Dinamičko koordiniranje dodjele zadatka multi-robotskim sustavima u pametnim prostorima

Matezović, Mislav

Master's thesis / Diplomski rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:235:749648

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

Repository / Repozitorij:

Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mislav Matezović

Zagreb, 2020. godina

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mentori:
Doc. dr. sc. Marko Švaco, dipl. ing.
Izv. prof. dr. sc. Marin Lujak, dipl. ing.

Student:
Mislav Matezović
0035197063

Zagreb, 2020. godina

Izjavljujem da sam ovaj rad izradio samostalno koristeći stečena znanja tijekom studija i navedenu literaturu.

Zahvaljujem se mentoru, doc. dr. sc. Marku Švaci, na stručnim savjetima i pruženoj pomoći pri izradi ovoga rada.

Također se zahvaljujem komentoru izv. prof. dr. sc. Marinu Lujaku s IMT Lille Doauija na stručnim savjetima i pruženoj pomoći tokom izrade ovoga rada.

Mislav Matezović

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE
Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za diplomske radove studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment,
inženjerstvo materijala te mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum:	Prilog:
Klasa: 602 - 04 / 20 - 6 / 3	
Ur. broj: 15 - 1703 - 20 -	

DIPLOMSKI ZADATAK

Student: **MISLAV MATEZOVIĆ** Mat. br.: 0035197063

Naslov rada na hrvatskom jeziku: **Dinamičko koordiniranje dodjele zadataka multi-robotskim sustavima u pametnim prostorima**

Naslov rada na engleskom jeziku: **Dynamic coordination of multi-robot task assignment in smart spaces**

Opis zadatka:

Stanovništvo zapadnog svijeta stari. Sve veći broj starijih osoba sa smanjenom pokretljivošću treba stalnu pomoć u svakodnevnim zadacima. Koncept pametnih prostora odgovara potrebama ovog trenda i uključuje praćenje i nadzor korisnika. Međutim, taj koncept je ograničen samo na usluge koje ne uključuju fizičku interakciju s korisnikom. Korištenje mobilne robotike je obećavajuće i održivo rješenje u ovom smjeru. Konkretno, timovi mobilnih robota mogu intervenirati i doprinijeti nizu svakodnevnih ljudskih aktivnosti koje se ponavljaju i čija provedba ovisi o jednostavnim lokomotornim sposobnostima (npr. isporuke lijekova ili obroka). Neovisno o infrastrukturi (dom, bolnica, domovi umirovljenika itd.), istodobna primjena više robota u istom prostoru temelji se na njihovoj dinamičkoj koordinaciji u stvarnom vremenu s obzirom na razvoj događaja pri izvršavanju zadataka, njihov ograničen komunikacijski domet i nesavršenu komunikacijsku mrežu.

Zadaci ovog diplomskog rada su sljedeći:
1. Dati pregled distribuiranih pristupa problemu dodjele zadataka robotskim sustavima (po principu jedan zadatak svakom robotu) (MRTA) uzimajući u obzir ograničen komunikacijski domet robota.
2. Opisati potrebe robotskog sustava i odrediti modalitete koordinacije grupe robota na temelju komunikacijskog raspona (povezan i nepovezan graf komunikacije).
3. Razviti MRTA koordinacijski model za suočavanje s ograničenim dometom komunikacije i rezultirajućim nepovezanim grafom komunikacije.
4. Primijeniti model koordinacije MRTA u Matlabu u pojednostavnjenom simulatoru dostupnom na Institutu IMT Lille Douai
5. Definirati postavke eksperimenata i prikazati rezultate simulacije u Matlab simulatoru s naglaskom na prijeđene udaljenosti i broj razmijenjenih poruka.
6. Emulirati MRTA koordinacijski model na dva fizička robota u laboratoriju robotike IMT-a Lille Douai.

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:	Rok predaje rada:	Predviđeni datum obrane:
5. ožujka 2020.	7. svibnja 2020.	11. svibnja do 15. svibnja 2020.

Zadatak zadao: doc. dr. sc. Marko Švaco

Komentor: Assoc. Prof. Marin Lujak, PhD

IMT LILLE DOUAI
Site de Douai
941 rue Charles Bourseul
CS 10838
59508 DOUAI Cédex
Tél. 03 27 71 22 22 - Fax 03 27 71 25 25

Predsjednica Povjerenstva: prof. dr. sc. Biserka Runje

SADRŽAJ

SADRŽAJ . . . . . **I**

POPIS SLIKA . . . . . **II**

POPIS TABLICA . . . . . **IV**

POPIS OZNAKA . . . . . **V**

SAŽETAK . . . . . **VI**

SUMMARY . . . . . **VII**

1. Uvod **1**

2. Dodjela zadataka robotskim sustavima **3**

3. Rješavanje problema dodjele zadataka **5**
   - 3.1 Lekisografski algoritam 5
   - 3.2 FCFS algortihm 10
   - 3.3 Mađarski algoritam 11
   - 3.4 Nepotpuna komunikacija 13
   - 3.5 Aukcijski algoritam 15

4. Simulacije **17**
   - 4.1 Simulacije za potpunu komunikaciju 18
   - 4.2 Simulacije za nepotpunu komunikaciju 27

5. Nastavak i nadogradnja radnje **42**

6. Zaključak **44**

POPIS SLIKA

Slika 1. Prikaz idejnog sustava za uvođenje mobilnih robota u pametne prostorije . 2
Slika 2. Primjer bipartitnog grafa 4
Slika 3. Graphical representation of the first quadrant 7
Slika 4. Prikaz postupka dodjele zadataka kod mađarskog algoritma 12
Slika 5. Mobilni robot s komunikacijskim dometom 14
Slika 6. Međusobna komunikacija dvaju robota 14
Slika 7. Skokovita metoda komunikacije 14
Slika 8. Primjer nasumično generiranih robota i zadataka 18
Slika 9. Rezultat leksikografskog algoritma 19
Slika 10. Rezultat FCFS algoritma 19
Slika 11. Rezultat mađarskog algoritma 20
Slika 12. Prijeđene udaljenosti kod potpune komunikacije 21
Slika 13. Minimalne prijeđene udaljenosti kod potpune komunikacije 23
Slika 14. Maksimalne prijeđene udaljenosti kod potpune komunikacije 24
Slika 15. Prosječne prijeđene udaljenosti kod potpune komunikacije 26
Slika 16. Prijeđena udaljenost kod nepotpune komunikacije - po komunikacijskom dometu 28
Slika 17. Broj iteracija kod nepotpune komunikacije - po komunikacijskom dometu . 28
Slika 18. Normirane udaljenosti kod nepotpune komunikacije - po komunikacijskom dometu 31
Slika 19. Prijeđene udaljenosti kod nepotpune komunikacije - po broju robota 35
Slika 20. Broj iteracija kod nepotpune komunikacije - po broju robota 35
Slika 21. Normirane prijeđene udaljenosti kod nepotpune komunikacije - po broju robota 39
Slika 22. TurtleBot mobilni robot 43
Slika 23. Primjer programa u ROS-u 43

POPIS TABLICA

Tablica 1. Tablica dodjele zadatka za primjer na slici 4 12
Tablica 2. Parovi mobilnih robota i zadataka za primjer sa slike 8 20
Tablica 3. Tablica prijeđenih udaljenosti kod potpune komunikacije 22
Tablica 4. Tablica minimalnih prijeđenih udaljenosti kod potpune komunikacije 23
Tablica 5. Tablica maksimalnih prijeđenih udaljenosti kod potpune komunikacije 25
Tablica 6. Tablica prosjčnih prijeđenih udaljenosti kod potpune komunikacije 26
Tablica 7. Tablica prijeđenih udaljenosti i broja iteracija za 10 robota kod nepotpune komunikacije 29
Tablica 8. Tablica prijeđenih udaljenosti i broja iteracija za 20 robota kod nepotpune komunikacije 29
Tablica 9. Tablica prijeđenih udaljenosti i broja iteracija za 30 robota kod nepotpune komunikacije 30
Tablica 10. Tablica prijeđenih udaljenosti i broja iteracija za 40 robota kod nepotpune komunikacije 30
Tablica 11. Tablica prijeđenih udaljenosti i broja iteracija za 50 robota kod nepotpune komunikacije 31
Tablica 12. Tablica normiranih prijeđenih udaljenosti za 10 robota kod nepotpune komunikacije 32
Tablica 13. Tablica normiranih prijeđenih udaljenosti za 20 robota kod nepotpune komunikacije 32
Tablica 14. Tablica normiranih prijeđenih udaljenosti za 30 robota kod nepotpune komunikacije 33
Tablica 15. Tablica normiranih prijeđenih udaljenosti za 40 robota kod nepotpune komunikacije 33
Tablica 16. Tablica normiranih prijeđenih udaljenosti za 50 robota kod nepotpune komunikacije 34
Tablica 17. Tablica prijeđenih udaljenosti i broja iteracija za domet 0 36

Tablica 18. Tablica prijeđenih udaljenosti i broja iteracija za domet 10 36
Tablica 19. Tablica prijeđenih udaljenosti i broja iteracija za domet 20 36
Tablica 20. Tablica prijeđenih udaljenosti i broja iteracija za domet 30 37
Tablica 21. Tablica prijeđenih udaljenosti i broja iteracija za domet 40 37
Tablica 22. Tablica prijeđenih udaljenosti i broja iteracija za domet 50 38
Tablica 23. Tablica prijeđenih udaljenosti i broja iteracija za domet 60 38
Tablica 24. Tablica prijeđenih udaljenosti i broja iteracija za domet 70 38
Tablica 25. Tablica normiranih prijeđenih udaljenosti za domet 0 40
Tablica 26. Tablica normiranih prijeđenih udaljenosti za domet 10 40
Tablica 27. Tablica normiranih prijeđenih udaljenosti za domet 20 40
Tablica 28. Tablica normiranih prijeđenih udaljenosti za domet 30 41
Tablica 29. Tablica normiranih prijeđenih udaljenosti za domet 40 41
Tablica 30. Tablica normiranih prijeđenih udaljenosti za domet 50 41
Tablica 31. Tablica normiranih prijeđenih udaljenosti za domet 60 42
Tablica 32. Tablica normiranih prijeđenih udaljenosti za domet 70 42

POPIS OZNAKA

A	Matrica dodjele zadataka
α	Kut između pozicija zadatka i robota
C	Matrica troška
c_{ij}	Element matrica troška
D	Matrica udaljenosti
E	Skup rubova
e	Element u skupu rubova
G	Bipartitni graf
γ_i	Razlika cijena
i	Element u skupu robota
j	Element u skupu zadataka
n	Broj robota i zadtaka
p_j	Cijena zadatka
R	Set robota
S	Vektor koraka
T	Set zadataka
v_i	Zadatak s najboljom cijenom
w_i	Zadatak s drugom najboljom cijenom
x_{ij}	Upravljačka varijabla

SAŽETAK

Mobilni roboti mogu asistirati osoblju bolnice ili doma za starije osobe obavljanjem zadataka poput donošenja lijekova ili hrane. Za njihov uspješan rad potrebno je pravilno raspodijeliti zadatke između njih. Kroz ovaj rad pokazat će se problem dodjele zadataka robotskim sustavima. Zatim će se pokazati četiri algoritma za rješavanje problema dodjele zadataka. Prvo će se primijeniti leksikografski algoritam koji nema metodu odlučivanja, već leksikografski dodjeljuje zadatke. Nakon toga, će se primijeniti First come first serve algoritam koji dodjeljuje zadatke uzimajući u obzir minimalne udaljenosti dostupnih zadataka. Potom, će se primijeniti mađarski algoritam koji optimizira ukupne udaljenosti. Ova tri algoritma primjenjivat će potpunu komunikaciju. Kako bi se prikazao utjecaj nepotpune komunikacije koristi će se aukcijski algoritam. Prikazat će se kako vrijednosti komunikacijskog dometa i broja robota utječu na rezultate.

Ključne riječi: problem dodjele zadataka robotskim sustavim, leksikografski algoritam, First come first serve algoritam, mađarski algoritam, aukcijski algoritam.

SUMMARY

Mobile robots can assist workers in hospitals or senior homes, by performing simple tasks like brining people food or medicine. For their performance there needs to be a proper way of distributing tasks among them. In this thesis a multi-robot task assignment (MRTA) problem will be explained. First algorithm for solving task assignment problem will be lexicographic algorithm. Lexicographic algorithm assigns task using lexicographic method. After that First come first serve method will be used to assign tasks. To achieve optimal results Hungarian algorithm will be used. Previously mentioned algorithms use complete communication between mobile robots. To show how incomplete communication affects results Auction algorithm will be used. We will show how increase in communication range and number of mobile robots affects the results.

Key word: multi-robot task allocation, lexicographic algorithm, First come first serve, Hungarian algorithm, Auction algorithm

1. Uvod

Motivacija iza ovog diplomskog rada je integracija mobilnih robota u pametne prostore, posebno u bolnicama i domovima za starije. Trend svijeta je povećanje udjela starije populacije, te time raste i potreba za brigu oko njih. Kako bi se smanjio opseg rada osoblja koje se brine o starijim i bolesnim ljudima jedno od rješenja jest uvođenje robota kao ispomoć bolničkom osoblju. Mobilni roboti svake godine postaje sve napredniji i njihova integracija u svakodnevni život nije više područje znanstvene fantastike. Mobilni roboti mogu donijeti ljudima lijekove, hranu i druge potrepštine u isplaniranom rasporedu ili ako postoji izvanredna potreba. Ovaj će se rad fokusirati na rješavanje jednostavnih zadataka poput distribucije obroka i lijekova i njihovih zadatak između robota.

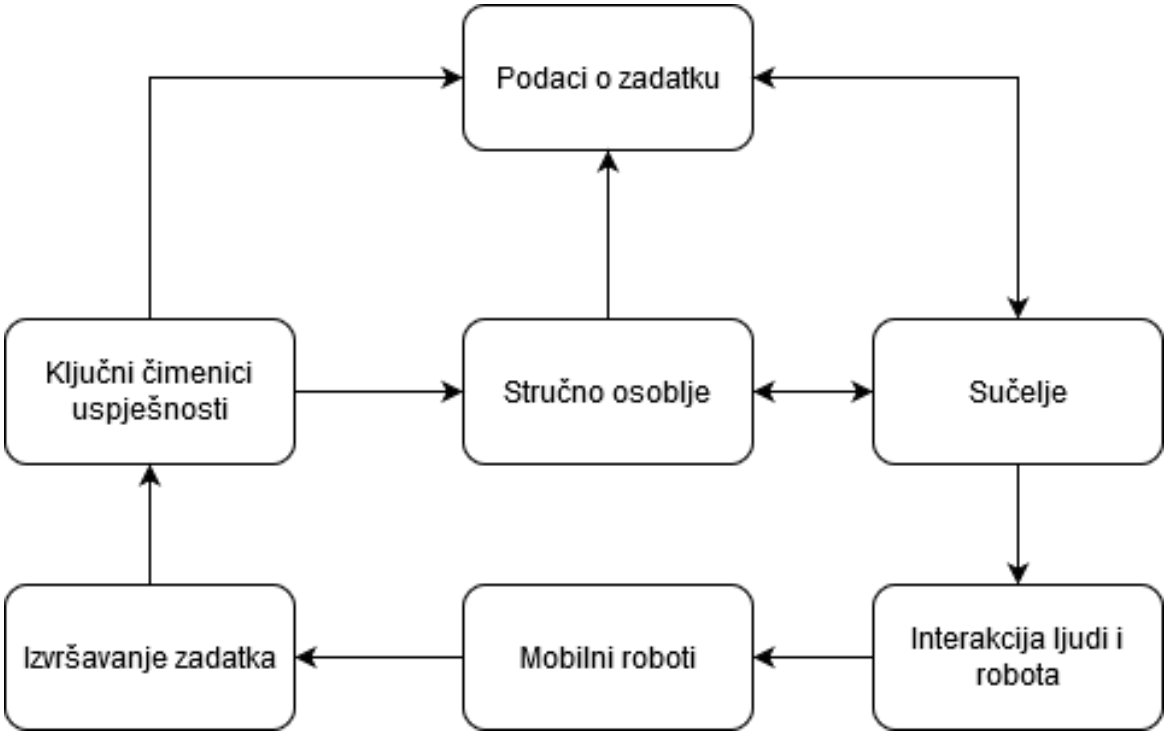
Za ovu tezu pretpostavit će se da objekt izvođenja simulacija ima bazu podataka s podatcima o pacijentima. Npr. broj sobe, plan prehrane, popis lijekova itd. Ovoj bazi podataka mogu pristupiti svi mobilni roboti. Obroci koji se distribuiraju dolaze u standardnim paketima, to jest, pakete već izrađuje osoblje ili drugi kuhinjski sustav. Mobilni roboti samo dolaze do mjesta preuzimanja i primaju paket. Smatra se da postupak preuzimanja paketa je već razvijen i da je trenutačan. Tako nema potrebe simulirati vrijeme potrebno za preuzimanje paketa.

Mobilni roboti znaju plan objekta tako da nema potrebe provoditi metodu istraživanja prostora ili mapiranje plana objekta. Zbog toga znaju i lokacije soba pacijenata koja su zapisana u bazi podataka. Osoblje objekta koristi već postojeće sučelje sustava te može promijeniti podatke ili dodati nove unose. Pomoću ovog sustava moguće je zadati robotima zadatke izvan rasporeda, na primjer, ako pacijentu hitno treba lijek, zadatak se može dati dinamički. Interakcija mobilnog robota i pacijenta nije tema ove teze, te smatra se dostupnom, funkcionalnom i trenutnom.

U ovoj tezi svi roboti mogu izvršavati sve zadatke i svi zadatci mogu biti dodijeljeni svim mobilnim robotima. Postoji samo jedna vrsta mobilnih robota i samo jedna vrsta zadataka. Ali ova teza se može proširiti tako da postoje različite vrste mobilnih robota ili zadataka, tako da samo neki roboti mogu izvršavati samo jedan ili nekoliko vrsta zadataka dok drugi mogu izvršavati sve zadatke. Isto se može primijeniti na zadatke tako da se neki zadatci mogu dodijeliti samo određenim vrstama mobilnih

robota.

