biti skupna kapaciteta vseh varnih mest večja ali enaka številu vseh agentov N_A . Iskanje plana π_{VM} lahko omejimo le v izbrano varno mesto ali pa v kateregakoli izmed nabora varnih mest. S predstavljenim splošnim konceptom lahko izvedemo serijska in vzporedna varna mesta. Če onemogočimo preverjanje trkov na določenih cestah znotraj G_{VM} , je možna izvedba tudi t. i. virtualnih varnih mest. S takšnimi varnimi mesti lahko nadzor nad upravljanjem z agenti prepustimo zunanjemu (avtonomnemu) nadzornemu sistemu, kar omogoča poveztljivost med različnimi sistemi.

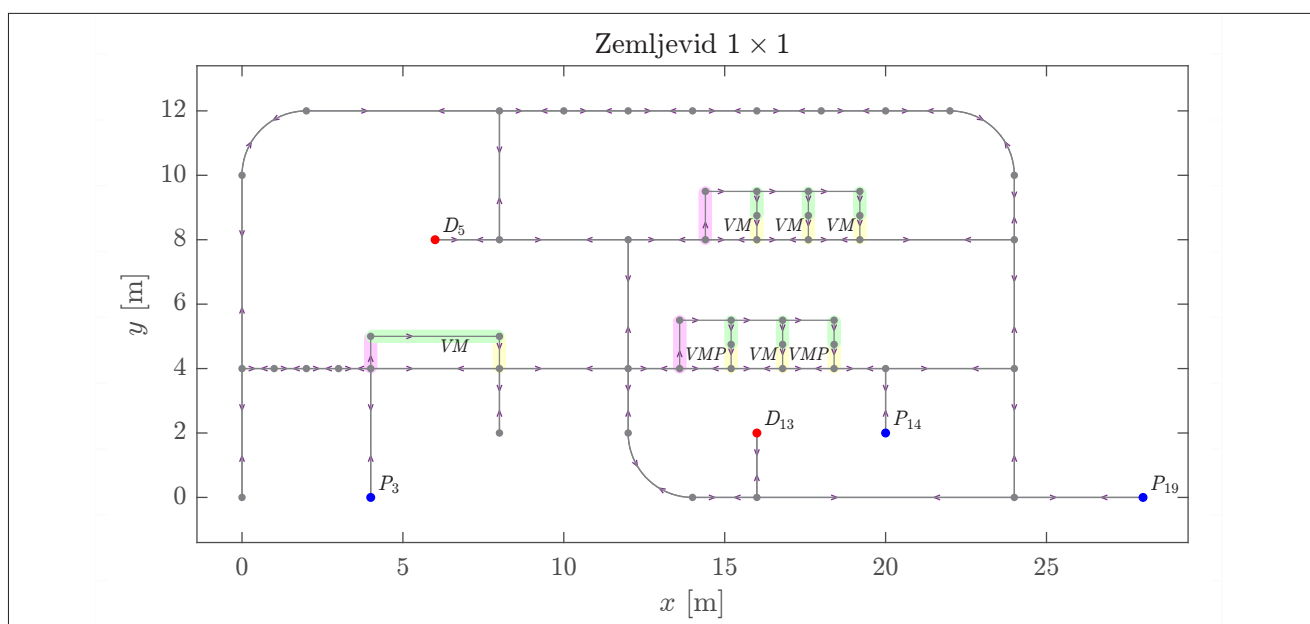
Z vpeljavo prioritete in varnih mest je omogočeno tudi vseživljenjsko delovanje, ko se naloge pojavljajo sproti in niso vnaprej znane. Osnovni koraki, ki skrbijo za dodeljevanje nalog, so:

Korak 1: Če so na voljo nove naloge in je prost vsaj en agent (ki je obiskal vse cilje in se pelje proti varnemu mestu ali že čaka v njem in je na voljo), dodelimo agentu najnižjo prioriteto.

Korak 2: Poiščemo plan za novo nalogo z izbranim agentom.

Korak 3: Če najdemo nov plan, dodelimo agentu novo nalogo, posodobimo rezervacije varnih mest, prekinemo morebitno izvajanje trenutnega plana, sprostimo zasedenosti za neizvedeni preostanek trenutnega plana in nastavimo intervale zasedenosti za novi plan ter ga začnemo izvajati. Nadaljujemo s korakom 1.

Korak 4: Če novega plana ne najdemo, nadaljujemo trenutno pot proti varnemu mestu ali v njem nadaljujemo čakanje. Za določen čas onemogočimo dodeljevanje nalog temu agentu. Nadaljujemo s korakom 1.



