



UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA



Departman za Energetiku, Elektroniku i Telekomunikacije

Milan Lukić

**ALGORITMI ZA DODELU ZADATAKA
IZVRŠIOCIIMA U BEŽIČNIM MREŽAMA
MIKROKONTROLERSKIH SENZORSKIH
UREĐAJA I AUTONOMNIH ROBOTA**

DOKTORSKA DISERTACIJA

Novi Sad, 2015.



УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
21000 НОВИ САД, Трг Доситеја Обрадовића 6

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР:	
Идентификациони број, ИБР:	
Тип документације, ТД:	Монографска публикација
Тип записа, ТЗ:	Текстуални штампани материјал
Врста рада, ВР:	Докторска дисертација
Аутор, АУ:	Милан Лукић
Ментор, МН:	Др Ладислав Новак, редовни професор
Наслов рада, НР:	Алгоритми за доделу задатака извршиоцима у бежичним мрежама микроконтролерских сензорских уређаја и аутономних робота
Језик публикације, ЈП:	Српски
Језик извода, ЈИ:	Српски/Енглески
Земља публикавања, ЗП:	Република Србија
Уже географско подручје, УГП:	АП Војводина
Година, ГО:	2015
Издавач, ИЗ:	Ауторски репринт
Место и адреса, МА:	Факултет техничких наука, Трг Доситеја Обрадовића 6, Нови Сад
Физички опис рада, ФО: <small>(поглавља/страна/ цитата/табела/слика/графика/прилога)</small>	4/155/164/27/52/29/2
Научна област, НО:	Електроника
Научна дисциплина, НД:	Електроника
Предметна одредница/Кључне речи, ПО:	Бежичне мреже, Сензори, Роботи, Алгоритми, Координација, Комуникација, Енергетска ефикасност
УДК	
Чува се, ЧУ:	Библиотека Факултета техничких наука у Новом Саду, Трг Доситеја Обрадовића 6, 21000 Нови Сад
Важна напомена, ВН:	
Извод, ИЗ:	<p>У бежичној мрежи сензора и робота, сензорски модули врше надзор физичких величина од значаја, а роботи имају улогу извршилаца задатака који им се додељују применом одговарајућег алгоритма. Након детекције догађаја од стране статичких сензорских чворова и прослеђивања информација о догађајима роботима, потребно је доделити задатке роботима на ефикасан начин. Додела задатака врши се у складу са природом различитих сценарија који се могу јавити у пракси. У оквиру дисертације разматран је случај када се конкурентно јавља више догађаја којима је потребно доделити извршиоце. У погледу енергетске ефикасности, у оваквим системима као кључни проблеми јављају се минимизација укупне дужине кретања робота и оптимизација комуникације у мрежи. Од комуникационих протокола за откривање извршилаца, у овој дисертацији представљена су побољшања постојећег <i>iMesh</i> протокола и уведен је нови <i>vCell</i> протокол заснован на локализованом формирању ћелија Воронои дијаграма. Такође, упоређене су перформансе новог протокола са постојећим (правоугаони кворум и <i>iMesh</i>) у густим мрежама, ретким мрежама и мрежама са рупама у топологији. Уз то, уведени су алгоритми за ажурирање локације којима мрежа реагује на кретање робота. Резултати симулација показују да <i>vCell</i> постиже ефикасност близу 100% у налажењу најближег робота у густим мрежама. У ретким мрежама, ефикасност му је до 40% боља у односу на остала решења.</p>



КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

	<p>Као главни резултат у дисертацији приказани су нови алгоритми за доделу робота као извршилаца задатака догађајима, чиме су превазиђени недостаци више до сада познатих решења овог проблема. За задати скуп догађаја и скуп робота, сваком догађају додељен је по један робот који је задужен за обилазак локације догађаја. Током појединачних рунди, роботима је дозвољен обилазак једног догађаја када се врши упаривање, или више догађаја, када се врши секвенцијална додела. У дистрибуираном случају, статички сензорски уређаји детектују догађаје и пријављују их оближњим роботима. Алгоритам <i>PDM</i> који се односи на унапређено упаривање са могућношћу размене партнера, елиминише дугачке ивице које се могу јавити приликом упаривања. Алгоритам <i>SQD</i> за секвенцијалну доделу догађаја роботима итеративно проналази пар робот-догађај са најмањим међусобним растојањем, уврштава изабрани догађај у листу за обилазак изабраног робота и ажурира позицију робота. Такође су предложене генерализације које омогућавају да догађаји буду посећени од стране више робота и које узимају у обзир временска ограничења. Дистрибуирани алгоритам <i>MAD</i>, који је заснован на iMesh информационој структури и локалним аукцијама у роботској мрежи, врши доделу робота догађајима на локализован и енергетски ефикасан начин. Резултати симулација потврђују предности предложених алгоритама у односу на постојећа решења, како у погледу скраћивања дужина путања робота, тако и у продужењу животног времена система.</p>			
Датум прихватања теме, ДП :	25.06.2015.			
Датум одбране, ДО :				
Чланови комисије, КО :	Председник:	др Иван Мезеи, доцент		
	Члан:	др Растислав Струхарик, ванредни професор		
	Члан:	др Александар Менићанин, научни сарадник		
	Члан:	др Бранислав Боровац, редовни професор	Потпис ментора	
	Члан:	др Дејан Вукобратовић, ванредни професор		
	Члан, ментор:	др Ладислав Новак, редовни професор		



KEY WORDS DOCUMENTATION

Accession number, ANO :	
Identification number, INO :	
Document type, DT :	Monograph
Type of record, TR :	Printed text
Contents code, CC :	Doctoral thesis
Author, AU :	Milan Lukić
Mentor, MN :	Ladislav Novak, Ph.D, full professor
Title, TI :	Algorithms for task assignment in wireless networks of microcontroller sensor nodes and autonomous robots
Language of text, LT :	Serbian
Language of abstract, LA :	Serbian/English
Country of publication, CP :	Republic of Serbia
Locality of publication, LP :	Autonomous Province of Vojvodina
Publication year, PY :	2015
Publisher, PB :	Author's reprint
Publication place, PP :	Faculty of Technical Sciences, Trg Dositeja Obradovića 6, 21000 Novi Sad
Physical description, PD : (chapters/pages/ref./tables/pictures/graphs/appendixes)	4/155/164/27/52/29/2
Scientific field, SF :	Electronics
Scientific discipline, SD :	Electronics
Subject/Key words, S/KW :	Wireless networks, Sensors, Robots, Algorithms, Coordination, Communication, Energy efficiency
UC	
Holding data, HD :	Library of the Faculty of Technical Sciences in Novi Sad, Trg Dositeja Obradovica 6, 21000 Novi Sad
Note, N :	
Abstract, AB :	<p>In a typical wireless sensor and robot network, sensor nodes monitor physical values of interest, while robots perform some automated tasks. The tasks are assigned to robots by means of an appropriate algorithm. Upon the occurrence of events which are detected by sensor nodes, the information about the events needs to be delivered to robots. Afterwards, it is necessary to assign tasks to robots in an efficient way. Task assignment is performed according to the nature of different scenarios which might occur in practice. This thesis is focused on the case when multiple events, all of which require to be visited by robots, happen simultaneously. Regarding energy efficiency, the key issues which arise in such systems are minimization of robot travel paths, and optimization of the network traffic. In this thesis, the following service discovery protocols are presented: improvements of the existing <i>iMesh</i> protocol, and the novel <i>vCell</i> protocol, which is based on localized formation of an information structure which resembles Voronoi diagram. Furthermore, the performances of new <i>vCell</i> protocol is compared with the existing protocols (<i>Quorum</i> and <i>iMesh</i>) in dense networks, sparse networks, and networks with holes in topology. Also, location update algorithms are introduced, which deal with robot mobility. The simulations show that <i>vCell</i> achieves nearly 100% success rate in finding the nearest robot in dense networks. In sparse networks, it outperforms the other existing solutions by up to 40%.</p>



