



Универзитет у Београду  
Машински и Математички факултет  
Индустрија 4.0



## Пројектни задатак

ACO for robot (autonomous vehicle) path planning

Рачунарска интелигенција

Адријана Симеуновић 4001/2023

Марко Симић 4002/2023

Београд, октобар 2025. године

# Садржај

<b>1. Увод</b>	<b>3</b>
<b>2. Опис решења проблема</b>	<b>4</b>
2.1. Традиционални АСО алгоритам . . . . .	4
2.1.1. Вероватноћа одабира путање . . . . .	4
2.1.2. Ажурирање феромона . . . . .	5
2.2. Имплементација АСО алгоритма за проблем планирања путање аутономном роботу . . . . .	5
2.2.1. Функција циља . . . . .	6
<b>3. Експериментални резултати</b>	<b>8</b>
3.1. Проблем 1 . . . . .	8
3.2. Проблем 2 . . . . .	10
3.3. Проблем 3 . . . . .	12
3.4. Проблем 4 . . . . .	14
<b>4. Експериментални резултати (динамичко окружење)</b>	<b>16</b>
4.1. Проблем измене статичких препрека . . . . .	16
4.2. Проблем постојања динамичких препрека . . . . .	19
4.3. Комбиновани динамички проблем . . . . .	21
<b>5. Закључак</b>	<b>25</b>
<b>6. Литература</b>	<b>26</b>

## Списак слика

3.1.	<i>Најкраћа путања проблем 1</i>	9
3.2.	<i>Најкраћа путања из рада [1] проблем 1</i>	9
3.3.	<i>Најкраћа путања - проблем 2</i>	11
3.4.	<i>Најкраћа путања из рада [2] - проблем 2</i>	11
3.5.	<i>Најкраћа путања - проблем 3</i>	13
3.6.	<i>Најкраћа путања из рада [2] - проблем 3</i>	13
3.7.	<i>Најкраћа путања - проблем 4</i>	15
3.8.	<i>Најкраћа путања из рада [3] - проблем 4</i>	15
4.1.	<i>Проблем измене статичких препрека - итерација 50</i>	17
4.2.	<i>Проблем измене статичких препрека - итерација 69</i>	17
4.3.	<i>Проблем измене статичких препрека - итерација 90</i>	18
4.4.	<i>Проблем измене статичких препрека - итерација 120</i>	18
4.5.	<i>Проблем постојања динамичких препрека - итерација 10</i>	19
4.6.	<i>Проблем постојања динамичких препрека - итерација 32</i>	20
4.7.	<i>Проблем постојања динамичких препрека - perrohodan koridor</i>	20
4.8.	<i>Комбиновани динамички проблем - мапа феромона</i>	21
4.9.	<i>Комбиновани динамички проблем - итерација 51</i>	22
4.10.	<i>Комбиновани динамички проблем - итерација 63</i>	23
4.11.	<i>Комбиновани динамички проблем - итерација 64</i>	23

## 1. Увод

Планирање путање представља један од кључних проблема у области аутономних возила и мобилне роботике. Циљ овог процеса је одређивање оптималне руте од почетне до циљне тачке, уз избегавање препрека и поштовање ограничења окружења. Ефикасно планирање путање има директан утицај на безбедност, поузданост и енергетску ефикасност аутономних система.

Алгоритам **Ant Colony Optimization (ACO)**, инспирисан колективним понашањем мрава у природи, један је од познатијих метахеуристичких приступа који се успешно примењује за решавање проблема оптимизације, укључујући и проналазак путање. Традиционални АСО алгоритам показује добре резултате у проналажењу глобално оптималних решења захваљујући механизму феромонске комуникације и хеуристичког претраживања простора. Међутим, у својој основној форми, АСО се суочава са одређеним ограничењима, као што су спора конвергенција, могућност заглављивања у локалним оптимумима и формирање стаза које нису доволно глатке за практичну примену у реалним условима.

У овом раду фокус је на имплементацији и анализи традиционалног АСО алгоритма за проналазак путање аутономног возила. Добијени резултати упоређени су са резултатима из релевантних истраживања из литературе, конкретно радова *Research on path planning of mobile robot based on improved ant colony algorithm [1]* (Xue и Wu, 2021), *Path planning of mobile robot based on improved ant colony algorithm for logistics [2]* (Xue и др., 2021) и *Path planning for autonomous vehicles based on improved ant colony algorithm [3]* (Wen и др., 2025), како би се сагледале перформансе основне верзије алгоритма у односу на приступе које смо применили.

## 2. Опис решења проблема

### 2.1. Традиционални ACO алгоритам

**Ant Colony Optimization** (ACO) је метахеуристички алгоритам инспирисан природним понашањем колоније мрава у процесу тражења најкраћег пута између мравињака и извора хране. Основна идеја алгоритма је симулација колективне интелигенције мрава који, кроз интеракцију путем феромона, постепено граде оптимално решење.

ACO алгоритам се заснива на итеративном процесу у коме мрави граде могућа решења и ажурирају информације о феромону на основу квалитета пронађених путања. Током сваке итерације сваки мрав:

- Креће са почетне тачке.
- Бира следећу тачку (чврор) на основу вероватноће засноване на количини феромона и хеуристичке информације.
- Након што сви мрави изграде своје путање, вредности се ажурирају тако да боље путање добију више феромона.

На овај начин, с временом, мрави колективно конвергирају ка најкраћој (или приближно најкраћој) путањи.

#### 2.1.1. Вероватноћа одабира путање

Када се мрав налази у чврору  $i$  и треба да изабере следећи чврор  $j$ , вероватноћа избора дефинише се формулом:

$$P_{ij}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{k \in N_i} [\tau_{ik}(t)]^\alpha \cdot [\eta_{ik}]^\beta}, & \text{ако } j \in N_i \\ 0, & \text{иначе} \end{cases}$$

где су:

- $(\tau_{ij}(t))$  — количина феромона на грани између чврора  $i$  и  $j$  у тренутку  $t$
- $\eta_{ij} = \frac{1}{d_{ij}}$  — хеуристичка информација (обично обрнута вредност растојања)

- $\alpha$  — параметар који одређује утицај феромона
- $\beta$  — параметар који одређује утицај хеуритичке информације
- $N_i$  — скуп чворова доступних из чвора  $i$

Ова формула моделује компромис између експлоатације (прати се већ позната добра путања) и експлорације (истражују се нове руте).

### 2.1.2. Ажурирање феромона

По завршетку једне итерације, количина феромона на свакој грани ажурира се по следећој формулам:

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t)$$

где су:

- $\rho$  — коефицијент испаравања феромона  $0 < \rho < 1$
- $m$  — број мрава
- $\Delta\tau_{ij}^k(t)$  — количина феромона коју мрав  $k$  оставља на грани  $(i, j)$ , дефинисана као:

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k}, & \text{ако мрав } k \text{ користи грану } (i, j) \\ 0, & \text{иначе} \end{cases}$$

где је  $Q$  константа, а  $L_k$  дужина путање коју је пронашао мрав  $k$ .

На овај начин, краће и боље путање добијају више феромона, што повећава вероватноћу да буду изабране у следећим итерацијама.