Slika 3 : Zemljevid 1x1 (58 vozlišč, 68 povezav, 3 prevzemna mesta P , 2 dostavni mesti D) s serijskim varnim mestom in vzporednim varnim mestom ter vzporednim varnim mestom z dvema polnilnicama

Varna mesta so praktično uporabna tudi kot polnilnice ali servisna mesta. Varnim mestom lahko torej priredimo tudi eno ali več storitev, ki jih ponujajo. Postavitev polnilnic v varna mesta se zdi logična izbira, saj varna mesta po definiciji omogočajo neovirajoče čakanje. Varno mesto s polnilnico (VMP) torej obravnavamo kot poseben tip VM -ja, ki omogoča polnjenje baterij. Primer nekaj možnih izvedb varnih mest je prikazan na *sliki 3*.

2.4 Strategija izbiranja polnilnic

V primeru, ko se upošteva tudi energija agentov, strategijo dodeljevanja nalog nekoliko spremenimo, da po potrebi omogočimo tudi polnjenje baterij v polnilnicah. V nadaljevanju je predstavljena ena možna strategija dodeljevanja nalog, ki vključuje tudi načrtovanje poti v polnilnice, ko je to potrebno ali zahtevano.

Ko nivo baterije pade pod določeno minimalno spodnjo vrednost C_m , mora agent oditi v VMP , preden začne izvajati naslednjo nalogo. Pri polnjenju predpišemo minimalni zgornji nivo baterije C_p , ki ga mora agent doseči, preden se polnjenje lahko zaključi in agent nadaljuje z izvajanjem nalog. Če nivo baterije pade pod kritično spodnjo mejo C_o , agent obstoji in ne more več izvajati nalog – potreben je poseg operaterja, zato tega stanja nikoli ne želimo doseči. Zato predpišemo spodnji nivo baterije $C_d \geq C_o$ z nekaj rezerve, tako da pri načrtovanju poti vedno najdemo le poti, ki so dosegljive. Nivo $C_k \geq C_d$ izberemo tako, da agent iz kateregakoli varnega mesta lahko doseže vsaj eno polnilnico.

Če imamo na voljo več VM -jev, lahko agentu dodelimo VM vnaprej. Možno je izvesti mnogo strategij,