KEY WORDS DOCUMENTATION

		<p>As a key contribution, the novel dispatch algorithms have been introduced. Given a set of events and a set of robots, the dispatch problem is to allocate one robot for each event to visit it. In a single round, each robot may be allowed to visit only one event (matching dispatch), or several events in a sequence (sequence dispatch). In a distributed setting, each event is discovered by a sensor and reported to a robot. In this thesis, novel algorithms are presented, which are aimed at overcoming the shortcomings of several existing solutions. Pairwise distance based matching algorithm (<i>PDM</i>) eliminates long edges by pairwise exchanges between matching pairs. Sequence dispatch algorithm (<i>SQD</i>) iteratively finds the closest event-robot pair, includes the event in dispatch schedule of the selected robot and updates its position accordingly. When event-robot distances are multiplied by robot resistance (inverse of the remaining energy), the corresponding energy-balanced variants are obtained. Also, generalizations are introduced which handle multiple visits and timing constraints. Distributed algorithm <i>MAD</i> is based on information mesh infrastructure and local auctions within the robot network for obtaining the optimal dispatch schedule for each robot. The simulations conducted confirm the advantages of our algorithms over other existing solutions in terms of average robot-event distance and lifetime.</p>	
Accepted by the Scientific Board on, ASB :		June 25, 2015	
Defended on, DE :			
Defended Board, DB :	President:	Ivan Mezei, Ph.D, assistant professor	Menthor's sign
	Member:	Rastislav Struharik, Ph.D, associate professor	
	Member:	Aleksandar Menićanin, Ph.D, research fellow	
	Member:	Branislav Borovac, Ph.D, full professor	
	Member:	Dejan Vukobratović, Ph.D, associate professor	
	Member, Mentor:	Ladislav Novak, Ph.D, full professor	

Zahvalnica

Tokom rada na disertaciji, imao sam čast i privilegiju da obavljam istraživanja pod mentorstvom jednog od vodećih svetskih stručnjaka iz oblasti bežičnih senzorskih mreža i teorijskih računarskih nauka, dr Ivana Stojmenovića. Nažalost, profesor Stojmenović odnedavno nije više među nama, ali je za sobom ostavio neizbrisiv trag idući stazom kojom je malo njih u stanju da korača. Profesore, hvala Vam za sve i vecna Vam slava.

Takođe, dugujem neizmernu zahvalnost profesoru dr Ladislavu Novaku, koji je nakon tragične pogibije profesora Stojmenovića preuzeo ulogu mog mentora i uz čiju sam podršku i razumevanje rad na disertaciji uspešno priveo kraju.

Posebnu zahvalnost želim da izrazim kolegama sa Katedre za elektroniku, a naročito članovima grupe za embeded sisteme i algoritme. Njihova pomoć i podrška dali su mi dodatni vetar u jedra u trenucima kada mi je to bilo najpotrebnije.

*Mojim najdražima Maji i Filipu,
hvala vam što postojite! :)*

Sadržaj

Zahvalnica	v
Spisak slika	x
Spisak tabela	xii
Spisak algoritama	xiv
Spisak skraćenica	xv
Sažetak	xvi
1 Uvod i postavka problema	1
1.1 Tehnologija bežičnih senzorskih mreža	1
1.2 Bežične mreže senzora i robota	4
1.2.1 Komunikacioni model	6
1.2.2 Centralizovana ili distribuirana koordinacija?	8
1.3 Formalna definicija problema	8
1.4 Algoritamska složenost problema dodele zadataka izvršiocima	9
1.4.1 Analiza složenosti problema m-dodela(m,n)	9
1.4.2 Analiza složenosti problema s-dodela(m,n)	10
1.4.3 Osvrt na ciljeve istraživanja	11
1.5 Pregled literature	12
1.5.1 Koordinacija u sistemima sa više robota	12
1.5.1.1 Primena Voronoi dijagrama u BSM	15
1.5.1.2 Tržišne metode koordinacije robota	17
1.5.2 Komunikacioni aspekti koordinacije	19
1.5.2.1 Lociranje izvršilaca	20
1.6 Postojeća rešenja	24
1.6.1 Uparivanje robota sa događajima	24
1.6.1.1 Uparivanje u bipartitnom grafu sa minimalnom dužinom ivica - Mađarski algoritam	24
1.6.1.2 Problem stabilnih brakova	26