## 2.2. Имплементација ACO алгоритма за проблем планирања путање аутономног робота

Иплементиран је традиционалан Ant Colony Optimization (ACO) алгоритам за решавање проблема проналаска путање аутономног возила у дискретном простору. Алгоритам је реализован у програмском језику Python, уз коришћење библиотека NumPy и Matplotlib за нумеричке операције и визуелизацију резултата. Описана имплементација алгоритма може се пронаћи у оквиру GitHub репозиторијума [4].

Простор претраге је представљен као дводимензионални простор (grid) димензије GRID\_SIZE x GRID\_SIZE, где свако поље представља могућу позицију робота. Препреке су дефинисане као скуп координата obstacles, док су START и GOAL фиксне тачке које означавају почетак и крај путање.

Путања сваког мрава моделована је као низ парова координата ( $x,y$ ), при чему се мрав може кретати у осам могућих праваца (четири ортогонална и четири дијагонална корака). За дијагоналне покрете коришћен је повећани трошак од 1.41, што симулира реалну еуклидску дистанцу.

Избор следећег чвора врши се случајним одабиром, притом не узимајући у обзир претходно посећене чворове. Тада избор зависи како од количине феромона, тако и од дистанци од тренутног до наредних чворова. Однос између ове две карактеристике, односно колико која утиче на одлуку, моделује се параметрима  $\alpha$  и  $\beta$ .

Након сваке итерације, када последњи мрав одреди своју путању, ажурирају се вредности феромона. Најпре се, глобално, вредност феромона смањује за фактор испаравања, а након тога се вредност феромона на пољима која су посећена повећава вредност феромона. Вредност феромона се повећава у складу са дужином остварене путање, где се већа количина феромона додаје уколико је путања краћа.

У спроведеној имплементацији коришћени су следећи параметри:

- број итерацијама
- број мрава
- параметри  $\alpha$  и  $\beta$
- коефицијент испаравања феромона  $\rho$
- коефицијент количине феромона  $Q$

Алгоритам се извршава фиксни број итерација, при чему се након сваке итерације чува најкраћа пронађена путања. На крају извршавања врши се визуелизација најбоље путање у односу на препреке и крајњу тачку.

### 2.2.1. Функција циља

Функција циља дефинисана је као минимизација укупне дужине путање, при чему се укупни трошак израчунава као збир свих корака које мрав направи:

$$L_k = \sum_{(i,j) \in path_k} d_{ij}$$

Циљ алгоритма је проналажење путање са минималним  $L_k$  (односно са најмањом вредношћу `best_cost`). Најбоља пронађена путања се чува као `best_path`, а њен трошак као `best_cost`. На крају сваке итерације, уколико је пронађена краћа путања, она се исписује у конзоли.

### 3. Експериментални резултати

У овом поглављу биће приказани и анализирани експериментални резултати добијени применом АСО алгоритма. Затим су та решења упоређена са резултатима објављеним у литератури за исте експерименталне сценарије. На овај начин омогућено је вредновање квалитета, стабилности и ефикасности имплементације, као и сагледавање њене позиције у односу на постојећа решења.

#### 3.1. Проблем 1

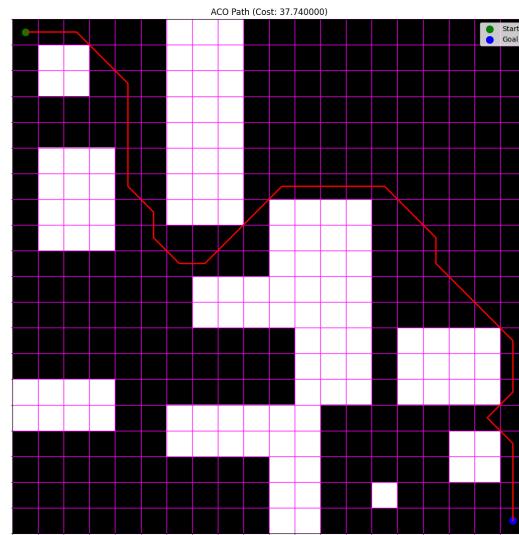
У циљу објективног вредновања ефикасности имплементације, алгоритам је тестиран на истом проблему и уз исте параметре као у референтном научном раду [1]. Поставке као што су број мрава, коефицијент испарања феромона и број итерација, преузети су директно из описаног експерименталног окружења, како би поређење било методолошки оправдано. Коефицијенти  $\alpha$  и  $\beta$  се разликују и износе 2 и 1 респективно. Алгоритам је покренут 10 пута, а у табели 3.1 представљени су резултати.

Табела 3.1. Резултати тестирања проблем 1

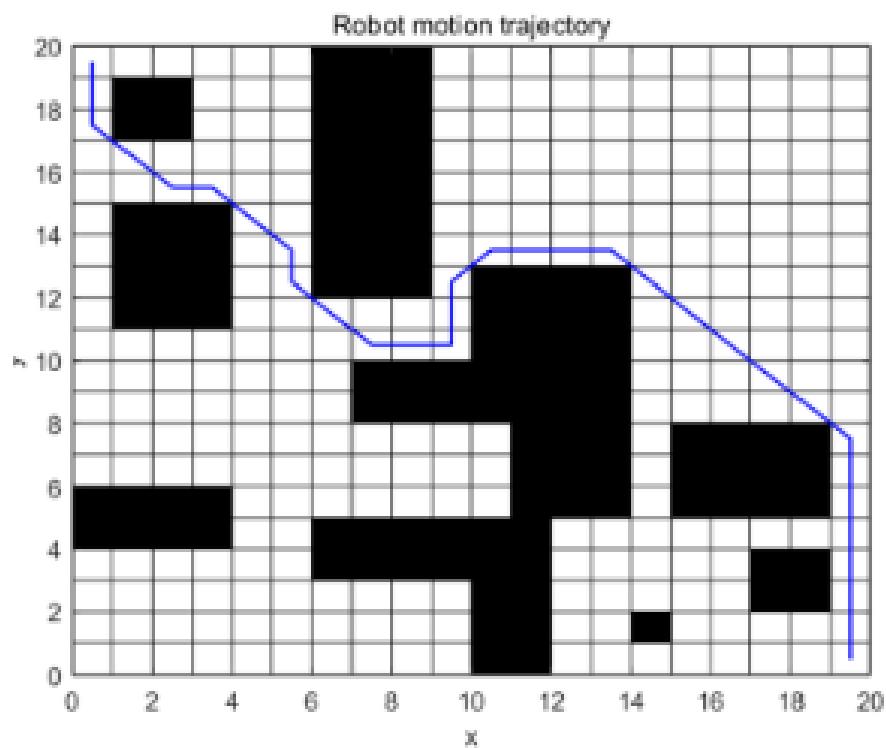
Алгоритам	Просечна дужина путање	Најбоља путања	Просечан број итерација
Алгоритам из рада [1]	36,30196	34,3848	10,2
Алгоритам из овог рада	42,46	37,74	128,0

На основу приказаних резултата може се уочити да имплементација алгоритма из овог рада показује спорију конвергенцију и нешто лошији квалитет решења у односу на референтну верзију. Просечна дужина путање и најбоља пронађена путања су веће, док је број итерација потребан да би се достигло најбоље решење значајно већи (128 у односу на 10,2). Ово указује да алгоритам из литературе ефикасније усмерава мраве ка оптималнијим решењима и брже стабилизује феромонску мапу.