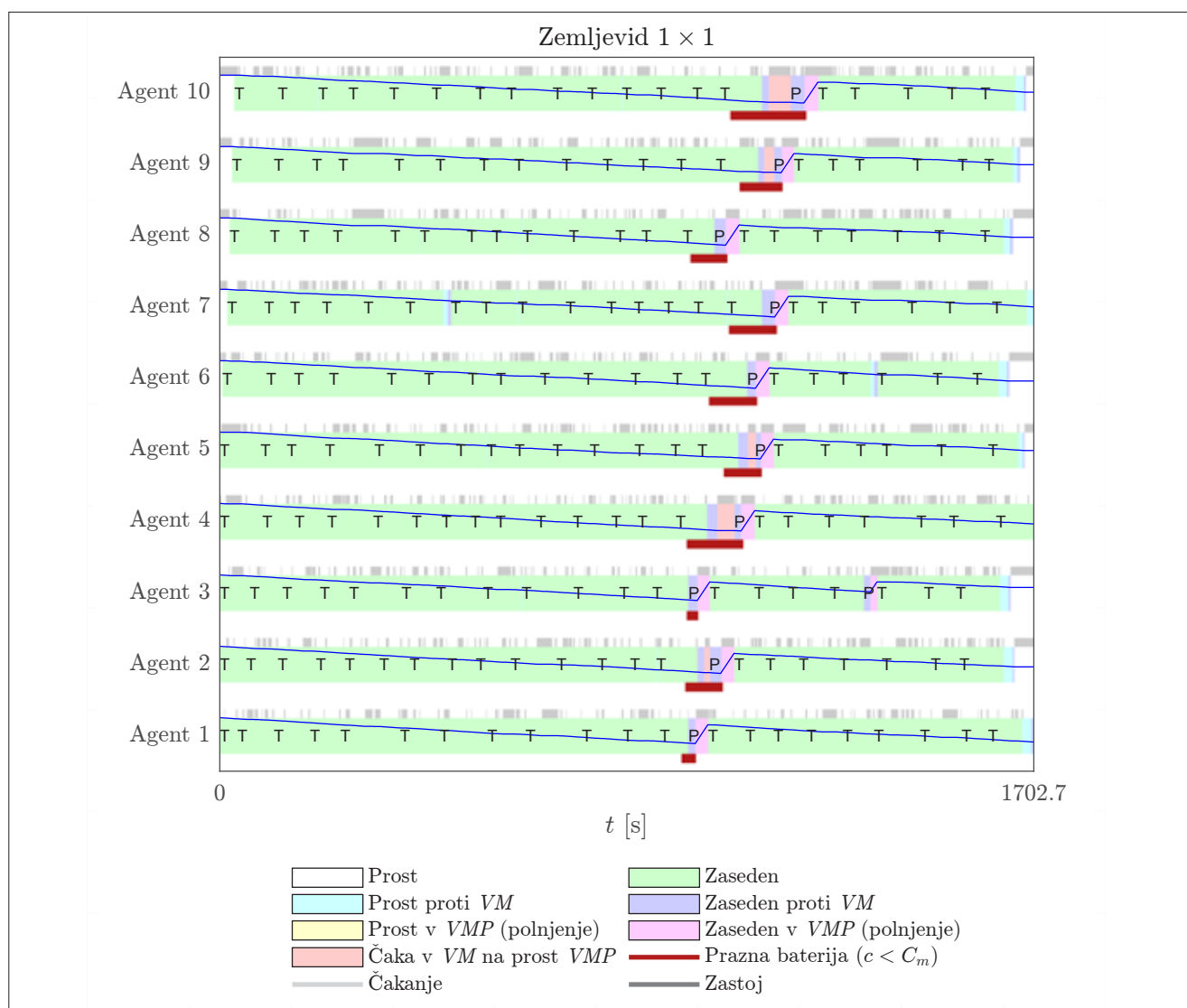
ki so odvisne od oddaljenosti vstopnega vozlišča v_{stop} in/ali izstopnega vozlišča v_{izstop} določenega VM -ja od agenta, prevzemnih in/ali dostavnih mest, tipa in zasedenosti VM -ja, nivoja baterije itd. Kot je bilo omenjeno, predstavljeni algoritem omogoča tudi implicitno iskanje optimalne poti do najboljšega cilja iz nabora ciljev. Zato v tem delu uporabimo strategijo implicitnega iskanja poti do najbolj optimalnega varnega mesta, ki ponuja zelene storitve (le čakanje ali tudi polnjenje baterije) in je na voljo (trenutno število vozil v varnem mestu je manjše od njegove kapacitete). Drugi korak za dodeljevanje nalog tako nekoliko spremenimo:

Korak 2.1: Če je nivo agentove baterije pod spodnjo mejo ($c(t) < C_m$), namesto nove naloge izberemo nalogo polnjenja in poiščemo pot do ene izmed prostih polnilnic v VMP -ju. Dokler agent ne doseže polnilnice, ne sme prevzeti nobene nove naloge. Nadaljujemo s korakom 3.

Korak 2.2: K ciljem v nalogi dodamo za zadnji del poti med nabor ciljev vse proste VM -je. Če ne najdemo izvedljive poti, k ciljem v nalogi dodamo za zadnji del poti med nabor ciljev še vse proste VMP -je. Če ne najdemo izvedljive poti, namesto nove naloge izberemo nalogo polnjenja in poskusimo poiskati le pot do katerekoli proste polnilnice v VMP -ju.

3 Simulacijski eksperimenti

S simulacijo smo preverili delovanje večagentnega načrtovanja poti za izvajanje transportnih nalog s predstavljenimi strategijami izbiranja varnih mest in polnilnic, ki implicitno išče optimalno rešitev v naboru VM -jev in/ali VMP -jev. Vsaka naloga zah-