1.6.1.3	Maksimalno uparivanje	27
1.6.2	Komunikacioni protokoli za otkrivanje izvršilaca	29
1.6.2.1	Prosleđivanje poruka u BSM sa garantovanom isporukom	29
1.6.2.2	Pravougaoni kvorum	31
1.6.2.3	iMesh struktura	32
1.6.3	Koordinacija među robotima u slučaju višestrukih događaja	33
1.6.3.1	Centralizovana koordinacija izvršilaca u slučaju višestrukih događaja zasnovana na klasterima	33
1.6.3.2	Pokrivanje grafa stablima sa korenjem u skupu R	36
1.6.3.3	Delimično distribuirani algoritmi koordinacije izvršilaca	37
1.6.3.4	Aukcijska agregacija u slučaju pojedinačnog događaja	39
1.6.4	Problem trgovačkog putnika	40
1.7	Doprinosi i pregled disertacije	42
2	Komunikacioni protokoli za otkrivanje izvršilaca	45
2.1	Karakteristike protokola za otkrivanje izvršilaca	46
2.1.1	Analiza nedostataka postojećih rešenja	48
2.2	Modifikovani iMesh protokol u gustim mrežama sa uniformnom raspodelom čvorova	53
2.2.1	Notacija	55
2.2.2	Formiranje modifikovane iMesh strukture	55
2.2.3	Ukrštena pretraga	59
2.2.4	Ažuriranje lokacije u okviru iMesh strukture	60
2.3	vCell protokol	62
2.3.1	Formiranje vCell informacione strukture	62
2.3.2	Ažuriranje lokacije u okviru vCell strukture	65
2.4	Procena komunikacionih troškova	67
2.4.1	Komunikacioni troškovi pravougaonog kvoruma	68
2.4.2	Komunikacioni troškovi iMesh protokola	68
2.4.3	Komunikacioni troškovi vCell protokola	69
3	Koordinacija među robotima u slučaju višestrukih događaja	70
3.1	Centralizovana rešenja problema m -dodela(m,n)	72
3.1.1	Centralizovano uparivanje zasnovano na udaljenostima (PDM algoritam)	73
3.1.2	Centralizovano uparivanje zasnovano na otpornostima (PRM algoritam)	75
3.2	Centralizovana rešenja problema s -dodela(m,n)	77
3.2.1	Centralizovana dodela sekvenci događaja izvršiocima (SQD i SRD algoritam)	77
3.2.2	Centralizovana dodela višestrukih izvršilaca događajima (GSQD algoritam)	79

3.2.3	Centralizovana dodela sekvenci događaja izvršiocima sa vremenskim ograničenjima (SQDT algoritam)	81
3.3	Distribuirana rešenja problema <i>s-dodela(m,n)</i>	83
3.3.1	Distribuirana dodela sekvenci događaja izvršiocima (MAD i MRD algoritam)	83
3.3.2	Distribuirana dodela višestrukih izvršilaca događajima (GAD algoritam)	87
4	Poređenje performansi algoritama sa postojećim rešenjima	88
4.1	Simulacije protokola za otkrivanje izvršilaca	89
4.1.1	Postavka simulacionog okruženja	89
4.1.2	Komunikacioni troškovi	92
4.1.3	Procenat uspeha u pronalaženju najbližeg izvršioca	94
4.1.4	Uticaj gustine mreže na performanse protokola za otkrivanje izvršilaca	95
4.1.5	Diskusija o životnom vremenu sistema	97
4.1.6	Zaključci u vezi sa protokolima za otkrivanje izvršilaca	98
4.2	Simulacije centralizovanih i distribuiranih algoritama za dodelu zadataka robotima	100
4.2.1	Centralizovano uparivanje robota s događajima	100
4.2.2	Centralizovana dodela sekvenci događaja izvršiocima	104
4.2.3	Centralizovana i distribuirana dodela višestrukih izvršilaca događajima	107
4.2.4	Distribuirana dodela sekvenci događaja izvršiocima	108
4.2.5	Zaključci u vezi sa algoritmima za dodelu zadataka izvršiocima	109
A	Rezultati simulacija protokola za otkrivanje izvršilaca	112
A.1	Komunikacioni troškovi faze formiranja informacionog direktorijuma	112
A.2	Komunikacioni troškovi faze otkrivanja izvršilaca	114
A.3	Komunikacioni troškovi faze održavanja informacionog direktorijuma	115
A.4	Procenat uspeha u pronalaženju najbližeg izvršioca	117
B	Rezultati simulacija algoritama za dodelu zadataka izvršiocima	121
B.1	Centralizovano uparivanje robota s događajima	121
B.2	Centralizovana dodela sekvenci događaja izvršiocima	123
B.3	Centralizovana i distribuirana dodela višestrukih izvršilaca događajima	124
B.4	Distribuirana dodela sekvenci događaja izvršiocima	124
	Literatura	126