Најкраћа пронађена путања постигнута у 128. итерацији и износи 37,74, приликом једног покретања, приказана је на слици 3.1, док је на слици 3.2 приказана најбоља пронађена путања са којом се упоређује.



Слика 3.1. Најкраћа путања проблем 1



Слика 3.2. Најкраћа путања из рада [1] проблем 1

### 3.2. Проблем 2

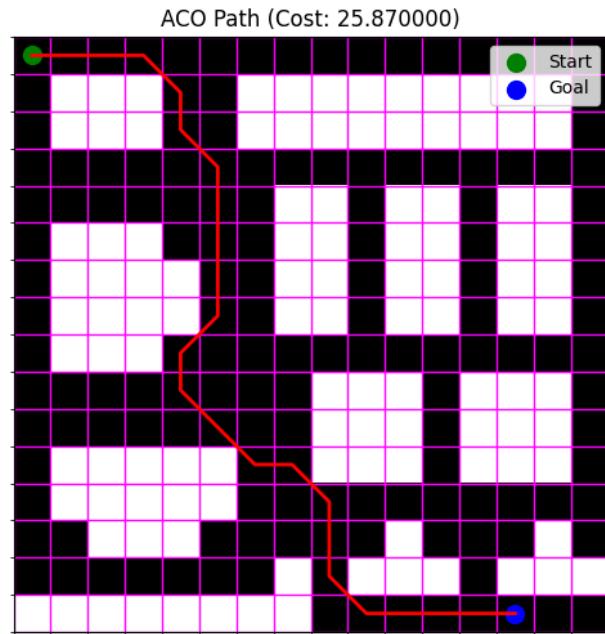
У циљу објективног вредновања ефикасности имплементације, алгоритам је тестиран на истом проблему и уз исте параметре као у референтном научном раду [2]. Поставке као што су број мрава, коефицијент исправавања феромона и број итерација, преузети су директно из описаног експерименталног окружења, како би поређење било методолошки оправдано. Коефицијенти  $\alpha$  и  $\beta$  се разликују и износе 2 и 1 респективно. Алгоритам је покренут 10 пута, а у табели 3.2 представљени су резултати.

Табела 3.2. Резултати тестирања проблем 2

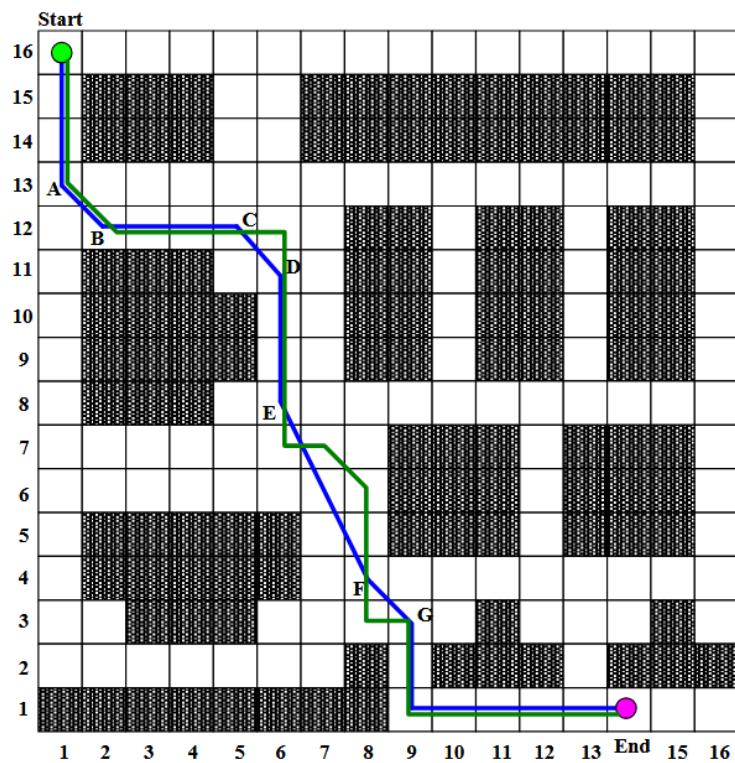
Алгоритам	Просечна дужина путање	Најбоља путања	Просечан број итерација
Алгоритам из рада [2]	28,14	24,53	61
Алгоритам из овог рада	28,11	25,87	9,67

На основу добијених резултата може се уочити да обе имплементације постижу сличан просечан квалитет путање (28,14 и 28,11), што указује на то да предложени алгоритам даје конкурентна решења у погледу дужине пронађене путање. Међутим, разлика је приметна када се посматра најбоља пронађена путања и динамика конвергенције. Референтни алгоритам је постигао нешто бољу минималну вредност путање (24.53 у односу на 25.87), али је за то било потребно знатно више итерација (61 уместо 9.67).

Најкраћа пронађена путања постигнута у 5. итерацији и износи 25,87, приликом једног покретања, приказана је на слици 3.3, док је на слици 3.4 приказана најбоља пронађена путања са којом се упоређује.



Слика 3.3. Најкраћа путања - проблем 2



Слика 3.4. Најкраћа путања из рада [2] - проблем 2

### 3.3. Проблем 3

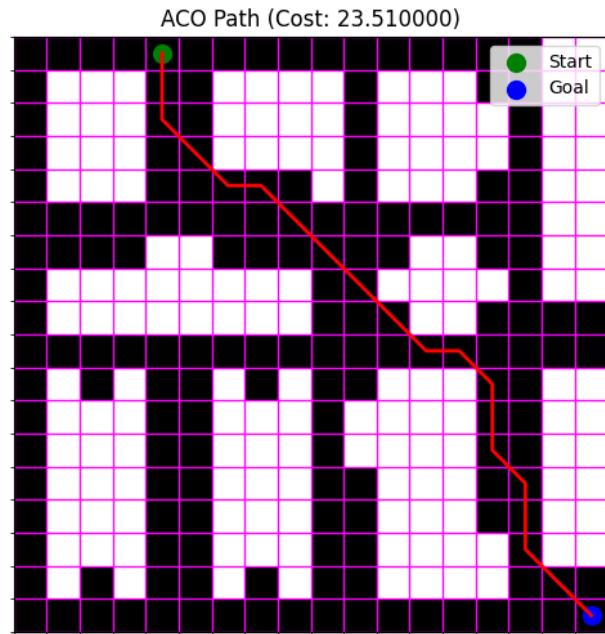
У циљу објективног вредновања ефикасности имплементације, алгоритам је тестиран на истом проблему и уз исте параметре као у референтном научном раду [2]. Поставке као што су број мрава, коефицијент исправавања феромона и број итерација, преузети су директно из описаног експерименталног окружења, како би поређење било методолошки оправдано. Коефицијенти  $\alpha$  и  $\beta$  се разликују и износе 2 и 1 респективно. Алгоритам је покренут 10 пута, а у табели 3.3 представљени су резултати.