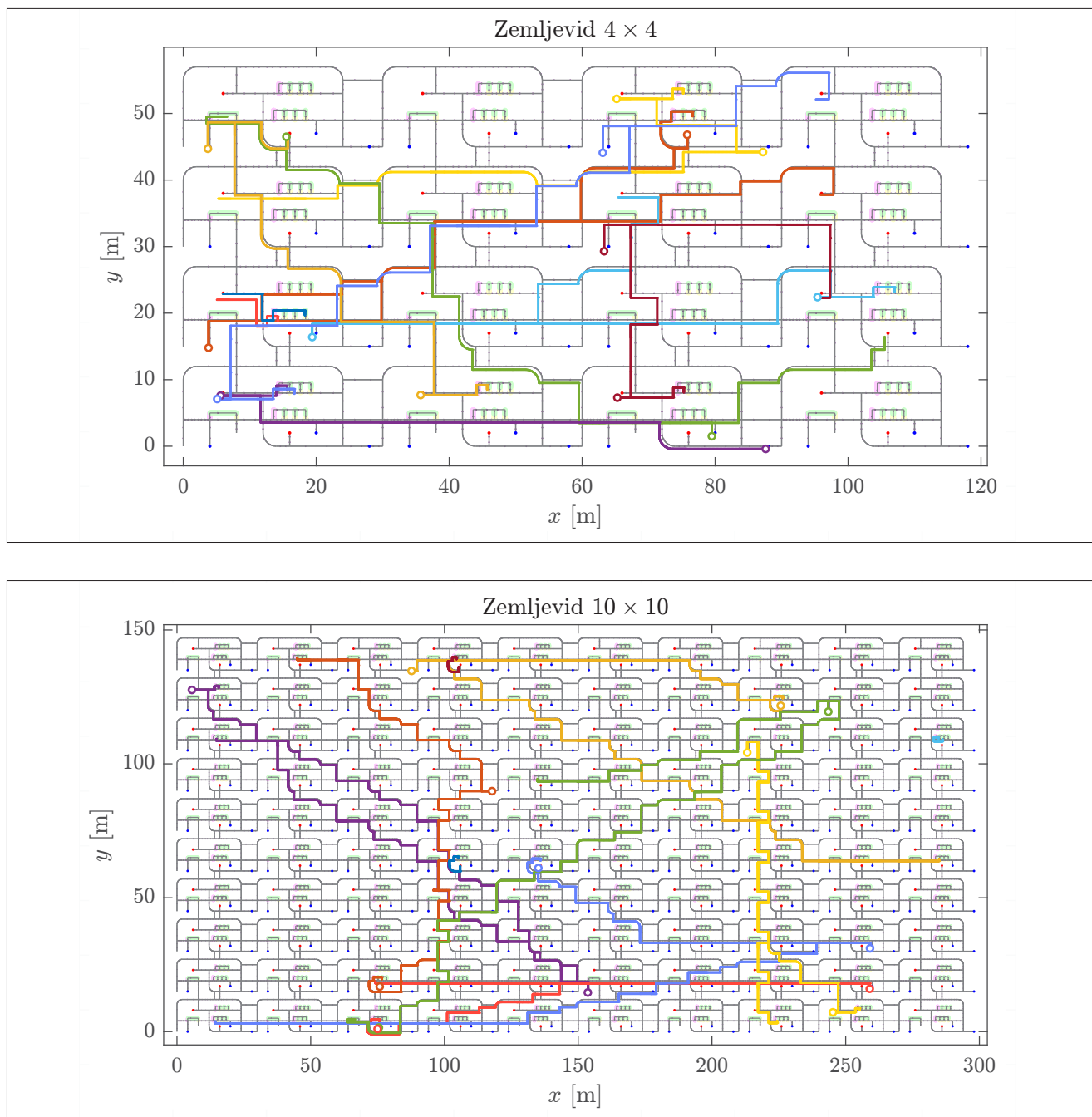
Slika 4 : Časovni potek stanj agentov in stanj baterij (modra polna črta) pri izvajanju 200 nalog na zemljevidu 1x1 (T označuje trenutek dodelitve nove transportne naloge, P pa trenutek dodelitve naloge polnjenja baterije)

teva plan vožnje π_p s trenutne lokacije agenta na naključno izbrano prevzemno mesto P , nato π_D na naključno dostavno mesto D in končno π_{VM} v varno mesto VM ali VMP . Celoten plan vsake naloge je torej $\pi = (\pi_p, \pi_D, \pi_{VM})$. Med delovanjem agenti po potrebi opravljajo tudi polnjenje svojih baterij. Izbrali smo naslednje parametre za simulacijo: dolžina vozila 0,48 m, širina vozila 0,24 m, hitrost vozila 1 m/s, varnostni faktor 2, faktor praznjenja baterije $\zeta_d = 0,001 \text{ m}^{-1}$, faktor polnjenja baterije $\zeta_p = 0,02 \text{ s}^{-1}$, kritični nivo baterije $C_o = 2 \%$, minimalni ocenjeni nivo baterije $C_d = 5 \%$, minimalni končni ocenjeni nivo baterije $C_k = 15 \%$, mejni nivo baterije $C_m = 30 \%$, minimalni nivo baterije po polnjenju $C_p = 80 \%$.