Spisak slika

1.1	Tipična arhitektura senzorskog čvora	2
1.2	Hronologija razvoja bežičnih senzorskih čvorova (preuzeto iz [AV10])	3
1.3	Osnovne mrežne topologije (preuzeto iz [L ⁺ 04])	3
1.4	Bežična mreža senzora i robota	5
1.5	Jedinični Disk Graf	7
1.6	Uparivanje robota sa događajima ne mora predstavljati optimalno rešenje u pogledu minimizacije ukupnog pređenog puta	10
1.7	Uparivanje robota sa događajima pomoću SB algoritma	27
1.8	Lica u planarnom grafu duž čijih ivica se vrši rutiranje (preuzeto iz [FS06])	30
1.9	GFG rutiranje (preuzeto iz [FS06])	31
1.10	Formiranje kvoruma (preuzeto iz [SLJ08])	32
1.11	iMesh struktura sa 4 robota i senzorom koji vrši pretragu	33
1.12	Metode formiranja klastera (preuzeto iz [WPT10])	35
1.13	Primer izvršenja RRTC algoritma (preuzeto iz [EGK ⁺ 04])	37
1.14	Putanje robota pri izvršenju distribuirane verzije WPT algoritma	39
1.15	k-SAAP protokol (preuzeto iz [MLMS13])	40
1.16	2-aproksimativni TSP algoritam (preuzeto iz [CLR ⁺ 01])	41
2.1	BMSR sa pojedinačnim događajem i 4 robota (preuzeto iz [MMS10])	46
2.2	Primer Voronoi dijagrama sa 8 tačaka	47
2.3	Slučajevi kada iMesh nije u stanju da otkrije najbližeg izvršioca	49
2.4	iMesh struktura sa GFG rutiranjem (preuzeto iz [LSS07])	50
2.5	Ukrštanje putanja registracionih poruka (preuzeto iz [LSS07])	50
2.6	Slučajevi u kojima ukrštena pretraga ne otkriva najbližeg izvršioca (preuzeto iz [LSS07])	51
2.7	Ilustracija nedostataka algoritma iMeshA	52
2.8	Minimalna širina iMesh pojasa	54
2.9	iMesh struktura	55
2.10	Strategije rutiranja registracionih poruka	56
2.11	iMeshD ukrštena pretraga	60
2.12	Ažuriranje lokacije robota u okviru iMesh strukture	61
2.13	vCell struktura	64
2.14	iMeshD i vCell u mreži sa prazninom u obliku polumeseca	65
3.1	Proces razmene partnera u okviru PDM algoritma	74
3.2	Primeri uparivanja koja predstavljaju lokalne optimume	75

3.3	Primer dodele sekvenci događaja izvršiocima	78
3.4	Primer GSQD	80
3.5	Implementacija aukcione faze MAD algoritma kao distribuiranog konačnog automata	86
3.6	Distribuirana dodela sekvenci događaja izvršiocima (MAD)	86
4.1	Mrežne topologije	90
4.2	Komunikacioni troškovi tokom faze konstrukcije direktorijuma	92
4.3	Komunikacioni troškovi tokom faze otkrivanja i obaveštavanja izvršilaca	93
4.4	Komunikacioni troškovi tokom faze održavanja direktorijuma	93
4.5	Procenat uspeha u pronalaženju najbližeg izvršioca za konstantan domet emitovanja $R = 75$	95
4.6	Stepen čvorova u zavisnosti od dometa emitovanja	95
4.7	Procenat uspeha u pronalaženju najbližeg izvršioca za konstantan broj robota $n = 5$	96
4.8	Mogućnost poboljšanja nalaženja najbližeg izvršioca u mreži sa uniformnom raspodelom čvorova	97
4.9	Uparivanje robota sa događajima ($n = m = 20$)	101
4.10	Rezultati simulacija za centralizovane algoritme uparivanja	102
4.11	Životno vreme sistema prilikom primene različitih algoritama za uparivanje robota s događajima	103
4.12	Životno vreme sistema prilikom primene različitih algoritama za sekvencijalnu dodelu događaja robotima	104
4.13	Primer dodele sekvenci događaja izvršiocima ($n = 10, m = 100$)	105
4.14	Prosečno trajanje runde	106
4.15	Procenat broja poseta koje se prenose u narednu rundu	107
4.16	Problematičan slučaj distribuirane koordinacije	110