Табела 3.3. Резултати тестирања проблем 3

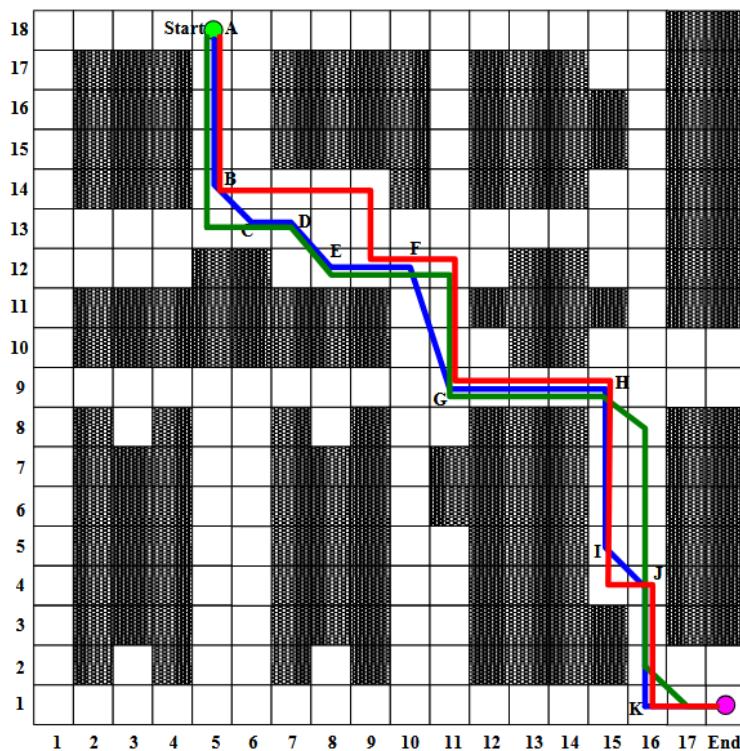
Алгоритам	Просечна дужина путање	Најбоља путања	Просечан број итерација
Алгоритам из рада [2]	29,14	25,42	72
Алгоритам из овог рада	26,69	23,51	32,4

На основу резултата приказаних у табели може се уочити да имплементација из овог рада остварује бољу најбољу пронађену путању (23,51 у односу на 25,42), као и мању просечну дужину путање (26,69 уместо 29,14). То значи да алгоритам у просеку и у најповољнијем случају проналази краће и ефикасније путање. Такође просечан број итерација који је потребан да се дође до оптималног решења је знатно мањи и алгоритам брже конвергира.

Најкраћа пронађена путања постигнута у 45. итерацији и износи 25,51, приликом једног покретања, приказана је на слици 3.5, док је на слици 3.6 приказана најбоља пронађена путања са којом се упоређује.



Слика 3.5. Нажкраћа путања - проблем 3



Слика 3.6. Нажкраћа путања из рада [2] - проблем 3

### 3.4. Проблем 4

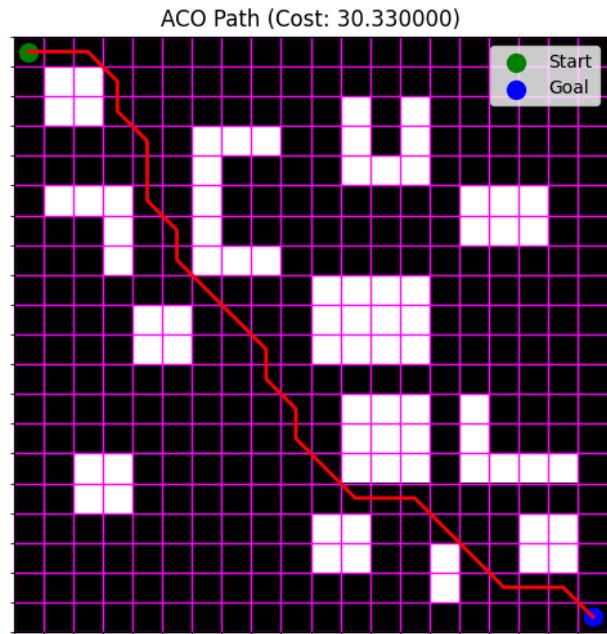
У циљу објективног вредновања ефикасности имплементације, алгоритам је тестиран на истом проблему и уз исте параметре као у референтном научном раду [3]. Поставке као што су број мрава, коефицијент исправавања феромона и број итерација, преузети су директно из описаног експерименталног окружења, како би поређење било методолошки оправдано. Коефицијенти  $\alpha$  и  $\beta$  се разликују и износе 2 и 1 респективно. Алгоритам је покренут 10 пута, а у табели 3.4 представљени су резултати.

**Табела 3.4.** Резултати тестирања проблем 4

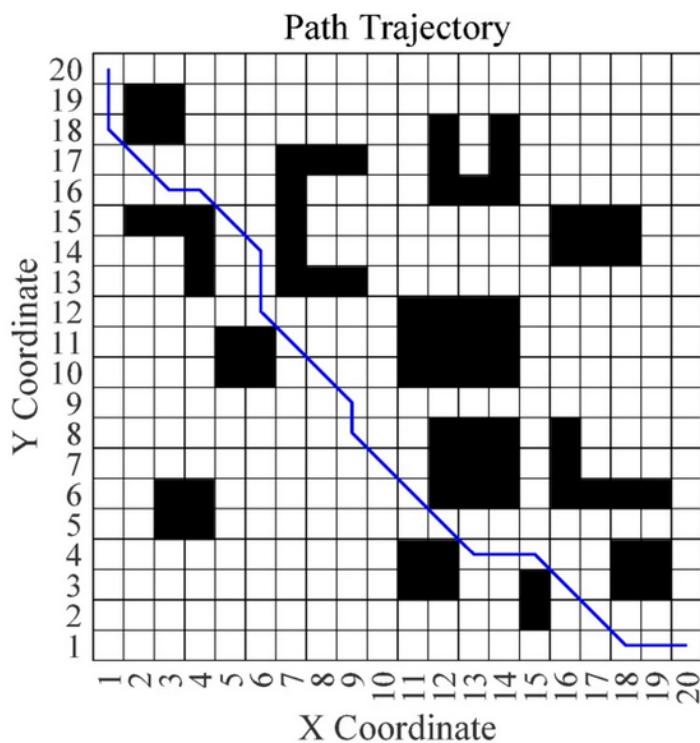
Алгоритам	Просечна дужина путање	Најбоља путања	Просечан број итерација
Алгоритам из рада [3]	?	29,8	16
Алгоритам из овог рада	32,98	30,330000000000002	142,33

Најбоља пронађена путања је упоредива (30,33 у односу на 29,8), а узевши у обзир просечну дужину путање коју која износи 32,98 примећује се да алгоритам конзистентно остварује добра решења. С друге стране, број итерација потребан за постизање најбољег решења је значајно већи у нашем алгоритму (142,33 у односу на 16), што показује спорију конвергенцију. Ово је очекивано, јер се ради о основној имплементацији АСО алгоритма без додатних механизама који побољшавају брзину конвергенције.