Na zemljevidu, ki je prikazan na sliki 3, smo simulirali izvajanje 200 naključnih nalog, ki jih opravlja $N_A = 10$ agentov. Slika 4 prikazuje časovne poteke stanj vseh agentov in tudi nivoje vseh baterij. Na sliki 4 so označeni tudi časovni trenutki dodeljevanja novih nalog in nalog polnjenja. Ko nivo bate-

rije pade pod mejni nivo C_m , se sproži iskanje plana do polnilnice. Če so vse polnilnice zapolnjene, opazimo, da gre agent v varno mesto, kjer čaka na prosto polnilnico – z izvajanjem novih nalog pa nadaljuje šele, ko baterija zopet doseže minimalni nivo napolnjenosti C_p . Čeprav vsak plan na koncu vsebuje tudi pot do varnega mesta, se izvajanje tega dela plana skoraj nikoli ne izvede ali pa se izvede le delno (razen, ko je zahtevano polnjenje). Običajno se takoj, ko je to mogoče, agentu dodeli nova naloga.

Delovanje večagentnega načrtovanja poti smo preizkusili še na dveh večjih zemljevidih 4x4 ter 10x10 (slika 5), ki sta zgrajena iz kopij osnovnega zemljevida (slika 3). Na sliki 6 so prikazani časovni poteki stanj še v primeru, kjer je $N_A = 10$ agentov opravilo 200 transportnih nalog na zemljevidu 4x4. V tem primeru je potrebnih več polnjenj, saj so razdalje do prevzemnih in dostavnih mest daljše. Ker so poti daljše, potrebujemo tudi več časa kot v prime-



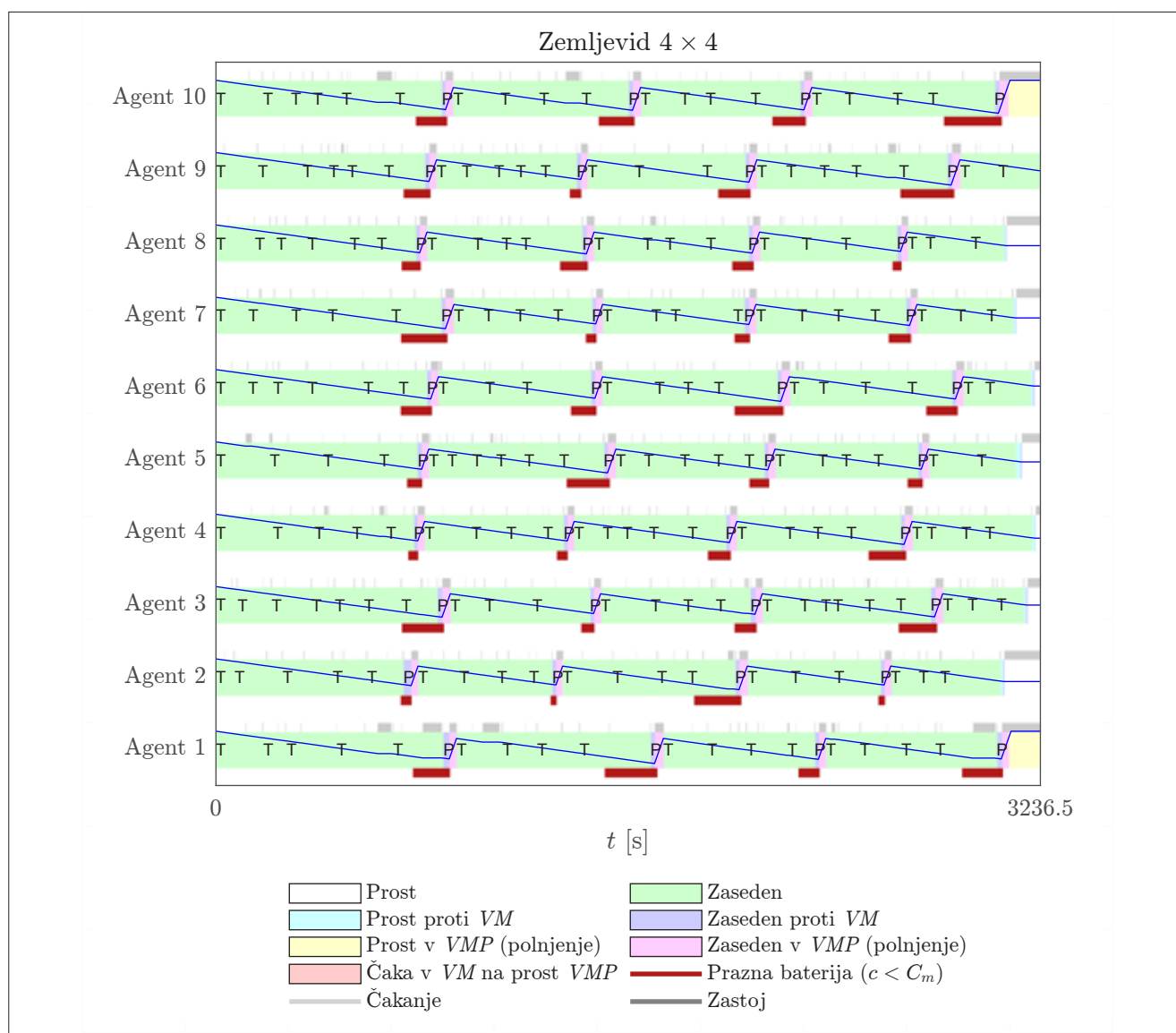
Slika 5 : Zemljevid 4×4 (928 vozlišč, 1160 povezav, 48 prevzemih mest, 32 dostavnih mest, 80 VM-jev, 32 VMP-jev) in zemljevid 10×10 (5800 vozlišč, 7340 povezav, 300 prevzemih mest, 200 dostavnih mest, 500 VM-jev, 200 VMP-jev) z nekaj vrisanimi plani