Spisak tabela

1.1	Aspekti problema koordinacije robota	5
2.1	Procena komunikacionih troškova tokom različitih faza protokola za otkrivanje izvršilaca	68
A.1	Troškovi formiranja direktorijuma - uniformna raspodela čvorova	112
A.2	Troškovi formiranja direktorijuma - praznina u obliku kvadrata	113
A.3	Troškovi formiranja direktorijuma - praznina u obliku kruga	113
A.4	Troškovi formiranja direktorijuma - praznina u obliku polumeseca	113
A.5	Troškovi pretrage i obaveštavanja izvršilaca - uniformna raspodela čvorova	114
A.6	Troškovi pretrage i obaveštavanja izvršilaca - praznina u obliku kvadrata	114
A.7	Troškovi pretrage i obaveštavanja izvršilaca - praznina u obliku kruga .	114
A.8	Troškovi pretrage i obaveštavanja izvršilaca - praznina u obliku polumeseca	115
A.9	Troškovi održavanja direktorijuma - uniformna raspodela čvorova	115
A.10	Troškovi održavanja direktorijuma - praznina u obliku kvadrata	116
A.11	Troškovi održavanja direktorijuma - praznina u obliku kruga	116
A.12	Troškovi održavanja direktorijuma - praznina u obliku polumeseca	116
A.13	Procenat uspeha u zavisnosti od broja robota - uniformna raspodela čvorova	117
A.14	Procenat uspeha u zavisnosti od broja robota - praznina u obliku kvadrata	117
A.15	Procenat uspeha u zavisnosti od broja robota - praznina u obliku kruga .	118
A.16	Procenat uspeha u zavisnosti od broja robota - praznina u obliku polumeseca	118
A.17	Procenat uspeha u zavisnosti od dometa - Uniformna raspodela čvorova	118
A.18	Procenat uspeha u zavisnosti od dometa - praznina u obliku kvadrata . .	119
A.19	Procenat uspeha u zavisnosti od dometa - praznina u obliku kruga	119
A.20	Procenat uspeha u zavisnosti od dometa - praznina u obliku polumeseca	119
A.21	Procenat uspeha u pronalaženju bilo kog izvršioca u zavisnosti od dometa - uniformna raspodela čvorova	120
B.1	Prosečno rastojanje robot-događaj	121
B.2	Premašaj rastojanja u odnosu na HU algoritam	122
B.3	Standardna devijacija rastojanja robot-događaj	122
B.4	Životno vreme sistema (u rundama)	122

B.5	Životno vreme sistema prilikom centralizovane dodele sekvenci događaja izvršiocima	123
B.6	Prosečno trajanje runde	123
B.7	Procenat broja poseta koje se prenose u narednu rundu	124
B.8	Distribuirani algoritmi za dodelu zadataka izvršiocima	125

Spisak algoritama

1	Mađarski algoritam [Kuh55]	25
2	Algoritam SB [AGB10][GS62]	26
3	Algoritam MM(β) [WPT10]	28
4	Algoritam RRTC [EGK ⁺ 04]	36
5	Formiranje iMesh strukture	58
6	iMesh ažuriranje lokacije	60
7	Formiranje vCell strukture	63
8	vCell ažuriranje lokacije	66
9	Centralizovano uparivanje zasnovano na udaljenostima (<i>PDM</i>)	74
10	Centralizovana dodela sekvenci događaja izvršiocima (<i>SQD</i>)	78
11	Generalizovana dodela sekvenci događaja izvršiocima (<i>GSQD</i>)	80
12	Centralizovana dodela sekvenci događaja izvršiocima sa vremenskim oganičenjima (<i>SQDT</i>)	82
13	Pojedinačna iteracija koju izvršava robot R_i u okviru algoritma <i>MAD</i>	85
14	Simulacije protokola za otkrivanje izvršilaca	91

Spisak skraćenica

BMSR	- Bežična Mreža Senzora i Robota
BSM	- Bežična Senzorska Mreža
RF	- Radio Frekvencija
A/D	- Analogno/Digitalni
JDG	- Jedinični Disk Graf
SB	- algoritam Stabilnih Brakova
MSR	- Minimalno Stablo Razapinjanja
WSRN ili WSAN	- engl. W ireless S ensor and R obot (A ctor) N etwork
WSN	- engl. W ireless S ensor N etwork
GPS	- engl. G lobal P osition S ystem
MRTA	- engl. M ulti- R obot T ask A llocation
MM	- engl. M aximum M atching
GFG	- engl. G reedy- F ace- G reedy
RRTC	- engl. R - R ooted T ree C over
TSP	- engl. T raveling S alesman P roblem
k-SAAP	- engl. k -hop S imple A uction A ggregation P rotocol
MFR	- engl. M ost F orward R outing
LDR	- engl. L east D eviation from stright line R outing
PDM	- engl. centralized P airwise D istance-based M atching
PRM	- engl. centralized P airwise R esistance-based M atching
SQD	- engl. S e Q uence D ispatch
SRD	- engl. S equence R esistance-based D ispatch
GSQD	- engl. G eneralized S e Q uence D ispatch
SQDT	- engl. S e Q uence D ispatch with T iming constraints
MAD	- engl. i Mesh and A uction-based D ispatch
MRD	- engl. i Mesh and auction R esistance based D istributed dispatch
MRD	- engl. i Mesh and auction R esistance-based D istributed dispatch