Најкраћа пронађена путања постигнута у 268. итерацији и износи 30,33, приликом једног покретања, приказана је на слици 3.7, док је на слици 3.8 приказана најбоља пронађена путања са којом се упоређује.



Слика 3.7. Најкраћа путања - проблем 4



Слика 3.8. Најкраћа путања из рада [3] - проблем 4

## 4. Експериментални резултати (динамичко окружење)

У овом поглављу биће приказани резултати експерименталног испитивања традиционалног АСО алгоритма у условима динамичког окружења. За разлику од статичких сценарија, у којима положај препрека остаје непромењен током процеса планирања путање, динамичко окружење подразумева промену конфигурације радног простора током итерација.

У оквиру спроведених испитивања разматрана су два типа промена:

1. **измене статичких препрека** – симулација случајева у којима се појављују или нестају фиксне препреке током рада алгоритма (нпр. померање кутија, отварање или затварање пролаза), и
2. **постојање динамичких препрека** које мењају свој положај у реалном времену, као што су други мобилни роботи или пешаци.

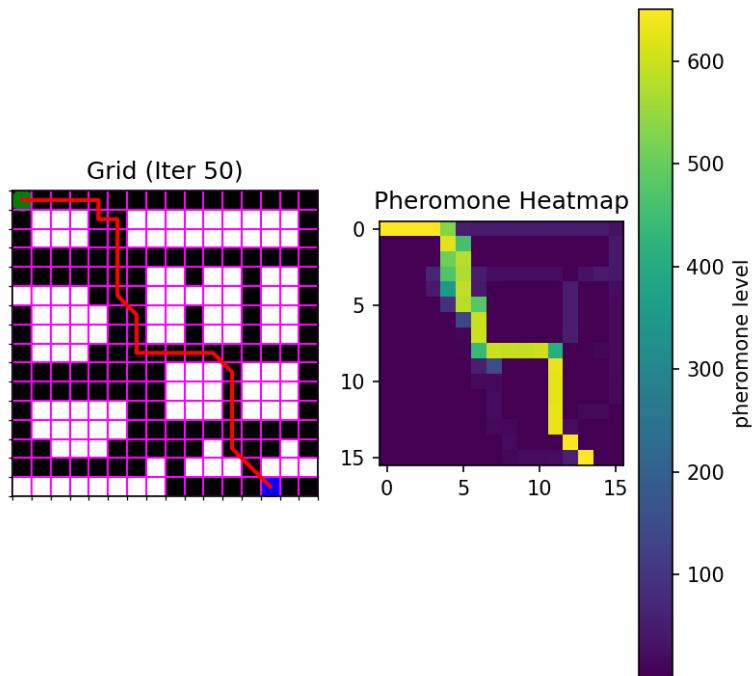
Циљ ових испитивања је да се оцени способност традиционалног АСО алгоритма да се адаптира на промене у окружењу, задржавајући прихватљиво време конвергенције, дужину путање и способност избегавања судара.

### 4.1. Проблем измене статичких препрека

У првом сценарију разматран је случај у коме се током извршавања алгоритма мења структура окружења додавањем или уклањањем одређених статичких препрека. Овај тип промене симулира реалне услове у којима робот делује у простору који није у потпуности познат унапред, као што је, на пример, складишни простор у коме се померају кутије или се отварају и затварају пролази.

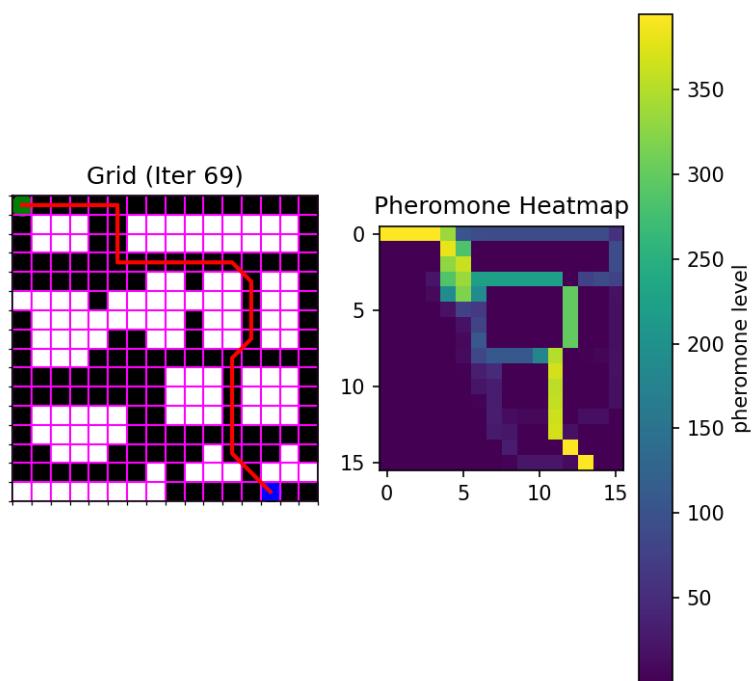
На почетку експеримента, АСО алгоритам је конверговао ка оптималној путањи у складу са иницијалном конфигурацијом препрека, што је приказано на слици 4.1. Већ у следећој итерацији препреке су постављене на позицијама кроз које приказана путања пролази, алгоритам се брзо прилагођава новонасталој ситуацији и у неколико итерација проналази нову оптималну путању приказану на слици 4.2. Након тога, у итерацији број 70, уклањају се претходно постављене препреке које су онемогућиле пролазак до циља путањом оствареном у итерацији број 50. Алгоритам се брзо прилагођава и поново проналази оптималну путању, минимално бољу у односу на путању остварену у итерацији број 50. Та путања приказана је на слици 4.3. Коначна оптимална путања, минимално боља у односу на претходно остварене, достиже се у 120 итерацији алгоритма и приказана је на слици 4.4.

Iteration 50 — Best Cost: 25.64



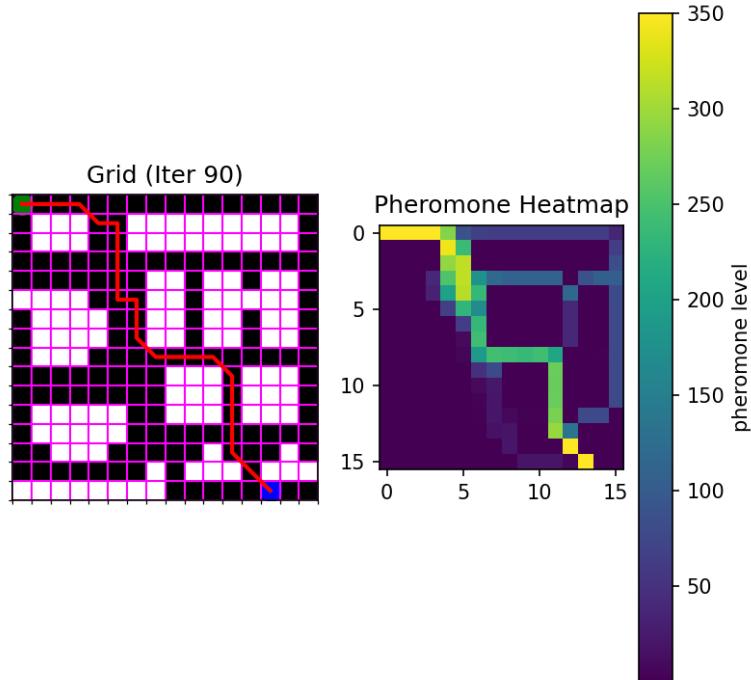
Слика 4.1. Проблем измене статичких препрека - итерација 50