ru zemljevida 1×1 , da opravimo vse naloge. Noben agent nikoli ne čaka v varnem mestu na prosto polnilnico, saj vedno najdemo prosto polnilnico. Vendar vse polnilnice lahko ne bi bile dosegljive. To je posledica dejstva, da imamo več polnilnih mest kot agentov in da so naloge razporejene enakomerno po celotnem prostoru ter da je C_k dovolj velik, da lahko iz vsakega varnega mesta dosežemo celo več polnilnic.

S pomočjo zemljevidov 4×4 in 10×10 smo analizirali še računsko učinkovitost trenutne izvedbe al-

goritma. Uporabili smo različno število agentov, ki opravljajo transportne naloge. Simulacijsko analizo smo izvedli na računalniku z Intel Core i7-6700HQ in 16 GiB delovnega spomina. Slika 7 prikazuje čase načrtovanja poti t_c in čas izvedbe vseh nalog t_e na zemljevidu 4×4 za 2000 nalog ter na zemljevidu 10×10 za 12000 nalog pri različnem številu agentov N_A – tu smo uporabili veliko število nalog, da dobimo statistično reprezentativne rezultate.

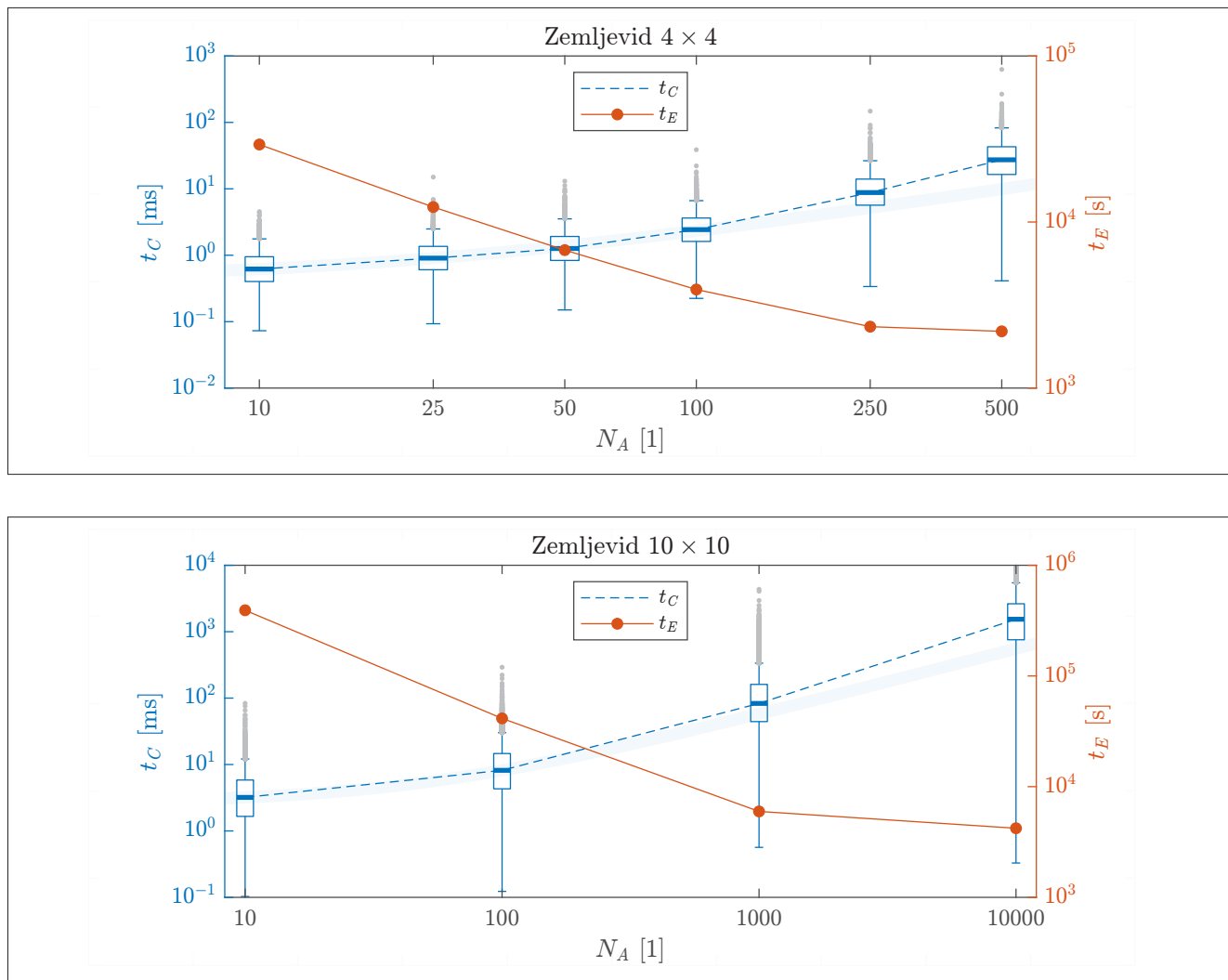
Iz podatkov o časih t_e za izvedbo vseh nalog (slika 7) lahko ugotovimo, da dvakrat več agentov lahko