Sažetak

Bežičnu mrežu senzora i robota tipično sačinjava veliki broj malih, jeftinih, baterijski napajanih statičnih senzorskih modula i manji broj mobilnih robota sa značajnim resursima. Senzorski moduli nadziru fizičke veličine od značaja u svom okruženju. Roboti su opremljeni uređajima koji omogućavaju automatizovano obavljanje zadataka bez upravljanja od strane čoveka. Kada dođe do kritične promene parametara, što je situacija koja se naziva događajem, potrebno je preduzeti akciju za koju će biti zadužen neki od robota. U ovom kontekstu, roboti imaju ulogu izvršilaca zadataka koji im se dodeljuju primenom odgovarajućeg algoritma. Nakon detekcije događaja od strane statičkih senzorskih čvorova i prosleđivanja informacija o događajima robotima, potrebno je dovesti robote u koordinaciju na efikasan način. Dodela zadataka vrši se u skladu sa prirodom različitih scenarija koji se mogu javiti u praksi. Tema istraživanja sprovedenog tokom rada na ovoj disertaciji bavi se slučajem kada se konkurentno javlja više događaja kojima je potrebno dodeliti izvršioce. U pogledu energetske efikasnosti, u ovakvim sistemima javljaju se dva ključna problema: minimizacija ukupne dužine kretanja robota i optimizacija komunikacije u mreži.

Kada su u pitanju komunikacioni protokoli za otkrivanje izvršilaca od strane senzorskih čvorova koji detektuju događaje, u ovoj disertaciji između ostalog predstavljena su poboljšanja postojećeg *iMesh* protokola i uveden je novi *vCell* protokol zasnovan na lokalizovanom formiranju ćelija Voronoi dijagrama. Takođe, upoređene su performanse novog protokola sa postojećim (pravougaoni kvorum i *iMesh*) u gustim mrežama, retkim mrežama i mrežama sa rupama u topologiji. Uz to, uvedeni su algoritmi za ažuriranje lokacije kojima mreža reaguje na kretanje robota. Rezultati simulacija pokazuju da *vCell* postiže efikasnost blizu 100% u nalaženju najbližeg robota u gustim mrežama. U retkim mrežama, efikasnost mu je do 40% bolja u odnosu na ostala rešenja.

Kao glavni rezultat u ovoj disertaciji prikazani su novi algoritmi za dodelu robota kao izvršilaca zadataka događajima, čime su prevaziđni nedostaci više do sada poznatih rešenja ovog problema. Za zadati skup događaja i skup robota, svakom događaju dodeljen je po jedan robot koji je zadužen za obilazak lokacije događaja. Tokom pojedinačnih rundi, robotima je dozvoljen obilazak jednog događaja kada se vrši uparivanje, ili više događaja, kada se vrši sekvencijalna dodela. U distribuiranom slučaju, statički senzorski uređaji detektuju događaje i prijavljuju ih obližnjim robotima. Algoritam *PDM* koji se odnosi na unapređeno uparivanje sa mogućnošću razmene partnera, eliminiše dugačke ivice koje se mogu javiti prilikom uparivanja. Algoritam *SQD* za sekvencijalnu dodelu događaja robotima iterativno pronalazi par robot-događaj sa najmanjim međusobnim rastojanjem, uvrštava izabrani događaj u listu za obilazak izabranog robota i ažurira poziciju robota. Takođe su predložene generalizacije koje omogućavaju da događaji budu posećeni od strane više robota i koje uzimaju u obzir vremenska ograničenja. Distribuirani algoritam *MAD*, koji je zasnovan na *iMesh* informacionoj strukturi i lokalnim aukcijama u robotskoj mreži, vrši dodelu robota događajima na lokalizovan i energetske efikasan način. Rezultati simulacija potvrđuju prednosti predloženih algoritama u odnosu na postojeća rešenja, kako u pogledu skraćivanja dužina putanja robota, tako i u produženju životnog vremena sistema.