Iteration 69 — Best Cost: 27.64



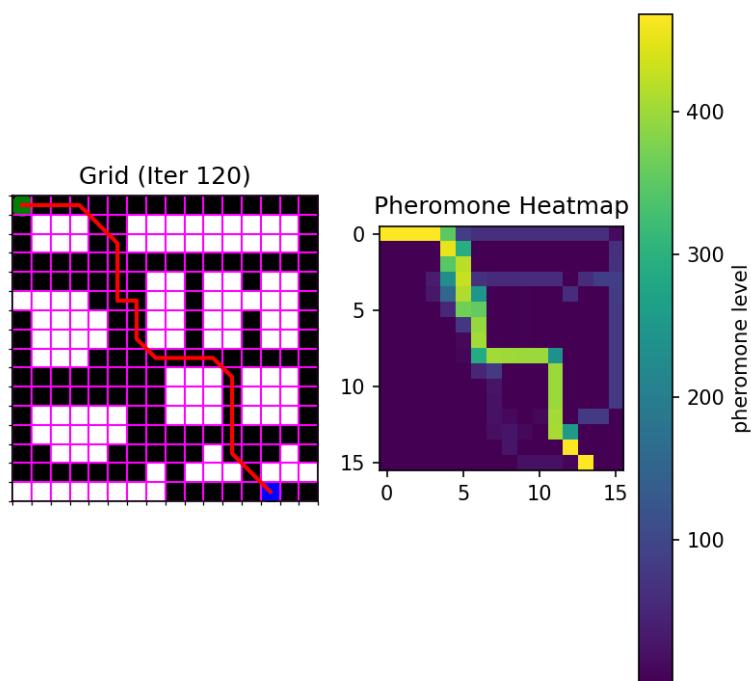
Слика 4.2. Проблем измене статичких препрека - итерација 69

Iteration 90 — Best Cost: 25.05



Слика 4.3. Проблем измене статичких препрека - итерација 90

Iteration 120 — Best Cost: 24.46



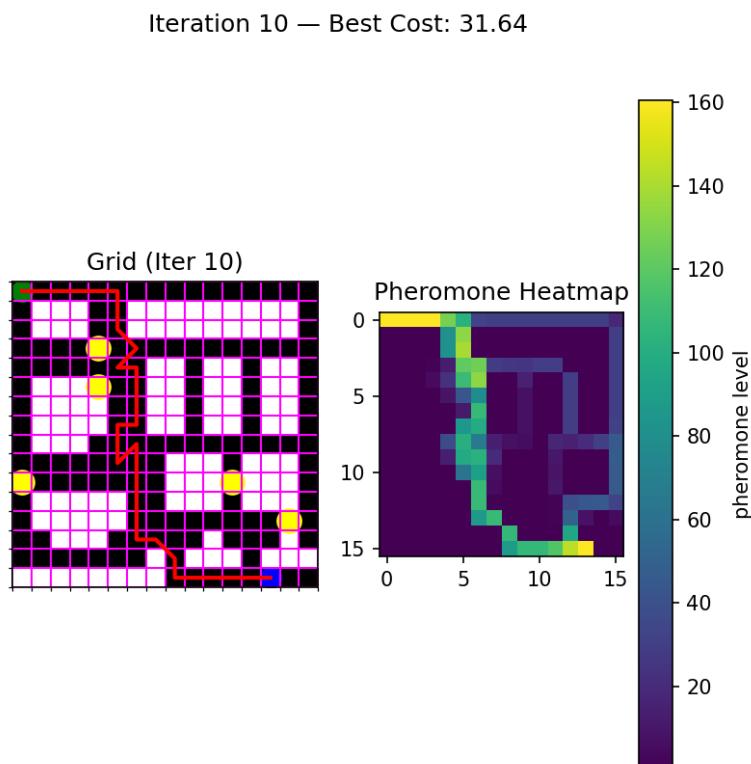
Слика 4.4. Проблем измене статичких препрека - итерација 120

## 4.2. Проблем постојања динамичких препрека

Други експериментални сценарио односи се на присуство покретних препрека које мењају свој положај током времена. Овај случај је значајан за реалне примене као што су координација више мобилних платформи или кретање у окружењу са људима, где се конфигурација простора константно мења.

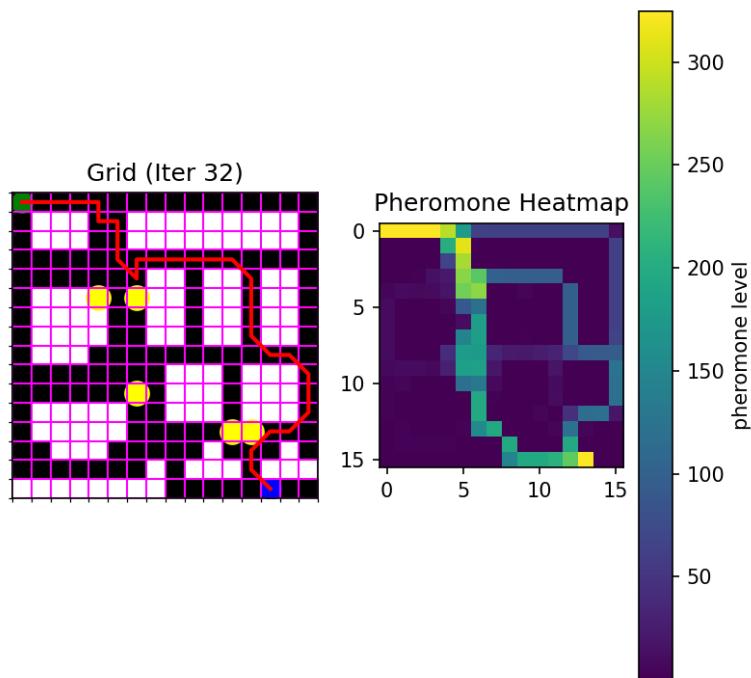
У току симулације више препрека су се кретале унапред дефинисаним путањама, а на сликама су приказане жутом бојом. На слици 4.5 приказана је путања пронађена након десет итерација алгоритма, која пролази средишњи простора који у том тренутку није окупирају динамичким препрекама. Међутим, већ неколико итерација касније, на простору кроз који та путања пролази, налази се већи број динамичких препрека. У тој ситуацији алгоритам проналази алтернативну путању приказану на слици 4.6. У даљем раду алгоритма, с обзиром на репетитивно кретање динамичких препрека, током итерација као најкраћа путања биће нека варијација једне од две претходно поменуте путање.