Na slici 1, prikazana je idejni sustav za uvođenje mobilnih robota u pametne prostorije.



Slika 1. Prikaz idejnog sustava za uvođenje mobilnih robota u pametne prostorije

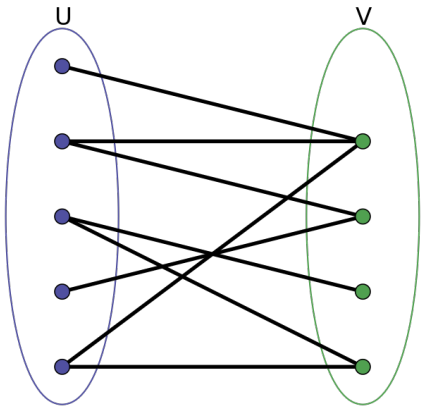
Način na koji medicinsko osoblje može komunicirati s mobilnim robotima je pomoću sučelja. Ovo sučelje dio je digitalizirane bolnice (za koju mi pretpostavljamo da postoji), a osoblje može preko tog sučelja vidjeti bazu podataka o pacijentima. Mobilni roboti mogu pristupiti toj bazi podataka. Pojam ”interakcija ljudi i mobilnih robota” sadrži zadatke koji se dodjeljuju mobilnim robotima. Mobilni roboti izvršavaju zadatke i na temelju ”ključnih čimbenika uspješnosti” može se ocijeniti njihova izvedba. Ključni čimbenici uspješnosti sadrže vrijeme (ekvivalentno udaljenosti) i prioritete, a mogu se koristiti za podešavanje ”zadatka”. ’Ključne faktore performansi’ može prilagoditi medicinski promijenivši bazu podataka o pacijentima. ”Podaci o zadatku” sadrže položaj zadatka, kategoriju zadatka (lijek, obrok ili drugo), te vrijeme kada zadatak treba izvršiti. Kategorija zadatka može se razgranati u obroke i lijekove, koji se zatim mogu razgranati u druge potkategorije (ručak1, ručak2, večera1, večera3, itd.).

Cilj ove teze je optimizirati prijeđene udaljenosti mobilnih robota. Hipoteza teze je da će kod nesavršene komunikacije između mobilnih robota povećavanjem broja robota ili pove cavanjem dometa komunikacije doći do poboljšavanja rezultata. Teza će započeti najjednostavnijim problemom s idealnim uvjetima i nakon toga će se uvesti limitacija dometa komunikacije između robota. U ovoj će tezi udaljenost i vrijeme biti zamjenjivi.

2. Dodjela zadataka robotskim sustavima

U ovom radu razmatramo problem dodjele zadatke robotskim sustavim (eng. *Multi-robot task allocation* - MRTA) čiji je cilj pronaći dodjeljivanje skupa od n broja zadataka skupu od n broja mobilnih robota tako da se optimizira neka od globalnih funkcija cilja (vidi, npr. [1, 2]). Ovaj problem odgovara linearnom problemu dodijele zadataka (eng. *linear assignment problem* -LAP) u području operacijskog istraživanja. Taj problem predstavlja problem gdje za zadani set agenata i za zadani set zadataka koji moraju biti dodijeljeni agentima po principu jedan na jedan cilj je minimizirati ukupni trošak izvršavanja dodijeljenih zadataka. Svaki agent (u ovom slučaju mobilni robot) obavlja jedan zadatak i svaki zadatak izvršava samo jedan agent.

Takav problem može se predstaviti bipartitnim grafom $G = (R \cup T, E)$ s dva seta vrhova R i T tako da svaki rub $e \in E$ u skupu rubova $E = R \times T$ povezuje jedan vrh iz skupa R s jednim vrhom iz skupa T. Primjer bipartitnog grafa prikazan je na slici 2. Neka je R skup robota $i \in R$ i neka je T skup zadatka $j \in T$. Nadalje, neka c_{ij} predstavlja trošak ruba $(i, j) \in E$, gdje je $i \in R$, $j \in T$. Taj trošak može biti prikazan kao vrijeme putovanja, udaljenost, trošak putovanja itd. Cilj je naći savršeno uparivanje između vrhova R i T kako bi suma svih troškova za uparene rubove minimalna. Rub $(i, j) \in E$ je uparen ako dva vrha i i j su međusobno povezani, a savršeno uparivanje je kada svaki vrh i od R je povezan točno jednom vrhu j od T i obrnuto. Problem LAP je identičan troškovnom bipartitom povezivanju, budući da smatrramo da je bipartitni graf uvijek kompletan. Ako nije uzimamo da su troškovi rubova koji nedostaju dovoljno veliki.



Slika 2. Primjer bipartitnog grafa

Neka su $|R|$ i $|T|$ kardinaliteti setova R i T i to tako da je $n = |R| = |T|$. Ukoliko $|R| \neq |T|$, može se dodati lažne (eng. *dummy*) čvorove setu koji ima manji kardinalitet i povezati ih s drugim setom dodatnim lukovima koji imaju trošak nula [3].

Matematička reprezentacija problema je:

$$\sum_{(i,j)\in RxT} c_{ij}x_{ij}. \tag{2.1}$$

Tako da:

$$\sum_{i\in R} x_{ij} = 1, \forall\, j \in T, \tag{2.2}$$

$$\sum_{j\in T} x_{ij} = 1, \forall\, i \in R, \tag{2.3}$$

$$x_{ij} \in \{0,1\} \forall i \in R, j \in T. \tag{2.4}$$

U jednadžbi 2.1, minimizira se dodjela zadataka robotima, gdje je x_{ij} upravljačka varijabla koja je

definirana kao:

$$x_{ij} = \begin{cases} 1 & \text{ako je rub (i,j) odabran,} \\ 0 & \text{inače.} \end{cases} \tag{2.5}$$

2.2 predstavlja ograničenje na uparivanje jednog zadatka s jednim robotom, a 2.3 predstavlja ograničenje uparivanja jednog robota s jednim zadatkom.

3. Rješavanje problema dodjele zadataka

U ovom poglavlju bit će prikazani različiti algoritmi za rješavanje problema dodijele zadataka mobilnim robotima. Prvo će se početi s potpunom komunikacijom između mobilnih robota i oni će znati lokacije svih ostalih mobilnih robota i lokacije svih zadataka. Nakon toga će se uvesti nepotpuna komunikacija s ograničenim dometom. Prvi algoritam koji će biti prikazan je leksikorafski algoritam.

3.1. Lekisografski algoritam

Uvest ćemo najprije najjednostavniju metodu raspodjele zadataka. Napravit će se dva skupa s podacima o položajima robota i položajima zadataka. Svaki će robot dobiti zadatak pomoću bijektivne funkcije, gdje je svaki element jednog seta uparen s točno jednim elementom drugog skupa, a svaki element drugog skupa je uparen s točno jednim elementom prvog skupa. Princip dodjele zadatka je jednostavan: prvom robotu u skupu R dodijeljen je prvi zadatak u skupu T, zatim drugom robotu drugi zadatak i sve tako dok zadnjem robotu nije dodijeljen zadnji zadatak iz skupa T. Budući da kod dodjele zadataka ne postoji sofisticirana metoda već ovisi o slučajnom generiranju setova robota i zadataka, rezultati ove metode bi trebali biti lošiji u odnosu na metode koje koriste neku bolju metodu odlučivanja.

Struktura seta robota je homogena, što znači da su svi roboti isti. Zadaci će također imati homogenu strukturu. Ne postoje težine niti prioriteti za određene zadatke. Cilj je samo odvesti mobilnog robota

na traženo mjesto.

Neka se definira matrica C kao matrica troška izvođenja algoritma, veličine $n \times n$ s elementima c_{ij}. Ta matrica predstavlja trošak dodijele zadatka za svaki mobilni robot, gdje element u i-tom retku i j-tom stupcu predstavlja cijenu dodijele j-tog zadatka i-tom robotu.

U nastavku će algoritam biti opisan pseudokodom. Algoritam započinje s funkcijom *Incijalizacija* koja uzima informacije položaja robota, položaja zadataka, brzine robota (u našem slučaju veličina koraka) i veličine prostora robota. Nakon toga funkcija *Dodjela zadataka* proizvodi matricu dodjele zadataka A, ovisno o kardinalitetima matrica R i T. Matrica A je veličine $|R| \times |T|$. Matrica R je veličine $i \times 2$, gdje je i broj mobilnih robota. Veličina matrice T je $j \times 2$, gdje je j broj zadataka. Stupci u matricama R i T odgovaraju osima x i y, prvi stupac je os x, a drugi stupac je os y. U ovoj tezi vrijedi $i = j = n$.

Nakon toga funkcija *Kretanje* uzima kao ulaz matrice A, R, T i vektor S (vektor veličine koraka). Funkcija *Kretanje* se izvodi u petlji sve dok vrijedi uvjet:

$$\sum A_{i,j} \neq min(|R|, |T|). \tag{3.6}$$

Sljedeći korak je izračunavanje udaljenosti između uparenih robota i zadataka koristeći izraz:

$$D_{i,j} = \sqrt{(T_{j,x} - R_{i,x})^2 - (T_{j,y} - R_{i,y})^2}. \tag{3.7}$$

Svakom iteracijom algoritma robot se pomiče u prostoru sve dok ne dođe na udaljenost koja je ista ili manja od vrijednosti koraka, te je u tom slučaju robot uspješno došao do zadataka. Ako robot još nije došao do zadatka izračunavanjem kuta α možemo odrediti u kojem se kvadrantu nalazi mobilni robot u odnosu na poziciju zadatka. Izraz za kut α je:

$$\alpha = \arctan \frac{(T_{j,y} - R_{i,y})}{(T_{j,x} - R_{i,x})} \cdot \frac{180}{\pi} [\text{deg}]. \tag{3.8}$$

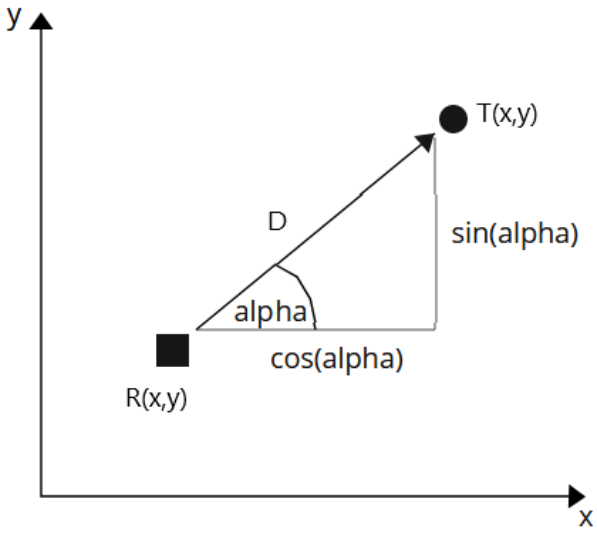
U prvom kvadrantu koordinatnog sustava komponente osi x i y se povećavaju. Dakle, ako je razlika između komponenti položaja zadatka i mobilnog robota za os y veća od nule, tada je mobilni robot ili u prvom ili drugom kvadrantu. Ako je razlika između komponenti položaja zadatka i mobilnog

robota osi x veća od nule, robot je u prvom kvadrantu. Sljedeći položaj mobilnih robota izračunava se izrazima:

$$R_{i,y} = R_{i,y} + S_i \cdot |\sin \alpha|, \tag{3.9}$$

$$R_{i,x} = R_{i,x} + S_i \cdot |\cos \alpha|. \tag{3.10}$$

Na slici 3, grafički je prikazan prvi kvadrant koordinatnog sustava. Zadatak je prikazan krugom a robot kvadrom.



Slika 3. Graphical representation of the first quadrant

Ako je razlika između komponeneti položaja zadatka i mobilnog robota za os x između 90 i $180°$, mobilni robot je u drugom kvadrantu i za izračunavanje položaja mobilnih robota koristit će se sljedeće formule:

$$R_{i,y} = R_{i,y} + S_i \cdot \sin\left(180 - |\alpha|\right), \tag{3.11}$$

$$R_{i,x} = R_{i,x} + S_i \cdot \cos\left(180 - |\alpha|\right). \tag{3.12}$$

Ako je razlika između komponeneti položaja zadatka i mobilnog robota za os y između 180 i $270°$, mobilni robot nalazi se u trećem. Formule za izračunavanje sljedećeg položaja mobilnih robota su:

$$R_{i,y} = R_{i,y} + S_i \cdot \sin(180 + |\alpha|), \tag{3.13}$$

$$R_{i,x} = R_{i,x} + S_i \cdot \cos(180 + |\alpha|). \tag{3.14}$$

Zadnji kvadrant je četvrti kvadrant gdje se komponente osi y smanjuju, a komponente osi x povećavaju. Formule koje se koriste za izračunavanje sljedećeg položaja mobilnog robota su:

$$R_{i,y} = R_{i,y} + S_{(i)} \cdot \sin(360 - |\alpha|), \tag{3.15}$$

$$R_{i,x} = R_{i,x} + S_{(i)} \cdot \cos(360 - |\alpha|). \tag{3.16}$$

U pseudo kodu koji slijedi indeks i bit će vezan za mobilne robote, dok će indeks j biti vezan za zadatke.Pseudo kod :

Algorithm 1 Leksikografski algoritam

input: R, T, S
Incijalizacija(R, T)
Dodjela zadataka(R, T)
Kretanje(A, R, T, S)
output: D, A

Algorithm 2 Dodjela zadataka - Leksikografski algoritam

input: R, T, A
for $i = 1 : n$ **do**
  for $j = 1 : n$ **do**
    $A_{i,j} = 1$
  end for
end for
output: A

Algorithm 3 Kretanje

input: A, R, T, S
while $\sum F_{i,j} \neq \min(|R|, |T|)$ **do**
  for $i = 1 : n$ **do**
    $D_{i,j} = \sqrt{(T_{j,x} - R_{i,x})^2 + (T_{j,y} - R_{i,y})^2}$
    if $D_{i,j} \leq S_i$ **then**
      $R_{i,x} = T_{j,x}$
      $R_{i,y} = T_{j,y}$
      $F_{i,j} = 1$
    else
      $\alpha = \arctan \frac{(T_{j,y} - R_{i,y})}{(T_{j,x} - R_{i,x})} \cdot \frac{180}{\pi}$
      if $(T_{j,y} - R_{i,y}) >= 0$ **then**
        $\%\ 0 - 89°$
        if $(T_{j,x} - R_{i,x}) >= 0$ **then**
          $R_{i,y} = R_{i,y} + S_i \cdot |\sin \alpha|$
          $R_{i,x} = R_{i,x} + S_i \cdot |\cos \alpha|$
          $\%\ 90 - 179°$
        else if $(T_{j,x} - R_{i,x}) < 0$ **then**
          $R_{i,y} = R_{i,y} + S_i \cdot \sin(180 - |\alpha|)$
          $R_{i,x} = R_{i,x} + S_i \cdot \cos(180 - |\alpha|)$
        end if
      end if
      if $(T_{j,y} - R_{i,y}) < 0$ **then**
        $\%\ 180 - 269°$
        if $(T_{i,x} - R_{i,x}) \geq 0$ **then**
          $R_{i,y} = R_{i,y} + S_{(i)} \cdot \sin(360 - |\alpha|)$
          $R_{i,x} = R_{i,x} + S_{(i)} \cdot \cos(360 - |\alpha|)$
          $\%\ 270 - 359°$
        else
          $R_{i,y} = R_{i,y} + S_i \cdot \sin(180 + |\alpha|)$
          $R_{i,x} = R_{i,x} + S_i \cdot \cos(180 + |\alpha|)$
        end if
      end if
    end if
  end for
end while
output: D, A

Ukratko, ovo je najjednostavniji algoritam za raspodjelu zadataka. Zadaci se robotima daju nasumično i ne postoji odluka o dodjeli zadataka. To nije optimalan algoritam za dodjelu zadataka jer vrijeme (ili udaljenost) potrebno za dovršavanje zadataka nije svedeno na minimum.

3.2. FCFS algortihm

Leksikografska metoda raspodjele zadataka je jednostavna, ali nije optimizirana zbog nedostatka metode odlučivanja. U sljedećoj verziji algoritma uvest ćemo jednu jednostavnu metodu donošenja odluka: "First come first serve" (FCFS). U metodi "First come first serve" mobilni robot koji je prvi u generiranom setu mobilnih robota R izračunava udaljenosti do svih zadataka i odabire onaj zadatak koja mu je najbliži. Odabrani zadatak se zatim ukloni iz seta preostalih zadataka. Nakon toga sljedeći izračunava udaljenosti i bira svoj najbliži zadatak, i tako dalje. Na kraju, posljednji mobilni robot uzima posljednji raspoloživi zadatak. Prednosti ove metode su u tome što smanjuje prosječnu minimalnu udaljenost i prosječnu udaljenost koju mobilni robovi prijeđu u usporedbi s leksikografskom metodom, no udaljenosti nisu raspoređene ravnomjerno među mobilnim robotima. Mobilni roboti koji kasnije odabiru zadatke imaju manje izbora od onih koje prvi odabiru i time prosječna udaljenost je veća za mobilne robote koji kasnije imaju mogućnost izbora.

Razlika u pseudo kodu (algoritam 4) između leksikografske metode i FCFS metode je različta funkcija *Dodjela* zadatka (algoritam 5). Funkcija *Dodjela zadatka* uzima matrice R i T kao ulaz. Zatim pomoću dvije petlje izračunava udaljenosti između mobilnog robota i dostupnih zadataka, te odabire najbliži zadatak, uklanja taj par mobilnog robota i zadatka i nastavlja sve dok svi mobilni roboti ne dobiju zadatak.