Slika 6 : Časovni potek stanj agentov in stanj baterij pri izvajanju 200 nalog na zemljevidu 4×4 (glej opis na sliki 4)

opravi naloge skoraj dvakrat hitreje. Učinkovitost je odvisna od nalog, ki so v simuliranem primeru imele enakomerno izbrana ciljna mesta. Na zemljevidu 4×4 se pri 100 agentih čas t_E skrajša skoraj na deseto časa glede na 10 agentov – pri večanju števila agentov pa je prihranek časa zaradi naporov koordinacije ustrezno manjši. Tudi na velikem zemljevidu 10×10 opazimo, da 1000 agentov v primerjavi z 10 agenti lahko opravi skoraj 100-krat več nalog v enakem časovnem obdobju. To drži le, dokler ni agentov glede na velikost zemljevida preveč – v primeru zemljevida 4×4 se to zgodi pri 100 agentih, na zemljevidu 10×10 pa pri 1000 agentih. Če je agentov preveč, postane promet zelo zgoščen, zato je potrebno veliko dodatnega čakanja pri koordinaciji vozil, da ne pride do trkov. Za določen zemljevid torej ni smiselno uporabiti preveč agentov, saj na ta način ne moremo doseči izvedbe bistveno več nalog v eni časovni enoti.

Na sliki 7 opazimo tudi, da računska zahtevnost t_C narašča linearno s številom uporabljenih agentov. To je res, vse dokler ne dosežemo nasičenja, ko se računska zahtevnost poveča – a uporaba preveč agentov tudi ni smotrna. Glede na optimalne pristope v literaturi [1], kjer računska zahtevnost narašča eksponentno, je to za praktično uporabo obetaven rezultat. Razlogov za linearno naraščanje računske kompleksnosti je več. Vsak plan vnese nove intervale zasedenosti, ki lahko, ne pa vedno, razdrobijo varne intervale, kar povzroči dodatne vejitve stanj pri nižjeprioritetnih načrtovanjih poti. Kasneje se načrtovanje poti izvede le, ko določen agent opravi nalogo in postane prost ter je na voljo nova naloga. Takrat algoritem lahko pobriše znaten del intervalov zasedenosti v preteklosti, ki niso relevantni za prihodnost. To sprosti spomin in zmanjša število stanj, ki jih je potrebno pregledati med načrtovanjem poti.



Slika 7 : Čas načrtovanja poti t_C in čas izvedbe vseh nalog t_E glede na število agentov N_A na zemljevidu 4×4 z 2000 nalogami in zemljevidu 10×10 z 12000 nalogami

4 Zaključek

Predstavili smo izvedbo algoritma za večagentno načrtovanje poti. V članku je predstavljeno, kako lahko vpeljana varna mesta koristno uporabimo kot polnilnice. V algoritmu načrtovanja poti smo vključili implicitno iskanje planov do nabora možnih ciljev in ocenjevanje nivoja baterije. To omogoča načrtovanje poti, ki so dosegljive glede na trenutni nivo baterije in izvedbo načrtovanja poti v najprimernejšo polnilnico glede na ocenjene energijske potrebe. Predstavljeni algoritem je kompleten, omogoča vseživljenjsko delovanje in je praktično uporaben v različnih scenarijih. S simulacijskimi eksperimenti smo pokazali, da je trenutna izvedba algoritma za večagentno načrtovanje poti računsko zelo učinkovita in deluje hitro tudi za velike zemljevide z veliko agenti.