Значајно је приметити да претходно остварена путања приказана на слици 4.4 није достижна у условима задатим овим сетом динамичких препрека, због окупирањости коридора истакнутог на слици 4.7

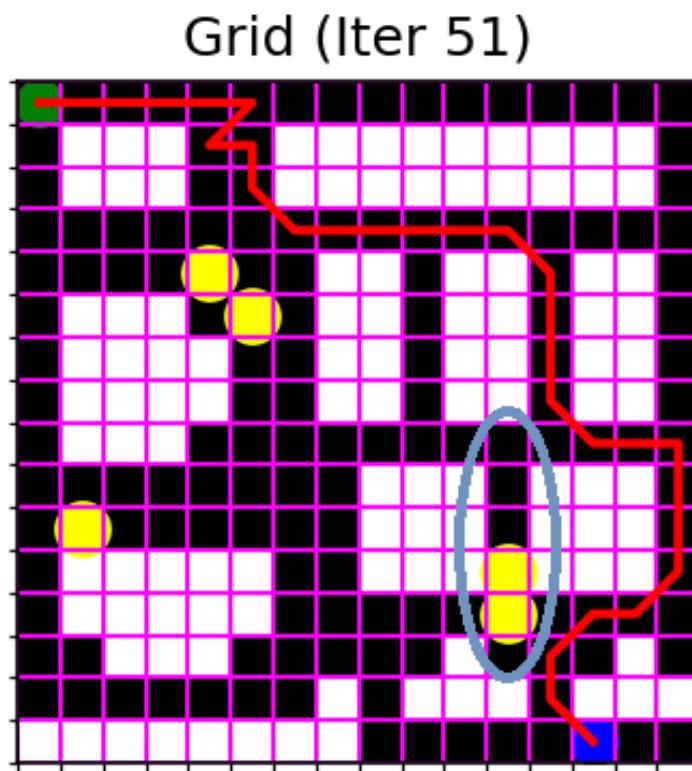


Слика 4.5. Проблем постојања динамичких препрека - итерација 10

Iteration 32 — Best Cost: 31.87



Слика 4.6. Проблем постојања динамичких препрека - итерација 32



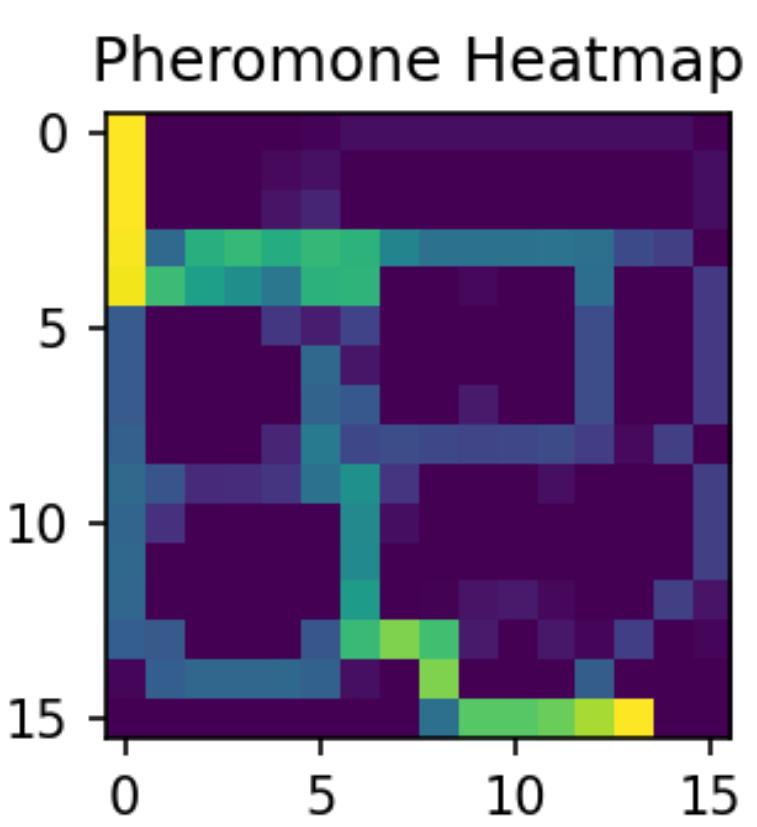
Слика 4.7. Проблем постојања динамичких препрека - непротоковани коридор

Ови резултати потврђују да чак и без додатних механизама адаптације, традиционални АСО поседује одређени ниво робусности у динамичким условима, што га чини погодном основом за развој напреднијих, хибридних приступа при планирању путање у реалном времену.

#### 4.3. Комбиновани динамички проблем

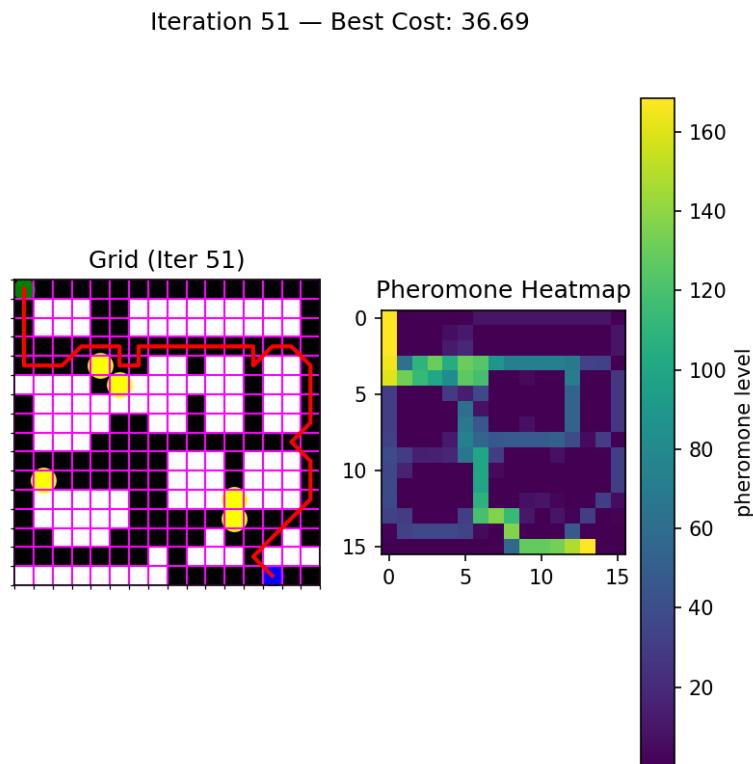
У најзахтевнијем сценарију разматране су истовремене измене статичких препрека и присуство покретних објекта, што представља реалистичан модел високо динамичког окружења. Оваква конфигурација значајно повећава сложеност проблема, јер се оптимална путања не мења само повремено, већ готово непрекидно, у зависности од интеракције статичких и динамичких фактора.

У оваквим условима АСО алгоритам показује знатно успорену конвергенцију. Како се у свакој или скоро свакој итерацији мења структура окружења, мрави често губе тек формиране трасе, јер се делови потенцијално оптималних путева повремено блокирају или постају ризични. На приказаној мапама феромона на слици 4.8 примећује се да је феромон рас прострањен свуда по простору претраге што је јасан доказ да мрави вишевремена проводе у истраживању него у експлоатацији.