Algorithm 4 FCFS algoritam

1: input: R, T, S
2: Incijalizacija(R, T)
3: Dodjela zadataka(R, T, A)
4: Kretanje(A, R, T, S)
5: output: D, A

Algorithm 5 Dodjela zadataka - FCFS

input: R, T, A
for $i = 1 : n$ **do**
  for $j = 1 : n$ **do**
    $D_{i,j} = \sqrt{(T_{j,x} - R_{i,x})^2 + (T_{j,y} - R_{i,y})^2}$
  end for
end for
for $i = 1 : n$ **do**
  for $j = 1 : n$ **do**
    $A_{i,j} = \text{argmin}(D_{ij})$
  end for
end for
output: A

Hipoteza je da FCFS algoritam ima bolje rezultate od Leksikografskog algoritma, te će rezultati simulacija biti prikazani u poglavlju 4.

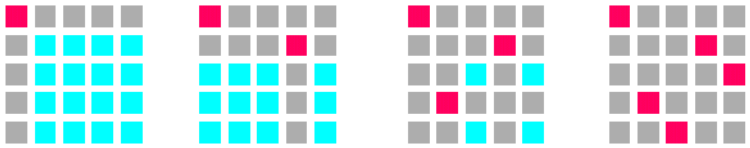
3.3. Mađarski algoritam

Sljedeća će verzija algoritma koristiti varijantu mađarskog algoritma. Razvijen od strane Harold Kuhna 1955. godine, a ime je dobio po djelima dvojice mađarskih matematičara: Dénesa Kőniga i Jenő Egerváry. James Munkres je 1957. unaprijedio algoritam i pridonio popularnosti algoritma [[4],[5]].

Mađarski algoritam, za razliku od algoritma koji forsira računanje svih mogućih udaljenosti između robota i zadataka (eng. *brute-force algorithm*) i time ima eksponencijalno kompleksno vrijeme računanja, nalazi najmanji element u svakom retku i oduzima ga od svih ostalih elemenata u istom retku. Rezultat je matrica koja ima najmanje jednu nulu u retku. Na temelju pozicija tih nula algoritam pokušava zadati zadatke mobilnim robotima tako da svaki agent radi samo jedan zadatak. Ponekad se može dogoditi da se ta matrica u ovoj fazi ne može koristiti za dodjeljivanje zadataka. U tom slučaju postupak je prvo dodijeliti što je moguće više zadataka tako da redovi koji imaju jednu nulu dodjele zadatake odgovarajućim mobilnim robotima. Ostatak postupka je označavanje redaka bez dodijelje-

nih zadataka i stupaca koji sadrže nule na sjecištu stupca i označenog retka. Nakon što je to učinjeno, algoritam pronalazi od preostalih elemenata najmanju vrijednost. Ovaj se element oduzima od drugih neoznačenih elementa i na temelju pozicija nula se pokuša dodijeliti zadatak. To se ponavlja sve dok se ne dodijele svi zadaci, odnosno kada je minimalni broj redaka koji se koriste za pokrivanje svih nula jednak n.

Prikaz prethodno opisanog postupka prikazan je na slici 4. Plavom bojom označeni su elementi koji se koriste za dodjelu zadatka, rozom bojom označeni su dodijeljeni zadaci mobilnim robotima. Tablica 1 prikazuje uparene mobilne robote i zadatke za primjer na slici 4.



Slika 4. Prikaz postupka dodjele zadataka kod mađarskog algoritma

Tablica 1. Tablica dodjele zadataka za primjer na slici 4

Par	Mobilni robot	Zadatak
#1	1	1
#2	2	4
#3	3	5
#4	4	2
#5	5	3

Algorithm 6 Mađarski algoritam

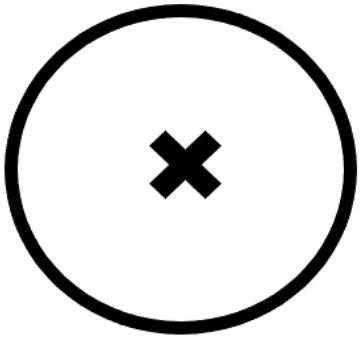
1: input: R, T, S
2: Incijalizacija(R, T)
3: Dodjela zadataka(R, T)
4: Kretanje(A, R, T, S)
5: output: D, A

Algorithm 7 Dodjela zadatka

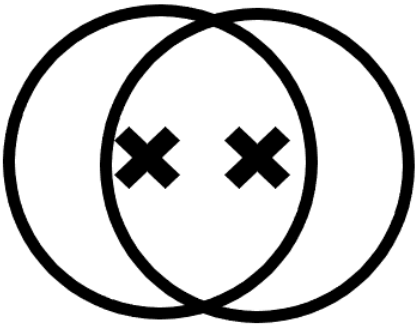
input :R, T
Korak 1. Kreirati matricu troška C
Korak 2. Pronaći element s najmanjom vrijednosti u svakom retku matrice C
Korak 3. Oduzeti element s najmanjom vrijednosti od ostalih elemenata u tom redu i to napraviti za svaki red.
Korak 4. Dodijeliti zadatak mobilnom robotu na poziciji gdje je element vrijednosti nula. Ako su svi zadaci dodijeljeni otiči na KRAJ. If not go to Step 6.
Korak 5. Označiti retke bez dodijeljenih zadataka i stupce koji sadrže nule na sjecištu stupca i označenog retka. Ponoviti dok nema novih označenih redaka.
Korak 6. Od preostalih elemenata oduzeti za svaki red oduzeti element najmanje vrijednosti od ostalih elemenata u tom redu. Dodijeliti zadatke na temelju pozicija elemenata s vrijednosti nula. Ako su svi zadaci dodijeljeni skočiti na KRAJ. Inače skočiti na Korak 5.
KRAJ
output: A, C

3.4. Nepotpuna komunikacija

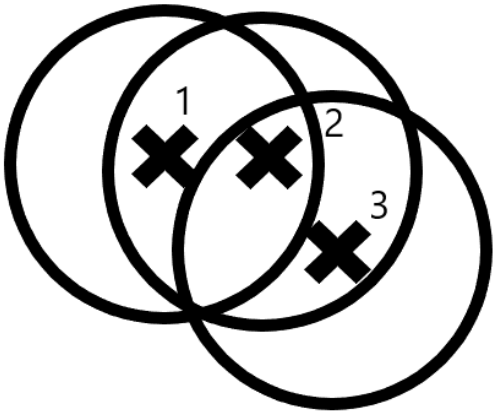
Sljedeća verzija algoritma je bez potpune komunikacije između mobilnih robota. Roboti mogu komunicirati samo ako su unutar komunikacijskog dometa. Također roboti mogu komunicirati s drugim robotima koji nisu u njihovom komunikacijskom dometu preko robota koji jesu u njihovom komunikacijskom dometu, ako ti roboti međusobno komuniciraju. Takav tip komunikacije zove se skokovita komunikacija (eng. *multi-hop communication*). Na slici 5, mobilni robot predstavljen je simbolom x, a krug oko njega predstavlja komunikacijski raspon mobilnog robota. Na slici 6, se vidi međusobna komunikacija dvaju robota koji su u komunikacijskom dometu jedan s drugim. Kad su dva ili više robota u komunikacijskom dometu jednog mobilnog robota oni mogu međusobno izmijeniti informacije iako nisu u međusobnom dometu, koristeći skokovitu metodu komunikacije. Na slici 7, mobilni robot 1 može komunicirati s mobilnim robotom 3 putem mobilnog robota 2, budući da je mobilni robot 2 u dosegu mobilnog robota 1, a mobilni robot 3 je u rasponu s mobilnim robotom 2.



Slika 5. Mobilni robot s komunikacijskim dometom



Slika 6. Međusobna komunikacija dvaju robota



Slika 7. Skokovita metoda komunikacije

Informacije koje mobilni roboti razmjenjuju su: zadatak koji im je dodijeljen, udaljenost koju su prošli, informacije o drugim mobilnim robotima s kojima su se susreli i starost informacija o drugim mobilnim robotima. Kad se dva ili više mobilnih robota međusobno susretnu, tada mogu međusobno podijeliti zadatke na temelju razmijenjenih informacija. Novije će se informacije upotrijebiti za rješavanje sukoba informacija koji se preklapaju. Na primjer, ako jedan robot ima informacije o jednom mobilnom robotu i dodijeljenom zadatku, a drugi mobilni robot ima različite podatke o istom mobilnom robotu, novije se informacije uzimaju kao ispravne.

3.5. Aukcijski algoritam

Za rješavanje problema dodjele zadtak s nepotpunom komunikacijom koristit će se aukcijski algoritam za dodjelu zadatka mobilnim robotima, prema [6]. Ovaj algoritam temelji se na nepovezanim i dinamičkim grafovima. Nepovezani grafovi, za razliku od povezanih, imaju više komponenti povezanosti, tj. postoji više povezanih komponenti jer nisu sve međusobno povezane. Dinamički grafovi su grafovi koji mijenjaju broj i strukture komponenti povezanosti kroz vrijeme kako se mobilni roboti kreću kroz prostor, i članovi povezanosti se mijenjanju iz jedne u drugu komponentu povezanosti. Kod aukcijskog algoritma roboti se međusobno nadmeću oko zadatka koji im najviše odgovora, po principu aukcije u stvarnom svijetu, tako da svaki iznese svoju ponudu (eng. *bid*). Ta ponuda je trošak putovanja tog mobilnog robota za taj zadatak u odnosu na drugi najbolji zadatak. Ako pretpostavimo da zadatak j ima cijenu p_j koju mobilni robot mora "platiti" kako bi izvršio taj zadatak. Tada, vrijednost zadatka j za mobilni robot i je $a_{ij} - p_j$. Svaki mobilni robot želi biti dodijeljen zadatku koji njemu ima najveću vrijednost, tj. tako da vrijedi:

$$j_i \in \arg\max_{j=1,\ldots,n} \{c_{ij} - p_j\}. \tag{3.17}$$

Aukcijski algoritam je iterativan, prvo se počinje s bilo kojom dodjelom zadatka i bilo kojim cijenama. Ako su svi roboti "zadovoljni" algoritam je završio no ako postoje roboti koji nisu "zadovoljni" dodijeljenim zadatkom oni se nadmeću za taj zadatak s drugim robotima koji su isto zainteresirani za taj zadatak. Svaki robot daje ponudu u iznosu:

$$p_{j_i} + \gamma_i, \tag{3.18}$$

gdje je p_{j_i} cijena j-tog zadtka za i-ti robot, a γ_i razlika između cijena za najbolji i drugi najbolji zadatak za tog robota. γ_i se može izraziti kao:

$$\gamma_i = v_i - w_i, \tag{3.19}$$

gdje je v_i zadatak s najboljom cijenom,

$$v_i = \max_j \{c_{ij} - p_j\}, \tag{3.20}$$

a w_i zadatak s drugom najboljom cijenom,

$$w_i = \max_{j \neq j_i} \{c_{ij} - p_j\}. \tag{3.21}$$

Proces nadmetanja se obavlja sve dok svi roboti nisu dobili zadatak s kojim su ”zadovoljni”. Algoritam ima četiri faze. Prva faza je faza inicijalizacije, gdje roboti provjeravaju postoje li roboti u njihovom komunikacijskom rasponu. Nakon toga provjere imaju li susjedni roboti druge robote u svom komunikacijskom rasponu.

Druga faza je dijeljenje informacija i ažuriranje matrice zadataka. Svaki mobilni robot ima informacije o dodijeljenom zadatku, udaljenosti koju je prešao i one informacije za ostale mobilne robote koje je susreo na putu. Postoje tri strategije kada je u pitanju dijeljenje informacija. Prva je da nema razmjene informacija, te svi mobilni roboti drže dodijeljene zadatke. Druga je dijeljenje samo informacija mobilnih robota koji su u dometu. Ne dijele se informacije o ostalim robotima koje su se susreli na putu. Treća strategija je dijeljenje svih informacija koje roboti imaju, što uključuje i znanje o položaju drugih mobilnih robota, koje se su susreli usput. Ako postoje podaci koji se ne podudaraju, uzima se da su novije informacije pouzdanije. U ovoj tezi uzet će se treća strategija, tj. dijeljenje svih

informacija. Na temelju zajedničkih informacija mobilni roboti mogu međusobno podijeliti zadatke, a to je treća faza.

U trećoj se fazi zadaci se mogu predodijeliti ako postoji sukob dva ili više robota kojima su dodijeljeni isti zadaci ili postoje nove informacije koje mogu dovesti do boljih rezultata. Predodijela zadatka obavlja se po prethodnom opisanom principu aukcije. Četvrta, i posljednja faza, je faza kretanja. Mobilni roboti kreću se prema zadanom zadatku po principu kao i u prethodnim algoritmima. Ove četiri faze zaključuju jednu iteraciju algoritma i ponavljaju se sve dok svi roboti ne pronađu svoje zadatke. Pseudo kod algoritma može se vidjeti u algoritmu 8.

Algorithm 8 Aukcijski algoritam

1: input: $R, T, S, domet$
2: **while** $\sum(T_{ij} - R_{ij}) \neq 0$ **do**
3: Incijalizacija($R, T, domet$)
4: Ažuriranje podataka(R, T, A)
5: Dodjela zadataka(R, T, A)
6: Kretanje($R, T, S, domet, A$)
7: **end while**
8: output: D, A

4. Simulacije

Za opisane algoritme provest će se simulacije u programskom alatu Matlab, kako bi se ispitala točnost hipoteze. Kod potpune komunikacije očekuje se da će leksikografski algoritam imati najlošije rezultate, FCFS algoritam bolje, a mađarski algoritam optimalne.

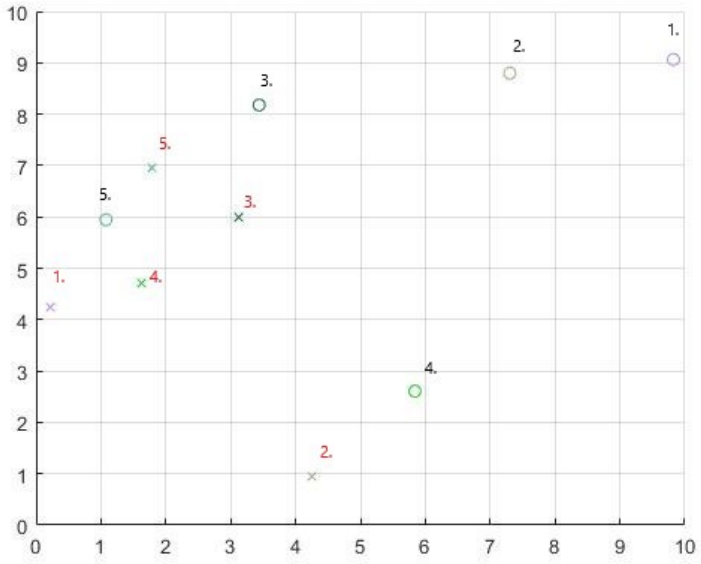
Kod nepotpune komunikacije hipoteza je da s povećanjem broja mobilnih robota ili povećanjem komunikacijskog dometa dolazi do poboljšavanja rezultata.

Za oba tipa komunikacije provest će se ispitivanja u grupama od 10, 20, 30, 40 i 50 mobilnih robota. Bit će pet mjerenja po grupi robota. Početni uvjeti će biti isti za sve tri algoritma kod potpune komunikacije. Također kod nepotpune komunikacije bit će isti početni uvjeti za različite veličine ko-

munikacijskog dometa kako bi se vjerodostojno usporedili dobiveni rezultati. Prostor u kojem će se izvoditi simulacije je bezdimenzijski prostor veličine 100×100.

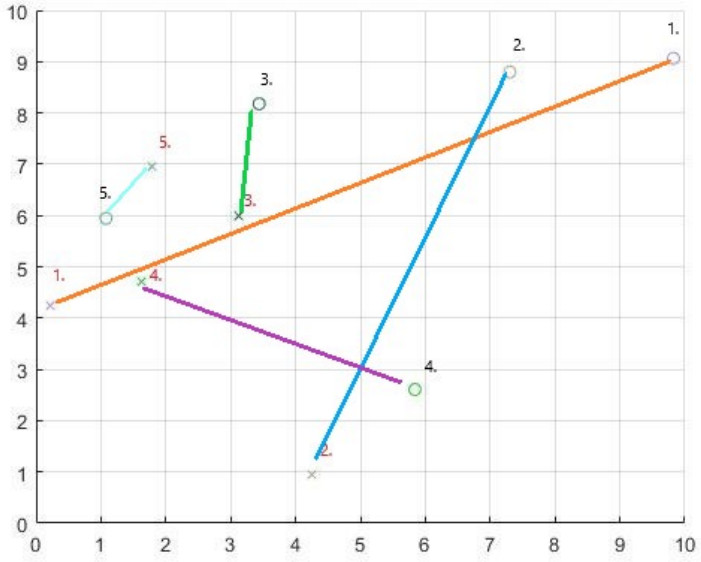
4.1. Simulacije za potpunu komunikaciju

Kako bismo pokazali razliku u načinu rada tri opisana algoritma s potpunom komunikacijom i istakli njihove prednosti i nedostatke, upotrijebit ćemo set od pet nasumično generiranih pozicija robota i zadataka na slici 8. Sa simbolom x označene su pozicije zadatka, a sa simbolom kruga označene su pozicije mobilnih robota. Brojevi pokraj simbola označavaju poziciju mobilnog robota ili zadatka u njihovim setovima. Npr. s 1 su označeni prvi mobilni robot i prvi zadatak u setovima R i T.