Viri

- [1] G. Sharon, R. Stern in sod. Conflict-Based Search for Optimal Multi-Agent Pathfinding. *Artificial Intelligence*, zv. 219, str. 40–66, 2015.
- [2] A. Andreychuk, K. Yakovlev in sod. Multi-Agent Pathfinding with Continuous Time. *Artificial Intelligence*, zv. 305, št. 103662, str. 1–32, 2022.
- [3] T. Žužek, R. Vrabič in sod. Simulation-Based Approach for Automatic Roadmap Design in Multi-AGV Systems. *IEEE Transactions on Automation Science and Engineering*, zv. 21, št. 4, str. 6190–6201, 2024.
- [4] V. Digani, L. Sabattini in sod. Ensemble Coordination Approach in Multi-AGV Systems Applied to Industrial Warehouses. *IEEE Transactions on Automation Science and Engineering*, zv. 12, št. 3, str. 922–934, 2015.
- [5] A. Zdešar, T. Žužek in sod. Vidiki koordiniranega načrtovanja poti na podlagi prioritet in časovnih oken za optimizacijo logistike avtomatsko vodenih vozil. V *Zbornik trinajste konference Avtomatizacija v industriji in gospodarstvu (AIG)*, str. 1–9. Društvo avtomatikov Slovenije, 2023.
- [6] M. Phillips in M. Likhachev. SIPP: Safe Interval Path Planning for Dynamic Environments. V *IEEE International Conference on Robotics and Automation*, str. 5628–5635, 2011.
- [7] M. Bošnjak, A. Zdešar in sod. Vseživljenjsko večagentno načrtovanje poti.

gentno planiranje poti avtomatsko vodenih vozil v intralogistiki. Ventil: revija za fluidno tehniko in avtomatizacijo, zv. 29, št. 4, str. 246–253, 2023.

- [8] M. Bošnjak, A. Zdešar in sod. Managing Just-In-Sequence Micro-Logistics with Multi-Agent Path-Finding. V IEEE International Conference on Automation Science and Engineering (CASE),

str. 404–409. 2024.

- [9] A. Zdešar, M. Bošnjak in sod. Vpeljava varnih mest za vseživljenjsko večrobotsko planiranje poti v interni logistiki. V Zbornik triintridesete mednarodne Elektrotehniške in računalniške konference (ERK), str. 163–166. Slovenska sekcija IEEE; Fakulteta za elektrotehniko, 2024.

Multi-agent path planning for a fleet of automated guided vehicles in industrial setting with charging stations

Abstract:

The paper presents an approach that enables efficient multi-agent path planning. The environment is described by a graph that represents a network of roads and crossroads for vehicles of different shapes and velocities. The multi-agent pathfinding approach is based on a safe interval path planning algorithm with priorities and safe locations, which together allow for lifelong operation. The algorithm is complete and it produces near-optimal plans that are continuous in time and space. We have extended the algorithm with battery energy estimation, which allows finding only feasible plans. Since we also added an option that implicitly selects an optimal goal in a set of goals, we can achieve selection of a safe location or charging station already during planning. The proposed approach is practically applicable in a wide variety of realistic industrial scenarios for material handling within industrial setting, where charging of the agents during execution of the tasks is also considered. We also analysed the computational efficiency of the algorithm with respect to the number of vehicles and the size of the map. The results show that the presented approach is computationally efficient even in the case of a large number of vehicles and large maps.

Keywords:

path planning, multi-agent system, automated guided vehicles, industrial logistics, graphs, safe intervals, safe locations, charging stations, priorities

Zahvala

Avtorji se zahvaljujemo za finančno podporo Javni agenciji za znanstvenoraziskovalno in inovacijsko dejavnost Republike Slovenije in podjetju Epilog d. o. o. (št. L2-60153 in P2-0219).



Clean visions.



Inovativni sistemi stavbnega pohištva za sodobno arhitekturo.

Lasten razvoj, natančna izdelava in strokovna montaža vhodnih vrat, oken, steklenih fasad, zimskih vrtov in pregradnih sten.