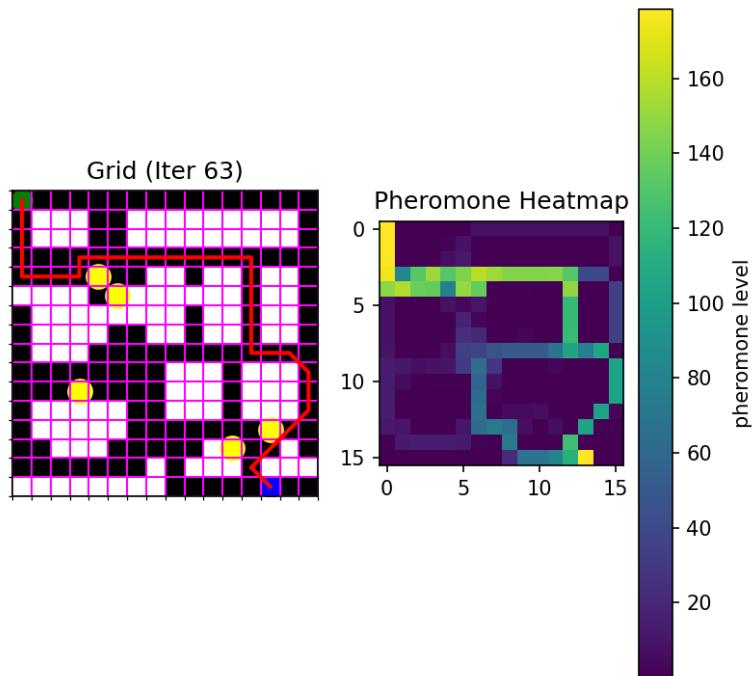
Слика 4.8. Комбиновани динамички проблем - мапа феромона

Последица јесте да услед повећане тежине проблема и учесалих промена у окружењу, мрави имају доста потешкоћа да пронађу стабилну, поуздану путању. Међутим, када открију пролаз који је у датом тренутку проходан, алгоритам има тенденцију да тај пут задржава, односно да му додељује значајно више феромона него осталим алтернативама. То доводи до привремене доминације једне трасе, али се она често показује кратког века, нарочито када покретна препрека накнадно пресече управо тај пролаз. Као што се може приметити на сликама 4.9, 4.10 и 4.11.



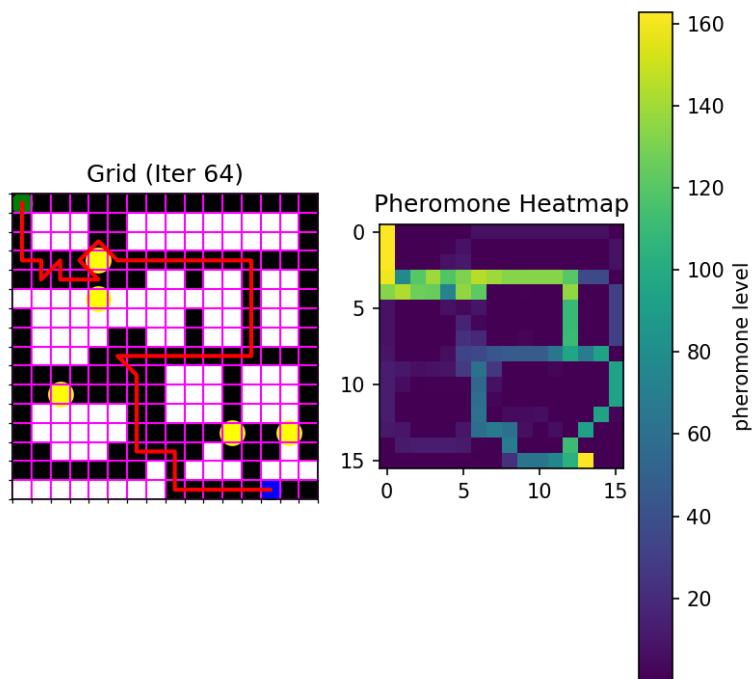
Слика 4.9. Комбиновани динамички проблем - итерација 51

Iteration 63 — Best Cost: 33.05



Слика 4.10. Комбиновани динамички проблем - итерација 63

Iteration 64 — Best Cost: 47.05



Слика 4.11. Комбиновани динамички проблем - итерација 64

Иако решење и даље постоји у свакој итерацији, ACO алгоритам овде слабо одржава стабилност, а процес поновне конвергенције је много спорији него у сценаријима са умереним променама. Ови резултати указују да је у високо динамичким условима неопходно увођење додатних механизама адаптације, попут бржег исправљања феромона, локалног ресетовања након промене окружења или интеграције реактивних метода избегавања препрека.

## 5. Закључак

У овом раду представљена је имплементација традиционалног ACO (Ant Colony Optimization) алгоритма за решавање проблема проналажења путање. Алгоритам је тестиран на више експерименталних сценарија и резултати су упоређени са подацима из релевантне литературе.

Анализа резултата показује да предложена имплементација доследно проналази квалитетна решења, али уз спорију конвергенцију и понекад нешто веће просечне дужине путање у односу на референтне алгоритме. Ово је очекивано, с обзиром на то да се ради о основној верзији ACO алгоритма, без додатних механизама за убрзавање конвергенције или побољшање претраживања простора решења.

У динамичким сценаријима који укључују измене статичких препрека, присуство покретних објекта и њихову комбинацију, алгоритам такође демонстрира способност адаптације. Резултати показују да ACO релативно успешно прати промене окружења, постепено мењајући структуру феромонске мапе и проналазећи нове проходне трасе. Ипак, у условима честих или наглих промена, уочава се успорена реакција и веће ослањање на већ једном пронађену проходну путању, што доводи до успорене конвергенције и нестабилности решења у поређењу са резултатима добијеним у експериментима са статичким окружењем.

Упркос тим ограничењима, експериментални резултати потврђују исправност и стабилност имплементације, као и способност алгоритма да у различитим условима проналази прихватљива решења. Резултати такође пружају добру основу за будућа побољшања, као што су увођење адаптивних параметара, локалних оптимизација или хибридних метода, с циљем да се постигну боља решења уз краће време извршавања.

## 6. Литература

- [1] Chenglong Xue, Jian Wu, *Research on path planning of mobile robot based on improved ant colony algorithm*, School of Mechanical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China
- [2] Path planning of mobile robot based on improved ant colony algorithm for logistics  
Tian Xue, Liu Li, Liu Shuang, Du Zhiping and Pang Ming  
Доступно: <https://www.aimspress.com/data/article/preview/pdf/6062fe4dba35de1a1fbde4ce.pdf>  
Приступљено: 28. октобар 2025.
- [3] Path planning for autonomous vehicles based on the improved ant colony algorithm for logistics  
Guoqiang Wen, Zhiwei Guan, Hongxia Zhang, Rui Liu, Shuntang Hu and Ruzhen Dou  
Доступно: <https://www.nature.com/articles/s41598-025-20120-8>  
Приступљено: 28. октобар 2025.
- [4] ACO\_robot\_path\_plannig Repo  
Доступно: [https://github.com/mark0marko/ACO\\_robot\\_path\\_plannig](https://github.com/mark0marko/ACO_robot_path_plannig)  
Приступљено: 28. октобар 2025.