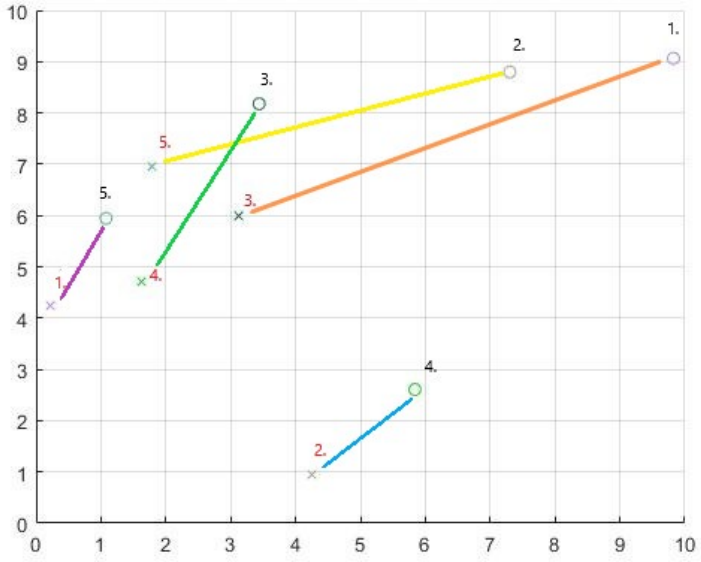


Slika 8. Primjer nasumično generiranih robota i zadataka

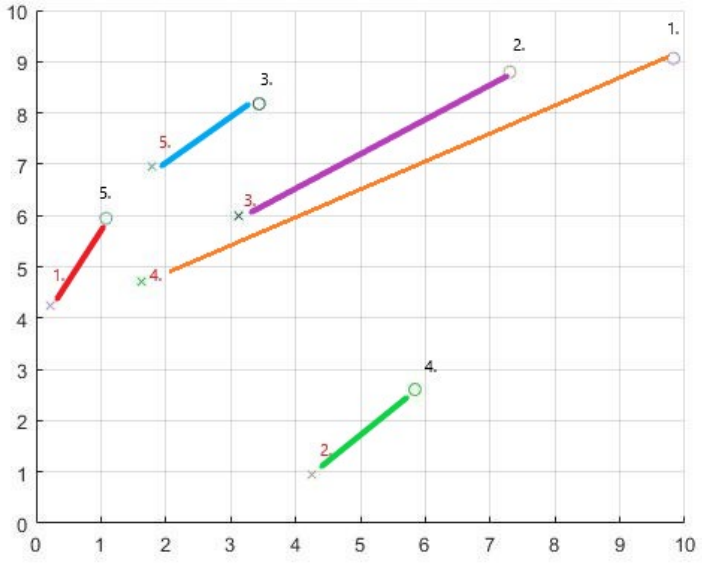
Slika 9 prikazuje parove robota i zadataka leksikografskog algoritma za primjer sa slike 8. Slika 10 prikazuje parove za FCFS algoritam, a slika 11 prikazuje parove mađarskog algoritma. Tablica 2 prikazuje parove mobilnih robota i zadataka za primjer sa slike 8. Desni stupac kod svake kategorije je vezan za mobilne robote, a lijevi za zadatke.



Slika 9. Rezultat leksikografskog algoritma



Slika 10. Rezultat FCFS algoritma



Slika 11. Rezultat mađarskog algoritma

Tablica 2. Parovi mobilnih robota i zadataka za primjer sa slike 8

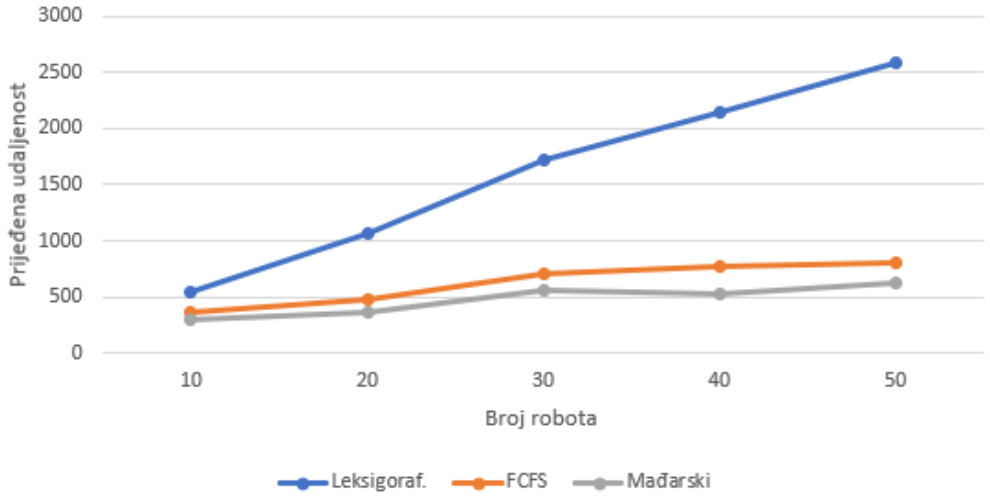
Par	Leksikografski		FCFS		Mađarski	
#1	1	1	1	3	1	4
#2	2	2	2	5	2	3
#3	3	3	3	4	3	5
#4	4	4	4	2	4	2
#5	5	5	5	1	5	1

Iz slika 9 do 11 može se vidjeti kako leksikografski algoritam ima velike udaljenosti za robote 1, 2 i 3. Robot 1 nalazi se u gornjem desnom kutu a zadatak 1 je na lijevoj strani radnog prostora. Budući da je kod leksikografskog algoritma prvomom robotu dodijeljen prvi zadatak neovisno o njihovim pozicijama može se vidjeti kako ovaj algoritam nije optimalan. Razlog je manjak metode odlučivanja. Također budući da su pozicije robota i zadataka nasumično izgenerirane nisu sve udaljenosti između parova ogromne, kao što se i može vidjeti iz para peti robot i peti zadatak. FCFS algoritam ima bolju raspodjelu udaljenosti od leksikografskog algoritma, budući da roboti uzimaju najbliži dostupni zadatak. Ovaj algoritam time smanjuje udaljenosti u odnosu na leksikografski algoritam budući da postoji ta, iako jednostavna metoda odlučivanja. Mađarski algoritam ima najbolju raspodjelu budući

da on optimira rezultate. Iako robot 1 dobiva udaljeniji zadatak 4 u odnosu na FCFS algoritam gdje robot 1 dobiva zadatak 3, ostatak parova je bolje dodijeljen. Mađarski algoritam je optimalan jer su uračunate kombinacije udaljenosti, što je kompleksnija metoda odlučivanja u odnosu na FCFS algoritam. U nastavku će biti detaljnije pokazani rezultati simulacija.

Za algoritme s potpunom komunikacijom ispitivat će se uspješnost na temelju četiri parametra: ukupna prijeđena udaljenost, minimalna udaljenost koju je mobilni robot prešao, maksimalnu udaljenost koju je mobilni robot prešao i prosječnu udaljenost. Prikazat će se prosjeci od 5 mjerenja u svakoj grupi robota.

Na slici 12, prikazane su ukupne prijeđenje udaljenosti za sve tri metode.



Slika 12. Prijeđene udaljenosti kod potpune komunikacije

Na slici 12, može se vidjeti kako leksikografski algoritam jednoliko povećava udaljenost kako se povećava broj robota. Budući da algoritam nasumično dodijele zadatke povećavanjem broja robota nije došlo do nikakvog poboljšanja. FCFS i mađarski algoritam imaju blagi rast. Blagi rast kod FCFS i mađarskog algoritma je za očekivati budući da broj robota raste, a time i ukupna prijeđena udaljenost. FCFS ima veće vrijednosti od mađarskog algoritma, ali značajnije smanjenje udaljenosti u odnosu na leksikografski algoritam. To pokazuje kako je za poboljšavanje rezultata dovoljno imati barem jednostavnu metodu odlučivanja. No, FCFS algoritam svejedno nije optimalan jer ne uzima dobrobit svih mobilnih robota, već prednost u odlučivanju imaju mobilni roboti koji su na prvim pozicijama u

setu robota R. Mađarski algoritam uzima u obzir kombinacije udaljenosti za mobilne robote i time optimira rezultate i dodatno smanjuje prijeđene udaljenosti u odnosu na oba prethodna algoritma.

U tablici 3 prikazani su prosjeci rezultata ukupne udaljenosti za mjerenja.

Tablica 3. Tablica prijeđenih udaljenosti kod potpune komunikacije

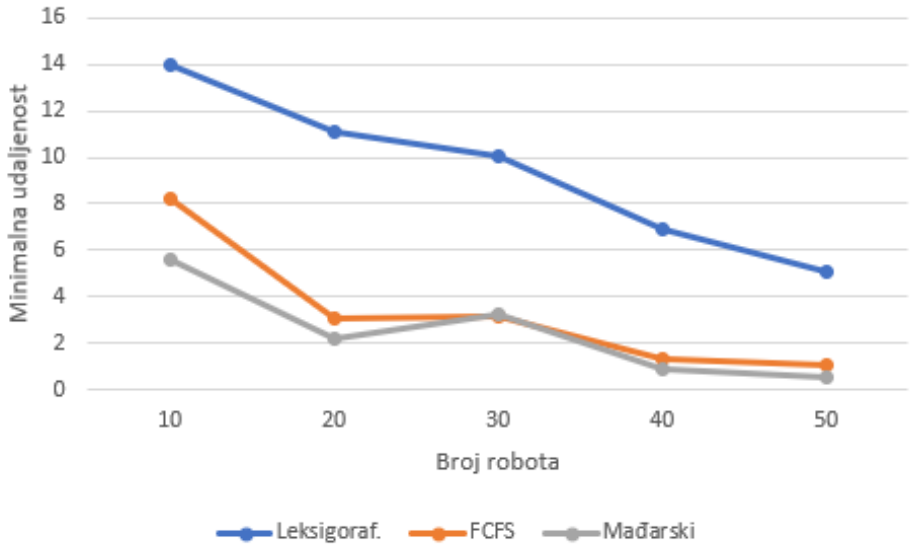
Br. robota	Leksikografski	FCFS	Mađarski algoritam
10	546,11	368,46	294,58
20	1063,50	475,02	371,09
30	1714,75	704,06	558,87
40	2146,52	774,51	531,27
50	2578,08	803,41	623,43

Iz tablice 3 vidi se za grupu od 10 robota FCFS ima smanjenje od $32,5\%$ u odnosu na leksikografski algoritam. Mađarski algoritam ima smanjenje od $20,05\%$ u odnosu na FCFS algoritam i $46,05\%$ u odnosu na leksikografski algoritam. Daljnjim povećavanjem broja robota razlike između leksikografskog i druga dva algoritma rastu. Za grupu od 20 robota FCFS ima smanjenje od 50%, a mađarski $21,81\%$ u odnosu na FCFS i $65,1\%$ u odnosu na leksikografski. Za zadnju grupu u mjerenju, 50 robota, FCFS ima smanjenje od $68,8\%$ u odnosu na leksikografski algoritam, a mađarski ima smanjenje od $22,4\%$ u odnosu na FCFS i $75,81\%$ u odnosu na FCFS algoritam. U prosjeku mađarski algoritam je doveo do smanjenja od $20-25\%$ u odnosu na FCFS algoritam, a leksikografski algoritam je sve lošiji kako raste broj robota.

Na slici 13, prikazani su rezultati mjerenja za minimalne udaljenosti koje su mobilni roboti prešli.

Minimalne prijeđene udaljenosti se ponašaju suprotno u odnosu na ukupne prijeđene udaljenosti po principu da kako broj robota raste tako pada minimalna udaljenost. Za sva tri algoritma najveći skok je od grupe 10 robota do grupe 20 robota. Leksikografski algoritam ima najlošije rezultate, ali minimalne udaljenosti padaju jer je više robota u istom prostoru te raste njihova gustoća. Minimalne udaljenosti leksikografskog algoritma jednoliko padaju kako raste broj robota. FCFS i mađarski algoritam se ponašaju slično. Od 10 do 20 robota smanjenje minimalne udaljenosti je $62,3\%$ za FCFS algoritam i $61,1\%$ za mađarski algoritam. Od 20 do 30 mobilnih robota FCFS skoro pa i nema pro-

mjenu u minimalnoj udaljenosti, dok mađarski algoritam ima rast. Razlog za taj rast kod mađarskog algoritma u odnosu na FCFS algoritam je činjenica da FCFS algoritam forsira minimalne udaljenosti dok mađarski algoritam optimira ukupan raspored udaljenosti i time potencijalno povećava minimalnu udaljenost kako bi smanjio ukupnu, maksimalnu i prosječnu udaljenost. Daljnje povećanje mobilnih robota od 30 do 40 mobilnih robota pridonosi smanjenju od $58,2\%$ za FCFS i $73,3\%$ za mađarski algoritam. Od 40 do 50 minimalna udaljenost nastavlja padati.



Slika 13. Minimalne prijeđene udaljenosti kod potpune komunikacije

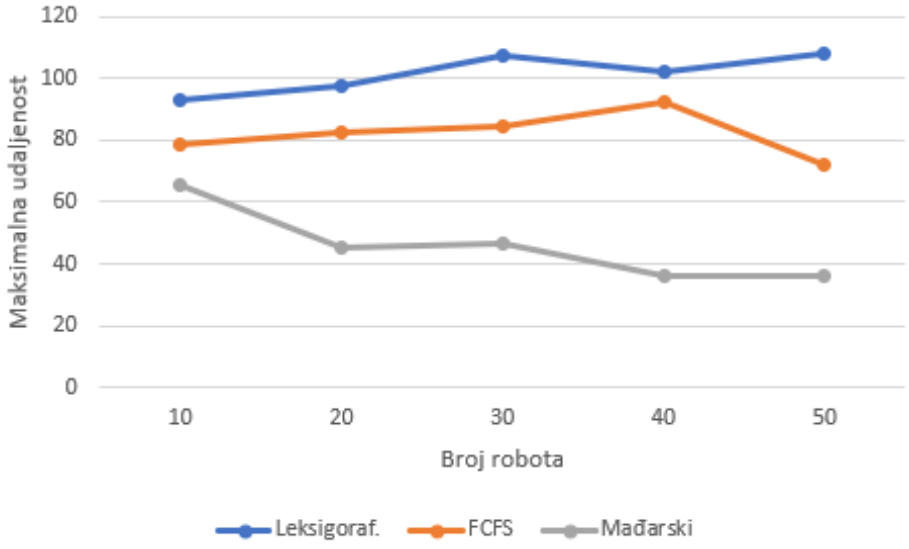
Tablica 4. Tablica minimalnih prijeđenih udaljenosti kod potpune komunikacije

Br. robota	Leksikografski	FCFS	Mađarski algoritam
10	13,96	8,22	5,55
20	11,08	3,10	2,16
30	10,02	3,17	3,23
40	6,89	1,33	0,86
50	5,08	1,02	0,51

U tablici 4 prikazane su minimalne prijeđene udaljenosti za mjerenja, te se može se vidjeti kako za grupu od 10 robota FCFS algoritam ima smanjenje od $41,05\%$ u odnosu na leksikografski algoritam. Mađarski algoritam ima smanjenje od $20,05\%$ u odnosu na FCFS i $60,18\%$ u odnosu na leksikografski

algoritam. Trend smanjenja je isti kao i kod ukupnih prijeđenih udaljenosti. Za grupu od 50 robota FCFS ima smanjenje od $79,77\%$ u odnosu na leksikografski algoritam, a mađarski algoritam ima smanjenje od $50,5\%$ u odnosu na FCFS algoritam i $89,97\%$ u odnosu na mađarski algoritam.

Na slici 14, prikazane su maksimalne prijeđene udaljenosti za provedena mjerenja.



Slika 14. Maksimalne prijeđene udaljenosti kod potpune komunikacije

Kao što se može vidjeti iz slike 14 od leksikografskog algoritma maksimalne prijeđene udaljenosti rastu s povećavanjem broja robota. Razlog kao i prije je manjak metode odlučivanja. Između grupa od 30 do 40 robota dolazi do smanjenja od $4,8\%$ te nakon toga dolazi do rasta od $5,9\%$. Razlog smanjenja je nasumično generiranje pozicija robota i zadataka te manjak metode odlučivanja. Zbog nasumičnog generiranje došlo je boljeg raspodjele pozicija robota i zadataka te je to odstupanje od ponašanja ostatka grupa mobilnih robota. Maksimalne prijeđene udaljenosti kod FCFS algoritma također rastu od 10 do 40 robota, te između 40 i 50 dolazi do naglog pada. FCFS algoritam uzima u obzir minimalne udaljenosti, a ne maksimalne te je to razlog rasta maksimalnih prijeđenih udaljenosti. Razlog smanjenja udaljenosti od grupe 40 do grupe 50 je isti kao i kod leksikografskog algoritma, a to je nasumično generiranje pozicija robota i zadatak te metoda odlučivanja koja ne uzima u obzir maksimalne prijeđene udaljenosti. Mađarski algoritam jedini ima smanjenja maksimalnih prijeđenih udaljenosti jer metoda odlučivanja kod mađarskog algoritma uzima u obzir raspored udaljenosti tako

da se minimizira ukupna udaljenost, a time i maksimalna udaljenost. Smanjenje od početne grupe od 10 robota do konačne grupe od 50 robota kod mađarskog algoritma je $44,9\%$.

Tablica 5. Tablica maksimalnih prijeđenih udaljenosti kod potpune komunikacije

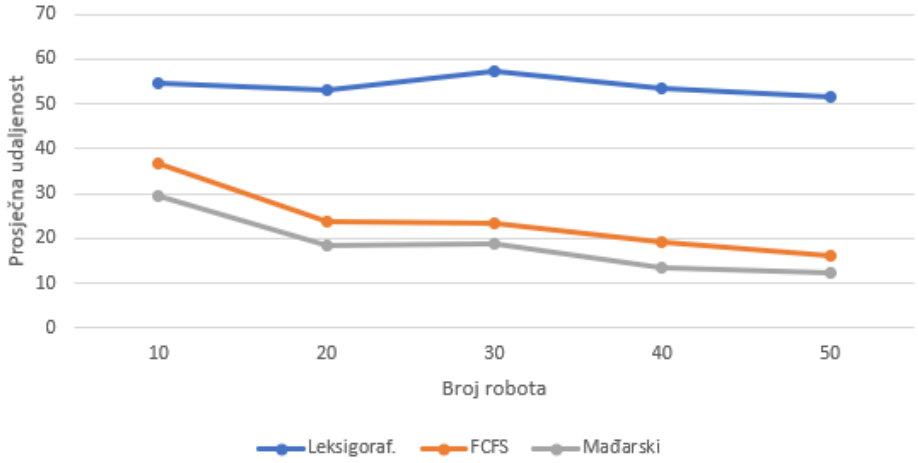
Br. robota	Leksikografski	FCFS	Mađarski algoritam
10	92,85	78,74	65,52
20	97,62	82,46	45,36
30	107,07	84,66	46,38
40	101,93	92,15	36,00
50	107,96	72,19	36,13

U tablici 5 prikazane su maksimalne prijeđene udaljenosti za provedena mjerenja. Za grupu od 10 robota došlo je do smanjenja od $15,19\%$ kod FCFS algoritma u odnosu na leksikografsku metodu. Za mađarski algoritam došlo je do smanjenja od $16,74\%$ u odnosu na FCFS algoritam i smanjena od $29,43\%$ u odnosu na leksikografski algoritam. Za grupu od 20 mobilnih robota FCFS algoritam doveo je smanjenja od $15,5\%$ u odnosu na leksikografski algoritma. Mađarski algoritam doveo je smanjenja od $44,9\%$ u odnosu na FCFS algoritam i do smanjenja od $53,5\%$ u odnosu na leksikografski algoritam. Sličan trend je i za grupu od 30 robota. Za grupu od 40 robota FCFS je doveo do smanjena od $9,6\%$ u odnosu na leksikografsku metodu što je manje od prethodnih grupa, ali je svejedno smanjenje. Mađarski algoritam doveo je do smanjenja od $60,9\%$ u odnosu na FCFS algoritam i do smanjenja od $64,67\%$ u odnosu na leksikografski algoritam. Za grupu od 50 robota, zbog naglog smanjenja maksimalne udaljenosti kod FCFS algoritam razlike su se promijenile. FCFS algoritma je doveo do smanjenja od $33,12\%$ u odnosu na leksikografsku metodu. Mađarski algoritam je doveo do smanjenja od $49,94\%$ u odnosu na FCFS algoritam i do smanjenja od $66,53\%$ u odnosu na leksikografski algoritam.

Na slici 15, prikazane su prosječne prijeđene udaljenosti za provedene simulacije.

Za leksikografsku metodu nema značajnog smanjenja prosječnih udaljenosti s povećanjem broj robota. Između grupa od 20 i 30 mobilnih robota došlo je do blagog povećanja prosječne udaljenosti od $7,49\%$, nakon toga dolazi do blagog smanjivanja prosječne udaljenosti. Razlog odstupanja je nasumičnost

leksikografskog algoritma. FCFS i mađarski algoritam se ponašaju jednako, uz razliku da mađarski algoritam ima u prosjeku oko 20% manje vrijednosti prosječnih udaljenosti.



Slika 15. Prosječne prijeđene udaljenosti kod potpune komunikacije

Tablica 6. Tablica prosječnih prijeđenih udaljenosti kod potpune komunikacije

Br. robota	Leksikografski	FCFS	Mađarski algoritam
10	54,61	36,85	29,45
20	53,17	23,75	18,55
30	57,15	23,46	18,63
40	53,66	19,36	13,28
50	51,56	16,06	12,47

U tablici 6 prikazane su vrijednosti prosječnih udaljenosti za provedena mjerenja. Razlike u prosječnim udaljenostima za tri navedena algoritma ponašaju se jednako kao i kod ukupnih prijeđenih udaljenosti. Za 10 robota FCFS algoritam doveo je do smanjenja od $31,5\%$ u odnosu na leksikografsku metodu. Mađarski algoritam je doveo do smanjenja od $20,05\%$ u odnosu na FCFS algoritam te smanjenja od $46,05\%$ u odnosu na leksikografski algoritam. Najveće razlike u odnosu na leksikografski algoritam su za grupu od 50 robota. FCFS algoritam doveo je do smanjenja od $68,8\%$, a mađarski algoritam do smanjenja od $75,8\%$ u odnosu na leksikografski algoritam. Mađarski algoritam za grupu od 50 robota doveo je do smanjenja prosječnih udaljenosti od $22,4\%$ u odnosu na FCFS algoritam.

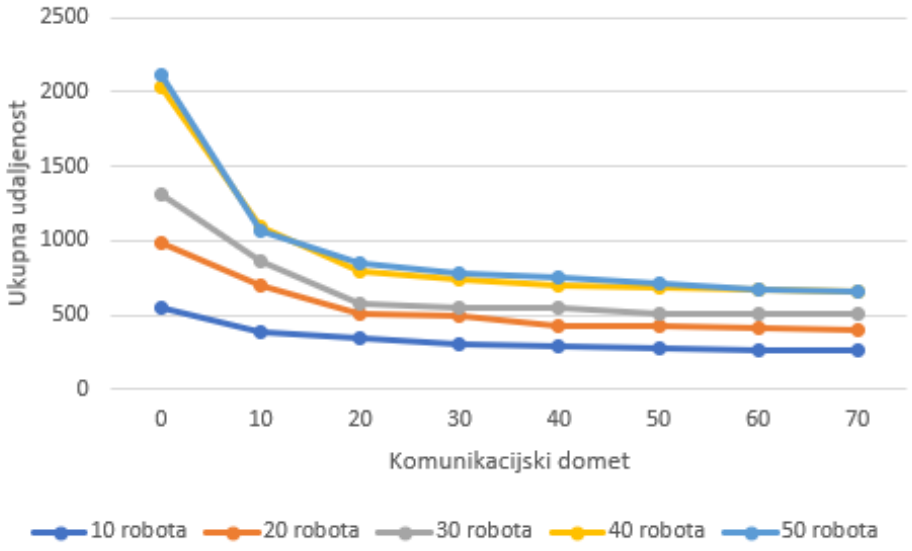
Prema navedenim mjerenjima može se zaključiti da je početna hipoteza potvrđena. Leksikografska metoda se ponaša najgore zbog nedostatka prave metode odlučivanja. FCFS sa svojom jednostavnom metodom odlučivanja, uzimanje minimalnih dostupnih udaljenosti, doveo je do značajnog smanjenja ukupne, minimalne i prosječne udaljenosti. Maksimalne udaljenosti su se smanjile u odnosu na leksikografsku metodu ali ne značajno, izuzev grupe od 50 mobilnih robota. Mađarski algoritam optimira ukupne udaljenosti sa svojom metodom odlučivanja koja uzima u obzir sve pozicije robota i zadataka. Za sve mjerene udaljenosti mađarski algoritam je imao najbolje rezultate.

4.2. Simulacije za nepotpunu komunikaciju

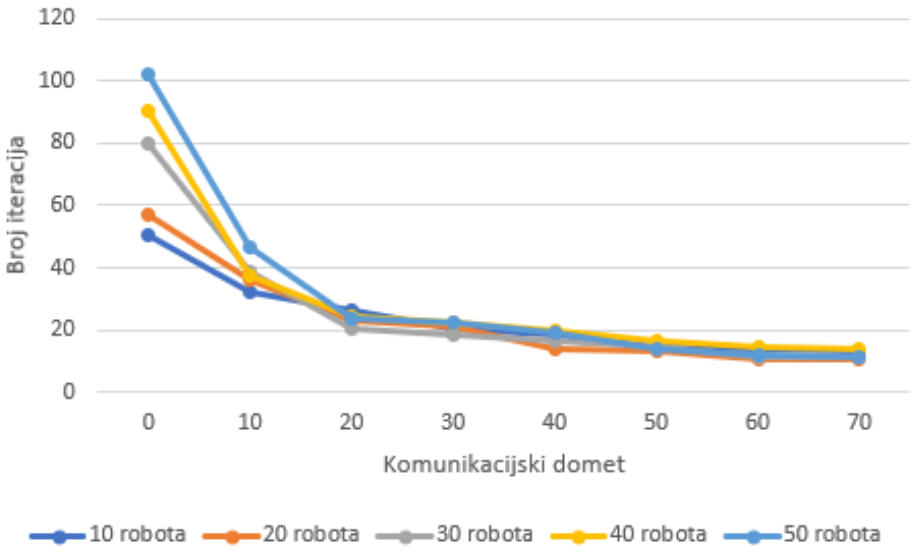
Kod nepotpune komunikacije mobilni roboti mogu komunicirati samo kada su u međusobnom komunikacijskom dometu. Također mogu komunicirati i s robotima koji nisu u njihovom komunikacijskom dometu preko robota koji jesu u komunikacijskom dometu. Hipoteza je da će povećanje broja robota poboljšati rezultate jer je gušći raspored robota te je mogućnost komunikacije vjerojatnija. Ako se poveća komunikacijski domet rezultati bi trebali biti bolji jer mobilni roboti pokrivaju više prostora sa svojim komunikacijskim dometom.

Simulirat će se utjecaj komunikacijskog dometa za grupe robota od 10 do 50 s inkrementom 10. U svakoj grupi će biti pet mjerenja i početni uvjeti će biti isti za sve komunikacijske domete u mjerenju.

Na slici 16, prikazane su ukupne prijeđene udaljenosti za grupe mobilnih robota. Za svaku grupu povećavat će se domet dok ne dođe do konvergencije rješenja. Može se vidjeti kako s povećavanjem dometa padaju prijeđene udaljenosti. Sve krivulje se ponašaju po istom principu. Najveći skokovi su kada se s nule uvede domet 10 jer roboti imaju veće vjerojatnosti za komunikaciju. Kod dometa 0 mobilni roboti mogu komunicirati samo ako su na istoj točki. Time se smanjuje broj informacija koje roboti dijele o drugim mobilnim robotima koje su susreli putem što dovodi do lutanja mobilnih robota koji nisu pronašli zadatak koji nije zauzet. Povećavanjem broja robota raste i ukupna prijeđena udaljenost koja je najveće vrijednosti za 50 robota. No, daljnjim povećavanjem dometa prijeđena udaljenost kreće konvergirati.



Slika 16. Prijeđena udaljenost kod nepotpune komunikacije - po komunikacijskom dometu



Slika 17. Broj iteracija kod nepotpune komunikacije - po komunikacijskom dometu

Na slici 17, prikazan je broj iteracija koje su provede u mjerenju. Broj iteracija s povećavanje dometa počinje konvergirati oko istih vrijednosti za različite grupe mobilnih robota. Kao i kod prijeđenih udaljenosti grupa od 50 robota ima najveći broj iteracija za domet 0, te kao i kod prijeđenih udaljenosti uvođenjem i male vrijednosti dometa broj iteracija se značajno smanjuje.

U nastavku će tablično biti prikazani rezultati mjerenja za različite grupe robota. Bit će prikazane udaljenosti i broj iteracija, te njihovo postotno smanjivanje povećavanjem veličine dometa.

Tablica 7. Tablica prijeđenih udaljenosti i broja iteracija za 10 robota kod nepotpune komunikacije

Domet	Udaljenost	Smanjenje udalj. [%]	Broj iteracija	Smanjenje iter. [%]
0	549.95	-	50,8	-
10	386,01	29,81	32,4	36,22
20	342,20	11,35	26,6	17,9
30	306,6	10,4	20,2	24,06
40	295,73	3,55	18,6	7,9
50	274,15	7,29	14,8	20,4
60	262,29	4,33	13,6	8,11
70	259,11	1,21	13,2	2,94

Tablica 7 pokazuje kako udaljenost uvođenjem dometa 10 pada za $29,81\%$, što je najveće smanjenje vrijednosti u mjerenju. Broj iteracija također pada za vrijednost $36,22\%$. Daljnjim povećavanjem dometa vrijednosti padaju te oko dometa 60 vrijednost konvergiraju, jer između dometa 60 i 70 smanjenje udaljenosti je samo $1,21\%$ što je relativno mala vrijednost.

Tablica 8. Tablica prijeđenih udaljenosti i broja iteracija za 20 robota kod nepotpune komunikacije

Domet	Udaljenost	Smanjenje udalj. [%]	Broj iteracija	Smanjenje iter. [%]
0	979,44	-	56,8	-
10	701,61	28,37	36,4	35,91
20	510,24	27,27	23	36,81
30	497,53	2,49	21,4	6,96
40	423,43	14,89	14	34,58
50	431,34	-1,87	13,4	4,29
60	409,56	5,04	10,4	22,39
70	401,76	1,91	10,4	0

Tablica 8 pokazuje mjerenja za grupu od 20 mobilnih robota. Može se vidjeti kako se vrijednosti ponašaju kao i kod grupe od 10 robota. Kod grupe od 50 mobilnih robota došlo je do blagog povećanja udaljenosti od $1,87\%$ što je zanemariva razlike, te se daljnjim povećavanjem dometa udaljenost sma-

njila i konvergirala oko vrijednosti dometa od 60. Broj iteracija je također konvergirao oko vrijednosti 60 te kod vrijednosti 70 nema daljnjem pada broj iteracija.

Tablica 9. Tablica prijeđenih udaljenosti i broja iteracija za 30 robota kod nepotpune komunikacije

Domet	Udaljenost	Smanjenje udalj. [%]	Broj iteracija	Smanjenje iter. [%]
0	1314,99	-	79,6	-
10	859,22	34,66	39	51,00
20	582,53	32,21	20,4	47,69
30	545,95	6,29	18,2	10,78
40	544,19	0,322	16,8	7,69
50	509,35	6,40	14,6	13,1
60	513,44	-0,80	11,6	20,55
70	512,34	0,21	11,4	1,72

Tablica 9 prikazuje vrijednosti udaljenosti za grupu od 30 mobilnih robota. Kao i prethodnim grupama udaljenosti se najviše smanje od vrijednosti dometa 0 do vrijednosti dometa 10 za $46,24\%$. Udaljenosti konvergiraju oko vrijednosti dometa od 50. Broj iteracija isto se najviše smanjio kod povećavanja dometa s 0 na 10.

Tablica 10. Tablica prijeđenih udaljenosti i broja iteracija za 40 robota kod nepotpune komunikacije

Domet	Udaljenost	Smanjenje udalj. [%]	Broj iteracija	Smanjenje iter. [%]
0	2034,4	-	90,2	-
10	1093,71	46,24	37,2	58,76
20	799,55	26,89	24,2	34,95
30	743,05	7,07	22,4	7,44
40	704,36	5,21	20	10,71
50	690,68	1,94	16,6	17
60	668,61	3,24	14,4	13,25
70	658,13	1,52	14	2,78

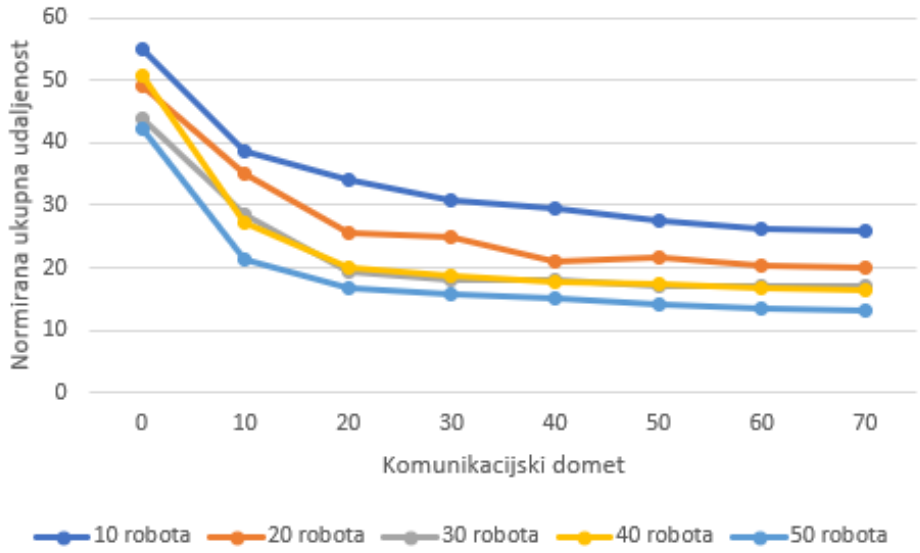
Tablica 10 prikazuje vrijednosti udaljenosti za grupu od 40 mobilnih robota. Vrijednosti udaljenosti se ponašaju jednako kao i prethodne grupe robota. Kod ove skupine robota konvergencija udaljenosti je oko vrijednosti dometa od 50.

Tablica 11. Tablica prijeđenih udaljenosti i broja iteracija za 50 robota kod nepotpune komunikacije

Domet	Udaljenost	Smanjenje udalj. [%]	Broj iteracija	Smanjenje iter. [%]
0	2116,61	-	102,2	-
10	1062,18	49,82	46,8	54,21
20	845,14	20,43	23,6	49,57
30	782,02	7,46	22,6	4,24
40	757,68	3,11	19,4	14,2
50	707,09	6,68	14,2	26,8
60	669,51	5,31	12	15,49
70	658,6	1,69	11,4	5

Tablica 11 prikazuje vrijednosti udaljenosti za grupu od 50 mobilnih robota. Udaljenosti se smanjuju kako raste vrijednost dometa, kao i u prethodnim grupama. Udaljenost konvergira oko dometa 60.

Na slici 18, prikazane su normirane vrijednosti udaljenosti u ovisnosti o vrijednosti dometa za nepotpunu komunikaciju. Udaljenosti su normirane po broju robota u grupi. Normirane udaljenosti se smanjuju kako raste vrijednost dometa. Vrijednost normirane udaljenosti za istu vrijednost dometa pada kako raste broj robota. Razlog je gušća mreža komunikacije.



Slika 18. Normirne udaljenosti kod nepotpune komunikacije - po komunikacijskom dometu

Tablica 12. Tablica normiranih prijeđenih udaljenosti za 10 robota kod nepotpune komunikacije

Domet	Normirana udalj.	Smanjenje norm. udalj. [%]
0	54,99	-
10	38,6	29,81
20	34,22	11,35
30	30,66	10,4
40	29,57	3,55
50	27,42	7,29
60	26,23	4,32
70	25,91	1,21

Tablica 12 prikazuje vrijednosti normirane udaljenosti za grupu od 10 robota. Najveće smanjenje normirane udaljenosti je s vrijednosti dometa 0 na vrijendost dometa 10 za $29,81\%$. Konvergencija normirane udaljenosti za ovu grupu robota je oko vrijednosti dometa od 60.

Tablica 13. Tablica normiranih prijeđenih udaljenosti za 20 robota kod nepotpune komunikacije

Domet	Normirana udalj.	Smanjenje norm. udalj. [%]
0	48,97	-
10	35,08	28,37
20	25,51	27,28
30	24,88	2,49
40	21,17	14,89
50	21,57	-1,87
60	20,48	5,05
70	20,09	1,91

Tablica 13 prikazuje vrijednosti normirane udaljenosti za grupu od 20 robota. Normirane udaljenosti, kao i kod grupe od 10 robota, se smanjuju povećavanjem vrijednosti dometa, te konvergiraju oko vrijednosti dometa 60.

Tablica 14. Tablica normiranih prijeđenih udaljenosti za 30 robota kod nepotpune komunikacije

Domet	Normirana udalj.	Smanjenje norm. udalj. [%]
0	43,83	-
10	28,64	34,66
20	19,42	32,20
30	18,19	6,28
40	18,14	0,32
50	16,98	6,40
60	17,11	-0,8
70	17,07	0,21

Tablica 14 prikazuje vrijednosti normirane udaljenosti za grupu od 30 robota. Od vrijednosti dometa 0 do vrijednosti dometa 10 smanjenje normirane udaljenosti je u iznosu od $34,66\%$. Daljnjim povećavanjem vrijednosti dometa od 10 do 20 smanjenje normirane vrijednosti je $32,20\%$. Nakon toga vrijednosti smanjenja su manja i normirana udaljenost konvergira oko vrijednosti dometa 50.

Tablica 15. Tablica normiranih prijeđenih udaljenosti za 40 robota kod nepotpune komunikacije

Domet	Normirana udalj.	Smanjenje norm. udalj. [%]
0	50,86	-
10	27,34	46,24
20	19,99	26,89
30	18,58	7,07
40	17,61	5,21
50	17,27	1,94
60	16,71	3,24
70	16,45	1,52

Tablica 15 prikazuje vrijednosti normirane udaljenosti za grupu od 40 robota. Kao i u prethodnim grupama najveći pad vrijednosti normirane udaljenosti je od vrijednosti dometa 0 do vrijednosti dometa 10 i iznositi $46,24\%$. Idući rast vrijednosti dometa od 10 do 20 ima pad vrijednosti normirane udaljenosti od $26,89\%$. Normirane udaljenosti konvergiraju oko vrijednosti dometa 50.

Tablica 16. Tablica normiranih prijeđenih udaljenosti za 50 robota kod nepotpune komunikacije

Domet	Normirana udalj.	Smanjenje norm. udalj. [%]
0	42,33	-
10	21,24	49,82
20	16,9	20,43
30	15,64	7,47
40	15,15	3,11
50	14,14	6,68
60	13,39	5,32
70	13,17	1,69

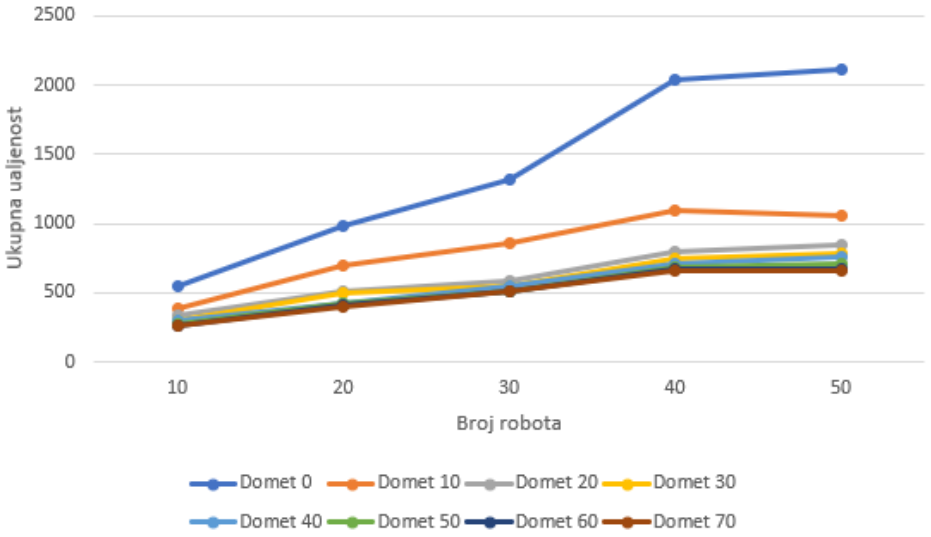
Tablica 16 prikazuje vrijednosti normirane udaljenosti za grupu od 50 robota. Grupa od 50 robota ima najmanje vrijednosti normirane udaljenosti po svim vrijednostima dometa u odnosu na ostale grupe robota. Konvergencija normirane udaljenosti je oko vrijednosti dometa od 60.

Prethodne simulacije prokazale su ispravnost hipoteze kako se povećavanjem vrijednosti dometa poboljšavaju rezultati. U ovoj tezi to je smanjenje ukupne udaljenosti i broja iteracija. Normirane vrijednosti su pokazale kako se povećavanjem broja robota smanjuje prijeđena udaljenost po robotu, što će biti prikazano u nastavku.

Hipoteza za povećavanje broja robota je da će kod većeg broja robota rezultati biti bolji kako se povećava broj robota. Razlog je gušća mreža komunikacije i veća vjerojatnost da će roboti doći u komunikacijski domet jedan s drugim. Simulacije će biti provedene za različite vrijednosti dometa tako da se povećava broj robota. Vrijednosti dometa će biti iste kao i u prethodnim simulacijama kod nepotpune komunikacije: od 0 do 70 s inkrementom 10. Broj robota će biti u grupama od 10 do 50 s inkrementom 10, kao i kod prethodnih simulacija.

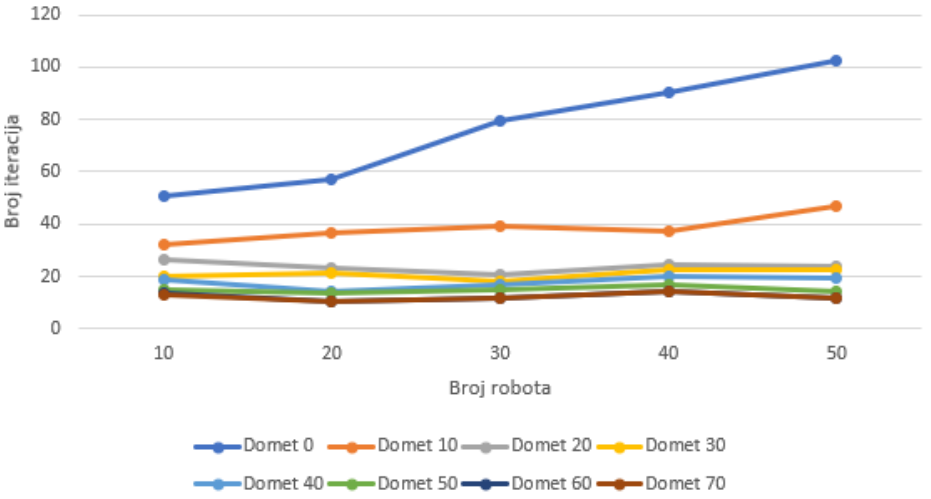
Na slici 19, prikazane su vrijednosti ukupnih udaljenosti za različite vrijednosti dometa po grupama robota. Može se vidjeti kako za svaku grupu robota ukupna udaljenost je veća za manje vrijednosti dometa. Te razlike s povećavaju kako raste broj robota. Kod vrijednosti dometa 0 povećavanje broja robota doprinosi skoro pa jednolikom rastu udaljenosti. Razlog je taj da kod dometa 0 roboti komuni-

ciraju samo kada su roboti na istoj točki. Za ostale vrijednosti dometa koje povećavanjem broja robota rast udaljenosti je blažeg nagiba. Kasnije u tezi će biti prikazane normirane vrijednosti udaljenosti po robotima gdje će se bolje vidjeti utjecaj povećavanja broja robota.



Slika 19. Prijeđene udaljenosti kod nepotpune komunikacije - po broju robota

Na slici 20, prikazani su brojevi iteracija za različite vrijednosti dometa po grupama robota. Kao i kod udaljenosti, broj iteracija za vrijednost dometa 0 raste jednoliko kako se povećava broj robota. Za domet 10 postoji rast ali je značajno manji od vrijednosti dometa 0. Broj iteracija za ostale vrijednosti dometa se ponaša jednoliko s povećavanjem broja robota, te je promjena broja iteracija skoro pa nikakva kako raste broj robota.



Slika 20. Broj iteracija kod nepotpune komunikacije - po broju robota

Tablica 17. Tablica prijeđenih udaljenosti i broja iteracija za domet 0

Br. robota	Udaljenost	Povećanje udalj. [%]	Broj iteracija	Povećanje iter. [%]
10	549,95	-	50,8	-
20	979,44	78,1	56,8	11,81
30	1314,99	34,26	79,6	40,14
40	2034,4	54,7	90,2	13,31
50	2116,6	4,04	102,2	13,31

Tablica 17 prikazuje vrijednosti udaljenosti i broja iteracija za domet 0. Povećavanjem broja robota raste i udaljenost i broj iteracija. Povećavanjem broja robota od 10 do 40 dolazi do značajnih povećanja udaljenosti, a između 40 i 50 robota povećanje je $4,04\%$. Broj iteracija rastu u svim grupama za prosječnu vrijednost od oko 12% izuzev mjerenja od 20 do 30 robota.

Tablica 18. Tablica prijeđenih udaljenosti i broja iteracija za domet 10

Br. robota	Udaljenost	Povećanje udalj. [%]	Broj iteracija	Povećanje iter. [%]
10	386,01	-	32,4	-
20	701,6	81,76	36,4	12,35
30	859,22	22,46	39	7,14
40	1093,71	27,79	37,2	-4,61
50	1062,17	-2,88	46,8	25,8

Tablica 18 prikazuje vrijednosti udaljenosti i broja iteracija za domet 10. Nakon početnog značajnog povećanja udaljenosti od $81,76\%$, daljnja povećanja su $22,46\%$ i $27,79\%$, te na kraju između 40 i 50 robota dolazi do smanjenja od $2,88\%$. Broj iteracija se ne mijenja značajno izuzev između 40 i 50 gdje je povećanje od $25,8\%$.

Tablica 19. Tablica prijeđenih udaljenosti i broja iteracija za domet 20

Br. robota	Udaljenost	Povećanje udalj. [%]	Broj iteracija	Povećanje iter. [%]
10	342,2	-	26,6	-
20	510,24	49,1	23	-13,53
30	582,53	14,16	20,4	-11,3
40	799,54	37,25	24,2	18,62
50	845,14	5,7	23,6	-2,47

Tablica 19 prikazuje vrijednosti udaljenosti i broja iteracija za domet 20. Kao i kod prethodnih vrijednosti dometa, za domet 20 između broja robota 40 i 50 povećanje je $5,7\%$ što je značajno manje od ostalih ($49,1\%, 14,16\%$ i $37,25\%$). Broj iteracija se ponaša slično kao i kod dometa 40, samo s manjim vrijednostima.

Tablica 20. Tablica prijeđenih udaljenosti i broja iteracija za domet 30

Br. robota	Udaljenost	Povećanje udalj. [%]	Broj iteracija	Povećanje iter. [%]
10	306,6	-	20,2	-
20	497,53	62,27	21,5	5,94
30	545,96	9,73	18,2	-14,95
40	743,04	36,09	22,4	23,07
50	782,02	5,25	22,6	0,89

Tablica 20 prikazuje vrijednosti udaljenosti i broja iteracija za domet 30. Povećanje udaljenosti je najveće za povećanje broja robota s 10 na 20 iznosa $62,27\%$. Na kraju između 40 i 50 robota povećanje udaljenosti je $5,25\%$. Broj iteracije se ponaša skokovito s rastom i padom postotnih promjena vrijednosti broja iteracija.

Tablica 21. Tablica prijeđenih udaljenosti i broja iteracija za domet 40

Br. robota	Udaljenost	Povećanje udalj. [%]	Broj iteracija	Povećanje iter. [%]
10	297,73	-	18,6	-
20	423,43	43,18	14	-24,7
30	544,19	28,52	16,8	20
40	704,36	29,43	20	19,04
50	757,68	7,57	19,4	-3

Tablica 21 prikazuje vrijednosti udaljenosti i broja iteracija za domet 40. Povećanje udaljenosti je s 10 na 20 iznosa $43,18\%$, a na kraju između 40 i 50 robota povećanje udaljenosti je $7,575\%$. Broj iteracije se ponaša skokvito s rastom i padom postonih promjena vrijednosti broja iteracija.

Tablica 22. Tablica prijeđenih udaljenosti i broja iteracija za domet 50

Br. robota	Udaljenost	Povećanje udalj. [%]	Broj iteracija	Povećanje iter. [%]
10	274,15	-	14,8	-
20	431,34	57,33	13,4	-9,45
30	509,35	18,08	14,6	8,95
40	690,68	35,59	16,6	13,69
50	707,09	2,37	14,2	-14,45

Tablica 22 prikazuje vrijednosti udaljenosti i broja iteracija za domet 50. Vrijednosti udaljenost i broj iteracija se ponašaju slično kao i prethodno opisane vrijednosti za domete veće od 0, uz manje vrijednosti u odnosu na prethodne.

Tablica 23. Tablica prijeđenih udaljenosti i broja iteracija za domet 60

Br. robota	Udaljenost	Povećanje udalj. [%]	Broj iteracija	Povećanje iter. [%]
10	262,29	-	13,6	-
20	409,56	56,14	10,4	-23,53
30	513,43	25,36	11,6	11,54
40	668,31	30,16	14,4	24,14
50	669,51	0,17	12	-16,67

Tablica 23 prikazuje vrijednosti udaljenosti i broja iteracija za domet 60. Vrijednosti udaljenosti rastu povećanjem broja robota od 10 do 40. Povećanjem broja robota od 40 do 50 za domet 60 nema nikakvog rasta vrijednosti udaljenosti.

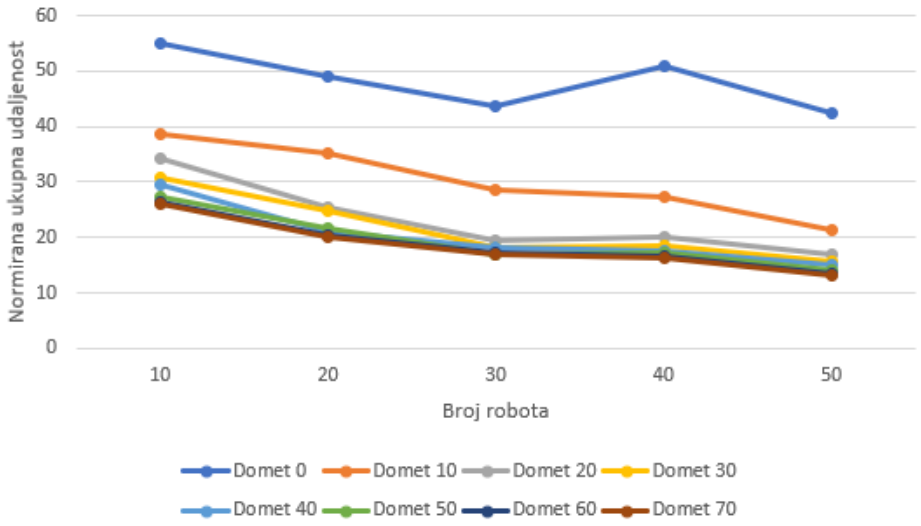
Tablica 24. Tablica prijeđenih udaljenosti i broja iteracija za domet 70

Br. robota	Udaljenost	Povećanje udalj. [%]	Broj iteracija	Povećanje iter. [%]
10	259,11	-	13,2	-
20	401,75	55,05	10,4	-21,21
30	512,33	27,52	11,4	9,6
40	658,13	28,46	14	22,8
50	658,6	0,07	11,4	-18,57

Tablica 24 prikazuje vrijednosti udaljenosti i broja iteracija za domet 70. Vrijednosti udaljenosti i broja iteracija su skoro identične kao i vrijednosti udaljenosti i broja iteracija za domet 60. Iz toga se može potvrditi prethodni zaključak da oko dometa 60 vrijednosti udaljenosti i broja iteracija konvergiraju. Također se može zaključiti kako za vrijednosti dometa od 60 i naviše dolazi do konvergencije rješenja za 40 mobilnih robota.

Na slici 21, prikazane su normirane vrijednosti udaljenosti po broju robota. Za domet 0 normirane vrijednosti udaljenosti se smanjuju od 10 do 30 mobilnih robota. Između 30 i 40 mobilnih robota normirana udaljenost se povećava u iznosu $16,03\%$, te nakon toga pada. Ponašanje normiranih vrijednosti za domet 0 je jedino koje odstupa od hipoteze da povećavanjem broja robota vrijednosti padaju. Razlog je nedostatak komunikacije tj. mogućnosti komunikacije, jer roboti mogu komunicirati samo kada se nalaze na istoj točki. Za domet 10 normirane vrijednosti udaljenosti jednoliko se smanjuju povećavanjem broja robota. Normirane vrijednosti udaljenosti za ostale domete se ponašaju identično, samo se povećavanjem dometa smanjuju normirane vrijednosti.

U tablici 25 su prikazane normirane vrijednosti udaljenosti za domet 0. Kao što je prethodno napisano između 30 i 40 mobilnih robota došlo je do povećanja normirane udaljenosti u iznosu od $16,03\%$. To je jedino odstupanje od ostatka normiranih udaljenosti za domet 0.



Slika 21. Normirnane prijeđene udaljenosti kod nepotpune komunikacije - po broju robota

Tablica 25. Tablica normiranih prijeđenih udaljenosti za domet 0

Br. robota	Normirana udalj.	Smanjenje norm. udalj. [%]
10	54,99	-
20	48,97	10,95
30	43,83	10,49
40	50,86	-16,03
50	42,33	4,044

Tablica 26. Tablica normiranih prijeđenih udaljenosti za domet 10

Br. robota	Normirana udalj.	Smanjenje norm. udalj. [%]
10	38,6	-
20	35,08	9,12
30	28,64	18,35
40	27,34	4,53
50	21,24	22,31

Tablica 26. sadrži vrijednosti normiranih udaljenosti za domet 10. Iz te tablice se može vidjeti kako se normirane udaljenosti smanjuju povećavanjem broja robota.

Tablica 27. Tablica normiranih prijeđenih udaljenosti za domet 20

Br. robota	Normirana udalj.	Smanjenje norm. udalj. [%]
10	34,22	-
20	25,51	25,45
30	19,42	23,88
40	19,98	-2,94
50	16,9	15,43

Tablica 27. sadrži vrijednosti normiranih udaljenosti za domet 20. Normirane udaljenosti za domet 20 također se smanjuju kako se povećava broj robota. Izuzev blagog rasta od $2,94\%$ između 30 i 40 robota.

Tablica 28. Tablica normiranih prijeđenih udaljenosti za domet 30

Br. robota	Normirana udalj.	Smanjenje norm. udalj. [%]
10	30,66	-
20	24,88	18,86
30	18,20	26,84
40	18,57	-2,074
50	15,64	15,8

Tablica 28 sadrži vrijednosti normiranih udaljenosti za domet 30. Za domet 30 povećavanje broja robota padaju vrijednosti normiranih udaljenosti. Izuzev blagog rasta od $2,07\%$ između 30 i 40 robota.

Tablica 29. Tablica normiranih prijeđenih udaljenosti za domet 40

Br. robota	Normirana udalj.	Smanjenje norm. udalj. [%]
10	29,57	-
20	21,17	28,4
30	18,14	14,32
40	17,61	2,93
50	15,15	13,94

Tablica 29 sadrži vrijednosti normiranih udaljenosti za domet 40. Normirane vrijednosti za domet 40 smanjuju se povećavanjem broja robota.

Tablica 30. Tablica normiranih prijeđenih udaljenosti za domet 50

Br. robota	Normirana udalj.	Smanjenje norm. udalj. [%]
10	27,42	-
20	21,57	21,33
30	16,98	21,28
40	17,27	-1,69
50	14,14	18,01

Tablica 30 sadrži vrijednosti normiranih udaljenosti za domet 50. Kao i kod dometa 20 i 30 vrijednosti normiranih udaljenosti padaju povećavanjem broja robota, izuzev blagog rasta od $1,69\%$ za pomak s 30 do 40 robota.

Tablica 31. Tablica normiranih prijeđenih udaljenosti za domet 60

Br. robota	Normirana udalj.	Smanjenje norm. udalj. [%]
10	26,23	-
20	20,48	21,93
30	17,11	16,42
40	16,71	2,37
50	13,39	19,86

Tablica 32. Tablica normiranih prijeđenih udaljenosti za domet 70

Br. robota	Normirana udalj.	Smanjenje norm. udalj. [%]
10	25,91	-
20	20,08	22,47
30	17,07	14,98
40	16,45	3,65
50	13,17	19,94

Tablice 31 i 32 imaju skoro identične vrijednosti normiranih udaljenosti. Povećavanjem broja robota padaju vrijednosti normiranih udaljenosti.

Kroz ove simulacije potvrđena je hipoteza da se povećavanjem broja robota poboljšavaju rezultati, tj. smanjuju se prijeđene udaljenosti i brojevi iteracija. Za domete ≥ 60 dolazi do konvergencije udaljenosti za ≥ 50 robota.

5. Nastavak i nadogradnja radnje

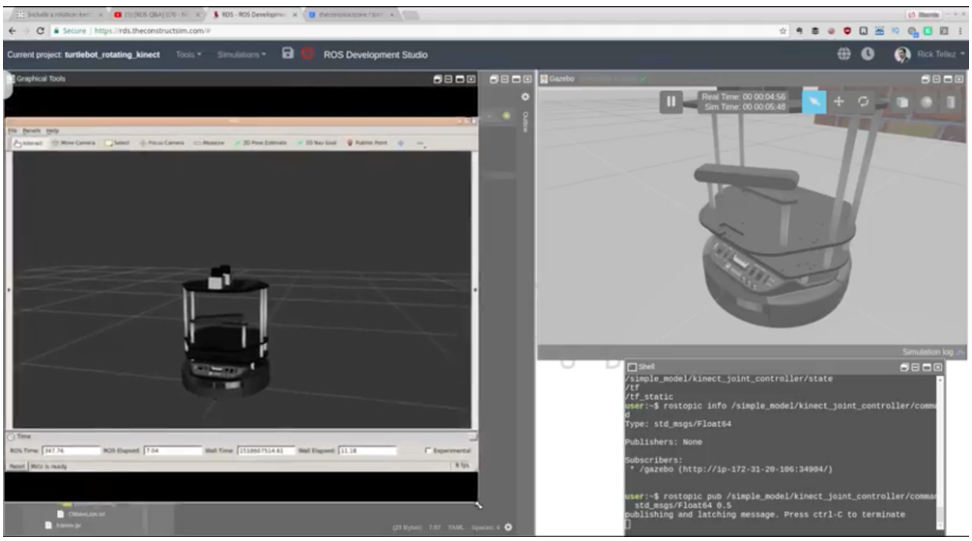
Nastavak ovog diplomskog rada bi bila emulacija aukcijskog algoritma na dva mobilna robota u laboratoriju. Jedan od tipova mobilnih robota koji bi bio prigodan je TurtleBot. TurtleBot je mobilni robot koji može raditi s ROS-om (eng. *Robot Operating System*) programskim alatom otvorenog koda. ROS-a je prigodan programski alata za istraživanja i edukaciju zbog svoje modularnosti i velikog broja literature i ostalih edukacijskih materijala.

Na slici 22, prikazan je TurtleBot mobilni robot koji je opremljen s 2D/3D senzorom i prijenosnim računalom s kojeg se pokreće program.



Slika 22. TurtleBot mobilni robot

Na slici 23, prikazano je sučelje ROS-a, te se može vidjeti primjer programa za interakciju s ROS-om.



Slika 23. Primjer programa u ROS-u

6. Zaključak

Mobilni roboti mogu smanjiti opseg posla zaposlenicima bolnica ili domova za starije osobe. Područje mobilne robotike doživjet će značajan razvoj u budućnosti, te će se uspješno integrirati u realni svijet. Za njihovo optimalno ponašanje potrebno je uspješno riješiti problem dodijele zadataka.

Odabirom pravilnog algoritma za problem dodjele zadataka robotskim sustavim mogu se dobiti optimalni rezultati. Leksikografski algoritam dodjeljuje zadatke koristeći leksikografsku metodu, tj.prvom robotu se dodjeljuje prvi zadatak i tako redom do kraja. Taj algoritam nema povoljne rezultate jer nema prave metode odlučivanja. Koristeći First come first serve (FCFS) algoritam s jednostavnom metodom odlučivanja značajno su se poboljšali rezultati. FCFS algoritam dodjeljuje zadatke uzimajući u obzir minimalne udaljenosti dostupnih zadataka. Dobiveni rezultati bili su povoljniji od onih dobivenih leksikografskim algoritmom, ali nisu doprinjeli optimizaciji svih udaljenosti budući da nekim robotima su dodjeljeni zadaci s nepovoljnim udaljenostima. Kako bi dobili optimalne rezultate koristilo se mađarski algoritam razvijen u 50-im godinama 20. stoljeća. Mađarski algoritam uzima u obzir raspodjelu svih udaljenosti koje će roboti prijeći te time smanjuje prosjećnu udaljenost koju roboti prijeđu. Prethodno opisani algoritmi koristili su potpunu komunikaciju.

Za nepotpunu komunikaciju koristio se aukcijski algoritam. Hipoteza je bila da će se povećanjem vrijednosti komunikacijskog dometa ili povećanjem broja robota dobiti bolji rezultati. Provedenim simulacijama u programskom alatu Matlab za prostor veličine 100×100 za različite vrijednosti komunikacijskog dometa došlo je do konvergencije udaljenosti oko vrijednosti dometa 60. Kod mjerenja utjecaja broja robota konvergencija rezultata je bila kod vrijednosti od 40 robota. Dobivenim rezultatima potvrđene su hipoteze. Povećavanjem dometa roboti pokrivaju veći dio prostora i povećava se vjerojatnost komunikacije. Također povećavanjem broja robota raste njihova gustoća u prostoru te opet dolazi do povećavanja vjerojatnosti komunikacije.

Prikazani algoritmi mogu se primjeniti na stvarnim mobilnim robotima (poput TurtelBot-a) koristeći neke od dostupnih programskih alata (npr. ROS)

Literatura

[1] B. P. Gerkey and M. J. Mataric, "A framework for studying multi-robot task allocation," 2003.

[2] S. Giordani, M. Lujak, and F. Martinelli, "A distributed algorithm for the multi-robot task allocation problem," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pp. 721–730, Springer, 2010.

[3] M. Lujak, S. Giordani, A. Omicini, and S. Ossowski, "Decentralizing coordination for scalable and dynamic task assignment in open vehicle fleets," *Complexity*, 2019.

[4] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the society for industrial and applied mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[5] H. W. Kuhn, "Variants of the hungarian method for assignment problems," *Naval Research Logistics Quarterly*, vol. 3, no. 4, pp. 253–258, 1956.

[6] D. P. Bertsekas, "Auction algorithms.," *Encyclopedia of optimization*, vol. 1, pp. 73–77, 2009.

PRILOG

Potpuna komunikacija

```
%task_assign.m
clear all
close all
clc
format compact
N_promp ='Specify the number of robots ';
N = input(N_promp);
if isempty(N)
N = 40;
end
dim_promp ='Sepcify the dimension of space ';
dim =input(dim_promp);
if isempty(dim)
dim = 100;
end
max_speed_promp ='Specify the speed of robots ';
max_speed =input(max_speed_promp);
if isempty(max_speed)
max_speed = 5;
end
n_promp ='Specify the task assignment method: 1. Lexicography, 2.
   FCFS, 3. Munkres ';
n = input(n_promp);
[F,R,T,c,v] = init(N,dim,max_speed);
```

```
24 [T1,A,M,cost] = assignment(R,T,c,dim,n,N);
25 dist=movement(R,T1,F,c,v,dim,N);
26 U = cost % udaljenost svih robota
27 MU = min(A) % min udaljenost
28 MaxU = max(A) %max udaljenost
29 AU = mean(A) %prosjecna udaljenost
```

```
1 %init.m
2 function [F,R,T,c,v] = init(N,dim,max_speed)
3 R = dim*rand(N,2);
4 T = dim*rand(N,2);
5 v = max_speed*ones(N,1);
6 r = size(R,1);
7 t = size(T,1);
8 F = [];
9 c = rand(N,3);
10 if r < t
11 F = zeros(r,1);
12 else
13 F = zeros(t,1);
14 end
15 end
```

```
1 %assignment.m
2 function [T1,A,M,cost] = assignment(R,T,c,dim,n,N)
3 T1 = [];
4 switch n
5 case 1
6 T1 = T;
7 A = distance(R,T1,N);
```

```
cost = sum(A);
M = [];
case 2
[A,M] = fcfs(R,T,c,dim);
T1 = sort1(T,M);
cost = sum(A);
case 3
[D2] = distance2(R,T,c,dim);
[M, cost] = munkres(D2);
T1 = sort1(T,M);
A = distance(R,T1,N);

end
end
```

```
%distance.m
function A = distance(R,T,N)
A=[];
for ind = 1:N
dist = sqrt((R(ind,1)-T(ind,1))^2+(R(ind,2)-T(ind,2))^2);
A = [A; dist];
end
```

```
%fcfs.m
function [A,M] = fcfs(R,T,c,dim)
A = [];
M =[];
scatter(R(:,1),R(:,2),[],c)%plot(R(:,1),R(:,2),'o')%c,
grid on
hold on
```

```
scatter(T(:,1),T(:,2),[],c,'X')%plot(T,'x')%c,
xlim([0 dim])
ylim([0 dim])
hold off
for i=1:length(R)
D = ones(size(R,1),1)*NaN; %vektor udaljenosti
B=find(~isnan(T(:,1)))'; %skup dostupnih taskova
for j=B
D1 = sqrt((R(i,1)-T(j,1))^2+(R(i,2)-T(j,2))^2);
D(j) = D1;
end
[A1,P1] = min(D);
A = [A;A1]; %matrica dodjeljenih udaljenosti
M = [M ;[i P1]];%assignent matrix
T(P1,:) = NaN; %uklanjanje dodjeljenog taska
R(i,:) = NaN; %uklanjanje dodjeljenog robota
end
M = transpose(M(:,2));
end
```

```
%distance2.m
function[D2] = distance2(R,T,c,dim)
D2 = [];
D2j = [];
scatter(R(:,1),R(:,2),[],c)%plot(R(:,1),R(:,2),'o')%c,
grid on
hold on
scatter(T(:,1),T(:,2),[],c,'X')%plot(T,'x')%c,
xlim([0 dim])
```

```
ylim([0 dim])
hold off
for i=1:length(R)
for j=1:length(T)
D1 = sqrt((R(i,1)-T(j,1))^2+(R(i,2)-T(j,2))^2);
D2j(j) = D1;
end
D2=[D2;D2j];
end
```

```
%munkres.m
function [assignment,cost] = munkres(costMat)
% Reference:
% "Munkres' Assignment Algorithm, Modified for Rectangular Matrices",
% http://csclab.murraystate.edu/bob.pilgrim/445/munkres.html
% version 2.3 by Yi Cao at Cranfield University on 11th September 2011
assignment = zeros(1,size(costMat,1));
cost = 0;
validMat = costMat == costMat & costMat < Inf;
bigM = 10^(ceil(log10(sum(costMat(validMat))))+1);
costMat(~validMat) = bigM;
% costMat(costMat~=costMat)=Inf;
% validMat = costMat<Inf;
validCol = any(validMat,1);
validRow = any(validMat,2);
nRows = sum(validRow);
nCols = sum(validCol);
```

```
18 n = max(nRows,nCols);
19 if ~n
20 return
21 end
22 maxv=10*max(costMat(validMat));
23 dMat = zeros(n) + maxv;
24 dMat(1:nRows,1:nCols) = costMat(validRow,validCol);
25 minR = min(dMat,[],2);
26 minC = min(bsxfun(@minus, dMat, minR));
27 zP = dMat == bsxfun(@plus, minC, minR);
28 starZ = zeros(n,1);
29 while any(zP(:))
30 [r,c]=find(zP,1);
31 starZ(r)=c;
32 zP(r,:)=false;
33 zP(:,c)=false;
34 end
35 while 1
36 if all(starZ>0)
37 break
38 end
39 coverColumn = false(1,n);
40 coverColumn(starZ(starZ>0))=true;
41 coverRow = false(n,1);
42 primeZ = zeros(n,1);
43 [rIdx, cIdx] = find(dMat(~coverRow,~coverColumn)==bsxfun(@plus,minR
    (~coverRow),minC(~coverColumn)));
44 while 1
45 cR = find(~coverRow);
```

```
46 cC = find(~coverColumn);
47 rIdx = cR(rIdx);
48 cIdx = cC(cIdx);
49 Step = 6;
50 while ~isempty(cIdx)
51 uZr = rIdx(1);
52 uZc = cIdx(1);
53 primeZ(uZr) = uZc;
54 stz = starZ(uZr);
55 if ~stz
56 Step = 5;
57 break;
58 end
59 coverRow(uZr) = true;
60 coverColumn(stz) = false;
61 z = rIdx==uZr;
62 rIdx(z) = [];
63 cIdx(z) = [];
64 cR = find(~coverRow);
65 z = dMat(~coverRow,stz) == minR(~coverRow) + minC(stz);
66 rIdx = [rIdx(:);cR(z)];
67 cIdx = [cIdx(:);stz(ones(sum(z),1))];
68 end
69 if Step == 6
70 [minval,rIdx,cIdx]=outerplus(dMat(~coverRow,~coverColumn),minR(~
      coverRow),minC(~coverColumn));
71 minC(~coverColumn) = minC(~coverColumn) + minval;
72 minR(coverRow) = minR(coverRow) - minval;
73 else
```

```
74  break
75  end
76  end
77
78  rowZ1 = find(starZ==uZc);
79  starZ(uZr)=uZc;
80  while rowZ1>0
81  starZ(rowZ1)=0;
82  uZc = primeZ(rowZ1);
83  uZr = rowZ1;
84  rowZ1 = find(starZ==uZc);
85  starZ(uZr)=uZc;
86  end
87  end
88  % Cost of assignment
89  rowIdx = find(validRow);
90  colIdx = find(validCol);
91  starZ = starZ(1:nRows);
92  vIdx = starZ <= nCols;
93  assignment(rowIdx(vIdx)) = colIdx(starZ(vIdx));
94  pass = assignment(assignment>0);
95  pass(~diag(validMat(assignment>0,pass))) = 0;
96  assignment(assignment>0) = pass;
97  cost = trace(costMat(assignment>0,assignment(assignment>0)));
98  function [minval,rIdx,cIdx]=outerplus(M,x,y)
99  ny=size(M,2);
100 minval=inf;
101 for c=1:ny
102 M(:,c)=M(:,c)-(x+y(c));
```

```
minval = min(minval,min(M(:,c)));
end
[rIdx,cIdx]=find(M==minval);
```

```
%sort1.m
function T1 = sort1(T,M)
T1=transpose(T);
T1=T1(:,M);
T1=transpose(T1);
end
```

```
%movement.m
function dist = movement(R,T1,F,c,v,dim,N)
T = T1;
scatter(R(:,1),R(:,2),[],c)%plot(R(:,1),R(:,2),'o')%c,
hold on
scatter(T(:,1),T(:,2),[],c,'X')%plot(T,'x')%c,
xlim([0 dim])
ylim([0 dim])
hold off
while ~prod(F)%tstep<t&&~prod(found)
for ind = 1:N
dist = sqrt((R(ind,1)-T(ind,1))^2+(R(ind,2)-T(ind,2))^2)
% t_rob = dist/v(ind)%number of time periods until the target
%t_rob-v(ind)
if dist<= v(ind)%(t_rob<1)
R(ind,:) = T(ind,:);
F(ind) = 1
else
alpha = atan((T(ind,2)-R(ind,2))/(T(ind,1)-R(ind,1)))*180/pi
```

```
20 if (T(ind,2)-R(ind,2))>=0
21 T(ind,2)-R(ind,2)
22 if (T(ind,1)-R(ind,1))>=0%first quadrant
23 T(ind,1)-R(ind,1)
24 R(ind,2) = v(ind)*abs(sind(alpha))+R(ind,2)
25 R(ind,1) = v(ind)*abs(cosd(alpha))+R(ind,1)
26 elseif (T(ind,1)-R(ind,1))<0%second quadrant
27 T(ind,1)-R(ind,1)
28
29 R(ind,2) = R(ind,2)+v(ind)*sind(180-abs(alpha))
30 R(ind,1) = R(ind,1)+v(ind)*cosd(180-abs(alpha))
31 end
32 elseif (T(ind,2)-R(ind,2))<0%third
33 if (T(ind,1)-R(ind,1))>=0 %fourth quadrant
34 R(ind,2) = R(ind,2)+v(ind)*sind(360-abs(alpha))
35 R(ind,1) = R(ind,1)+v(ind)*cosd(360-abs(alpha))
36 % R(ind,2) = -v(ind)*abs(sind(alpha+270))+R(ind,2)
37 % R(ind,1) = -v(ind)*abs(cosd(alpha+270))+R(ind,1)
38 elseif(T(ind,1)-R(ind,1))<0% Third quadrant
39 R(ind,2) = R(ind,2)+v(ind)*sind(abs(alpha)+180)
40 R(ind,1) = R(ind,1)+v(ind)*cosd(abs(alpha)+180)
41 end
42 end
43
44 end
45 end
46 %figure
47 axis([0, dim, 0, dim])
48 scatter(R(:,1),R(:,2),[],c)%plot(R(:,1),R(:,2),'o')%c,
```

```
49 hold on
50 axis([0, dim, 0, dim])
51 scatter(T(:,1),T(:,2),[],c,'x')%plot(T,'x')%c,
52 pause(0.05)
53 %      plotvec(R',c,'o')
54 hold on
55 %      plotvec(T',c,'x')
56 %  xlim([-1 1])
57 %  ylim([-1 1])
58 xlim([0 dim])
59 ylim([0 dim])
60 hold off
61 % tstep = tstep+1;
62 end
```

Nepotpuna komunikacija

```
1  %ta_domet.m
2  clear all
3  close all
4  clc
5  format compact
6  global dim step range
7
8  N_promp ='Specify the number of robots ';
9  N = input(N_promp);
10 if isempty(N)
11 N = 20;
12 end
```

```
dim_promp ='Sepcify the dimension of space ';
dim =input(dim_promp);
if isempty(dim)
dim = 100;

end

max_speed_promp ='Specify the speed of robots ';
max_speed =input(max_speed_promp);
if isempty(max_speed)
max_speed = 5;
end
step = max_speed;

range_promp ='Specify the range of robots ';
range =input(range_promp);
if isempty(range)
range = 0.001;
end
[F,R,T,c,v] = init(N,dim,max_speed) %init.m isti kao i kod potpune
   komunikacije
[totaldist, iter] = domet3(R,T);

disp(num2str(totaldist,'%.2f'))
disp(iter)
```

```
%domet3.m
function [totaldist, iter] = domet3(R,T)
global N no_messages assignment p a arrived_Target
```

```
4  R1=R;
5  T1=T;
6  Xr1=R(:,1);% initial x coordinates of the robots
7  Yr1=R(:,2);% initial y coordinates of the robots
8  Xt1=T(:,1); %x coordinates for targets
9  Yt1=T(:,2); %y coordinates for targets
10 totaldist=0;
11 %dr=ipdm(R1,'Subset','Maximum','limit',range); % the distance among
       the robots within sensor range
12 a=ipdm(R1,T1);% matrix of distances between all the robots and all
      the targets
13
14
15 iter=0;
16 [N, ~]=size(a);
17 p=zeros(N,N);
18 epsilon = 1/(N+1); %epsilon increment
19 assignment =Inf*ones(N,N);%assignment(i,k)=j means that obj j is
      assigned to person i in opinion of k
20 %arrived_Target=zeros(N,N);
21 distances=inf;
22 no_messages=0;
23 arrived_Target=zeros(N,N);
24
25 tstart=tic;
26 telapsed=toc(tstart);
27 while(distances>0) %do while at least one of the agents are not
      assigned to their respective targets
28 iter=iter+1;
```

```
update_range(Xr1,Yr1,Xt1,Yt1);
assignemnt_range(epsilon);
[totaldist,Xr1,Yr1,distances]=movement_range(Xr1,Yr1,Xt1,Yt1,
    totaldist);
telapsed=toc(tstart);
end
iter
```

```
%update_range.m
function update_range(Xr1,Yr1,Xt1,Yt1)
global N no_messages assignment p range a unAssignedPeople
    pers_inrange arrived_Target
%Update the Communication network on the base of the new positions
    %%%
dr=ipdm([Xr1, Yr1],'Subset','Maximum','limit',range); % read the
    sensory readings of the robot to the other robots
%dr is a matrix of the distances between the robots under the
    sensor
%%%%%%%%%%%%%%%%%%%%%%%%                                    %range limit

%Update of information in the new communication graph of ROBOTS
pers_inrange=cell(N,1); % indices of the persons in range including
     himself
pers_inrange_neigh=cell(N,1);
pers_inrange_new=cell(N,1);
unAssignedPeople=cell(N,1);
a=ipdm([Xr1, Yr1],[Xt1,Yt1]); %funkcija ipdm je dostpuna na
    internetu

```

```
16 %ROBOTS neighbors update
17 for i=1:N %for different groups not for each agent in the system,
     this loop exchanges the bids with
18 %i
19 pers_inrange{i}=find(dr(i,:)~=inf);                              %
     indices of the persons in range including himself
20 while (isequal(pers_inrange{i},pers_inrange_neigh{i}))        %there
      will always be at least 1 person in range-at least himself
21 [~,pers_inrange_loc]=find(dr(pers_inrange{i},:)~=Inf);      %All the
      agents in the network transmit also the bids also of the other
22 pers_inrange_local=unique(pers_inrange_loc)';                  %agents
      which are  connected only indirectly to
23 pers_inrange_neigh{i}=union(pers_inrange_neigh{i},
     pers_inrange_local);  %the agent i over some third agent
24 pers_inrange_new{i}=union(pers_inrange{i}, pers_inrange_neigh{i});
25 [~,pers_inrangex]=find(dr(pers_inrange_new{i},:)~=Inf);
26 pers_inrange{i}=(unique(pers_inrangex))';
27 no_messages=no_messages+length(pers_inrange{i});
28 end
29 if (length(pers_inrange{i})>1)
                                          %price update of all the
     agents in the range for all the objects.
30 %Without this part, the agents will know the availability of the
31 %target only when they arrive to the range of the
32 %agent on the target (if any)
33 for obj=1:N
34 % obj
35 p(obj,pers_inrange{i})=min(p(obj,pers_inrange{i}));
36 p_min=min(p(obj,pers_inrange{i}))    ;
```

```
37 no_messages=no_messages+(length(pers_inrange{i})-1);
38 p(obj,pers_inrange{i})=p_min;
39 [agents_assigned,~]=find(assignment(:,pers_inrange{i})==obj);% if
   more agents assigned to the same target,erase assignment if not
   arrived to the target
40 ag_sorted=sort(unique(agents_assigned));                    %since
   only the agent which is on the target remains assigned to the
   target,the flag is necessary
41 if length(ag_sorted)>1
42 % assignment
43 assignment(ag_sorted,pers_inrange{i})=Inf;
44 end
45 if(p_min==-Inf)
46 owner=find(arrived_Target(obj,:));
47 if(numel(owner)==1)
48 assignment(owner,pers_inrange{i})=obj;
49 end
50 end
51 end
52 end
53 end
54 end
```

```
1 %assignment_range.m
2 function assignemnt_range(epsilon)
3 global N no_messages assignment p a unAssignedPeople pers_inrange
4 for i=1:N % if there are other targets closer than the assigned one
   , break the assignment
5 % i
```

```
% pers_inrange{i}
unAssignedPeople{i}=find(isinf(diag(assignment(pers_inrange{i},
    pers_inrange{i}))));
%  a
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for brr=1:length(pers_inrange{i})
% brr
% assignment(pers_inrange{i}(brr),pers_inrange{i}(brr))
if assignment(pers_inrange{i}(brr),pers_inrange{i}(brr))~=Inf
mintar=assignment(pers_inrange{i}(brr),pers_inrange{i}(brr));

for brotar=1:N
if (a(pers_inrange{i}(brr),brotar)< a( pers_inrange{i}(brr),mintar)
    )
closetar=brotar;
if a(pers_inrange{i}(brr),brotar)>0
if (p(closetar,pers_inrange{i}(brr))~=-Inf)%
unAssignedPeople{i}=union(unAssignedPeople{i},brr);
assignment(pers_inrange{i}(brr),pers_inrange{i})= Inf;
end
else
if (p(closetar,pers_inrange{i}(brr))~=-Inf)
unAssignedPeople{i}=union(unAssignedPeople{i},brr);
assignment(pers_inrange{i}(brr),pers_inrange{i})= Inf;
end
end
end
end
end
```

```
33 end
34 %%%%
```

```
    bid on that object
59  if(~isempty(indices))                                    % if
    there are some bids
60  [lowestBidForJ,i_j]= min(temp(3,indices));               % find the
     minimum bid
61  no_messages=no_messages+(length(pers_inrange{i})-1);
62  [OldOptPersI,IthPersInRange]=find(assignment(:,pers_inrange{i})==j)
    ;
63  if(~isempty(OldOptPersI))
64  for ind=1:length(OldOptPersI)
65  assignment(OldOptPersI(ind),pers_inrange{i}(IthPersInRange(ind)))=
    Inf;
66  %no_messages=no_messages+1;
67  no_messages=no_messages+(length(pers_inrange{i})-1);
68  end
69  end
70  WinPerson= temp(1,indices(i_j));
71  no_messages=no_messages+1;
72  assignment(WinPerson,pers_inrange{i})=j;
73  p(j,pers_inrange{i})=p(j,pers_inrange{i})+lowestBidForJ;
74  no_messages=no_messages+(length(pers_inrange{i})-1);
75  end
76  end
77
78  [unAssignedPeople{i}, ~]=find(isinf(diag(assignment(pers_inrange{i
    },pers_inrange{i}))));
79  %no_messages=no_messages+length(pers_inrange{i});
80  %temp=zeros(3,length(unAssignedPeople{i}));
81  end
```

```
82 end
83 end
```

```
1  %movement_range.m
2  function [totaldist,Xr1,Yr1,distances]=movement_range(Xr1,Yr1,Xt1,
       Yt1,totaldist)
3  global N assignment dim step a pers_inrange p arrived_Target
       no_messages
4
5  dist_step=zeros(N,2);
6  % assignment
7  a=ipdm([Xr1, Yr1],[Xt1,Yt1]);
8  for mm=1:N
9  % mm
10 if ((assignment(mm,mm)~=Inf)&& a(mm,assignment(mm,mm))>0)
11 n =assignment(mm,mm);
12 dist_step(mm,1)= mm; % order all the agents in respect to the
       distances from them to their assigned targets
13 dist_step(mm,2)=a(mm,n);
14 end
15 %dist_step
16 end
17
18 [~,ord_values]=sort(dist_step);
19 movement_ord=ord_values(:,2);
20
21 for mov=1:length(movement_ord)
22 %  movement_ord(mov)
23 ii=movement_ord(mov);%agent ii, robot
```

```
24 % assignment(ii,ii)
25 if (assignment(ii,ii)~=Inf)
26 jj =assignment(ii,ii); % target
27 %a
28 if (a(ii,jj)>0)
29 %             % angle alpha of movement-all the possible movement
      directions- alltogether 4:
30 %
31 %
32 if a(ii,jj) <= step % if robot is close to the target, closer than
      the step size
33 Xr1(ii)=Xt1(jj);
34 Yr1(ii)=Yt1(jj);
35 totaldist=totaldist+a(ii,jj);
36 aa=ipdm([Xr1, Yr1],[Xt1,Yt1]);
37 for z=1:N
38 aaa = aa(:,z)==0;
39 if sum(aaa)>1
40 Xr1(ii)=Xt1(jj)+0.001;
41 end
42
43 end
44 if(isempty(find(arrived_Target(jj,:)==1, 1)))%||(arrived_TargetT(ii
      ,jj)==1)) %||(arrived_Target(j,i)==1)
45 arrived_Target(jj,ii)=1;
46 %arrived_TargetT(ii,jj)=1;
47 %assignmentT(jj,persT_inrange{jj})=ii;
48 p(jj,pers_inrange{ii})=-Inf;
49 %pT(ii,persT_inrange{jj})=-Inf;
```

```
50 no_messages=no_messages+length(pers_inrange{ii});
51 else
52 assignment(ii,pers_inrange{ii})=Inf;
53 p(jj,pers_inrange{ii})=-Inf;
54 no_messages=no_messages+length(pers_inrange{ii});
55 end
56 %
57 %
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
58 else
59 if ((Yt1(jj)-Yr1(ii)>0) &&((Xt1(jj)-Xr1(ii))>0)) %1st quadrant
60 alpha=atan((Yt1(jj)-Yr1(ii))/(Xt1(jj)-Xr1(ii))); %arctangent
61 %
62 else if (Yt1(jj)-Yr1(ii)<0) &&((Xt1(jj)-Xr1(ii))>0)% IV quadrant
63 alpha=(3*pi/2)+ atan((abs(Xt1(jj)-Xr1(ii)))/(abs((Yt1(jj)-Yr1(ii)))
   ));
64 %
65 else if (Yt1(jj)-Yr1(ii)>0) &&((Xt1(jj)-Xr1(ii))< 0)%II quadrant
66 alpha=pi/2+ atan((abs(Xt1(jj)-Xr1(ii)))/(abs(Yt1(jj)-Yr1(ii))));
67 %
68 else if (Yt1(jj)-Yr1(ii)<0) &&(Xt1(jj)-Xr1(ii)< 0)% III quadrant
69 alpha=pi+ atan((abs(Yt1(jj)-Yr1(ii)))/(abs(Xt1(jj)-Xr1(ii))));
70
71
72 else if((Yt1(jj)-Yr1(ii)==0) &&((Xt1(jj)-Xr1(ii))>0)) %1st quadrant
73 alpha=0 ; %arctangent
74 else if (((Yt1(jj)-Yr1(ii)>0) &&((Xt1(jj)-Xr1(ii))== 0)))%II
   quadrant
75 alpha=pi/2;
```

```
76 else if (((Yt1(jj)-Yr1(ii)==0) &&((Xt1(jj)-Xr1(ii))< 0)))% III
   quadrant
77 alpha=pi;
78 else if (Yt1(jj)-Yr1(ii)<0) &&((Xt1(jj)-Xr1(ii))==0)% IV quadrant
79 alpha=3*pi/2;
80 end
81 end
82 end
83 end
84 end
85 end
86 end
87 end
88
89 %
90 Xr1(ii)=Xr1(ii)+step*cos(alpha);  % new X coordinates of the robots
91 Yr1(ii)=Yr1(ii)+step*sin(alpha);  % new Y coordinates of the robots
92 totaldist=totaldist+step*sqrt((cos(alpha))^2+(sin(alpha))^2);
93 end
94 end
95 end
96
97 end
98
99 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% draw the movements
100 plot(Xr1,Yr1,'ko',Xt1,Yt1,'.','MarkerSize',12), axis([0, dim, 0,
    dim]) % commented if you want
101 hold on
102 %AXIS([0 100 0 100])
```

```
103 % hold on
104 %             for q=1:N % for each point q representing one robot
105 %                 X1=X+Xr1(q);
106 %                 Y1=Y +Yr1(q);
107 %                 plot(X1,Y1,'r') % plot coordinates of the circle
    around the robot
108 %               %  hold on
109 %                 pause(0.5);
110 %                 X1=0; %reset the values of the coordinates
111 %                 Y1=0;
112 %             end
113 %  Voronoi(Xr1,Yr1)
114 pause(0.05);
115 xlim([0 dim])
116 ylim([0 dim])
117 hold off
118 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
119 %
120
121
122 % New distances of the robots
123
124 a=ipdm([Xr1, Yr1],[Xt1,Yt1]); % updated matrix of distances between
     all the robots and
125 %dr=ipdm([Xr1, Yr1],'Subset','Maximum','limit',range); % read the
    sensory readings of the robot to the other robots
126 %dr is a matrix of the distances between the robots under the
    sensor
127 %range limit
```

```
128 %pers_inrange=cell(N,1);
129
130 %distance measure
131 distances=0;
132 for da=1:N
133 if  assignment(da,da)==Inf
134 distances=Inf;
135 break
136 end
137 distances=distances+a(da, assignment(da,da));
138 end
139 end
```