Poglavlje 1

Uvod i postavka problema

Brz i dinamičan razvoj novih tehnologija u oblastima robotike, bežičnih komunikacija i mikrokontrolerskih *embedded* sistema doveo je poslednjih nekoliko godina do pojave relativno nove istraživačke discipline. U pitanju su bežične mreže senzora i robota (*BMSR*, engl. *WSRN*), u kojima se prepliću tradicionalne i relativno dobro istražene oblasti sa mnogobrojnim novim istraživačkim izazovima i otvorenim pitanjima i problemima koji ili još uvek nisu rešeni, ili nisu rešeni na adekvatan način. U uvodnom poglavlju su navedeni osnovni pojmovi vezani za oblast *BMSR*, nakon čega je problem koji je predmet istraživanja sprovedenog u okviru rada na ovoj disertaciji formalno definisan.

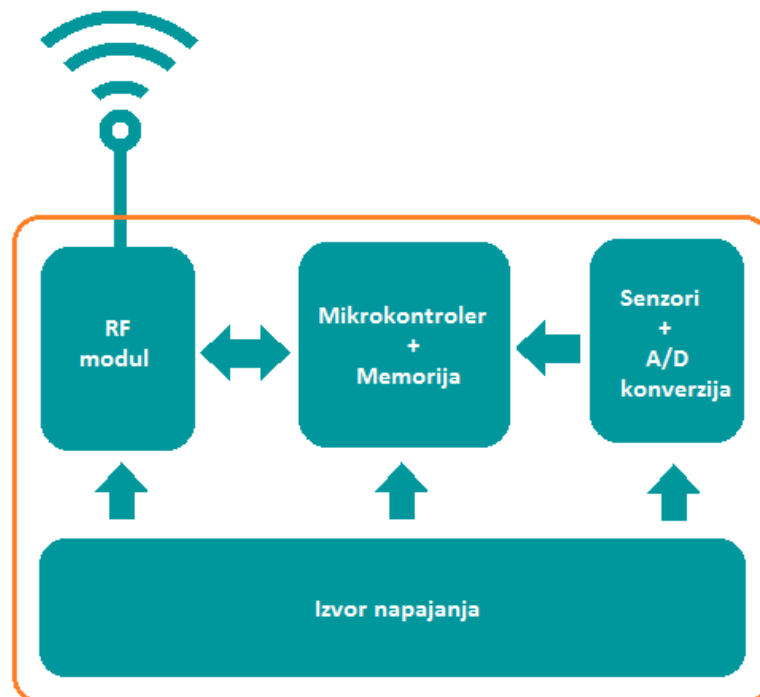
1.1 Tehnologija bežičnih senzorskih mreža

Tokom proteklih dvadesetak godina, bežične senzorske mreže (*BSM*, engl. *WSN* = *Wireless Sensor Network*) su bile i još uvek su predmet intenzivnog izučavanja, što je praćeno objavljivanjem brojnih naučnih publikacija među kojima su [DP10][Sto05][SMZ07] [AV10][L⁺04][RSZ04]. Generalna ideja je da se na određenom prostoru rasporede autonomni senzorski uređaji koji se nazivaju još i čvorovima¹. Njihova uloga je da nadziru fizičke parametre okruženja kao što su temperatura, pritisak, osvetljenost, koncentracija hemikalija i sl. Rezultati očitavanja senzora prenose se kroz mrežu ka sabirnom mestu (baznoj stanici, kolektoru, engl. *sink*) korišćenjem odgovarajućih komunikacionih protokola [AY05][YHE02][VDL03]. U cilju uštede ograničenih energetske resursa kojima ovakvi uređaji raspolažu, podaci

¹Senzorski uređaji se nazivaju čvorovima (engl. *nodes*) stoga što je uobičajeno da se topologija mreže opisuje pojmovima preuzetim iz teorije grafova.

se prema baznoj stanici najčešće prenose u više hopova² (engl. *hops*) [BMJ⁺98][DPZ04]. Čvorovi posrednici koji vrše retransmisiju mogu uz to da obavljaju i fuziju podataka [CSZ⁺09][GP08], u cilju dodatnog smanjenja broja poruka koje se prenose kroz mrežu.

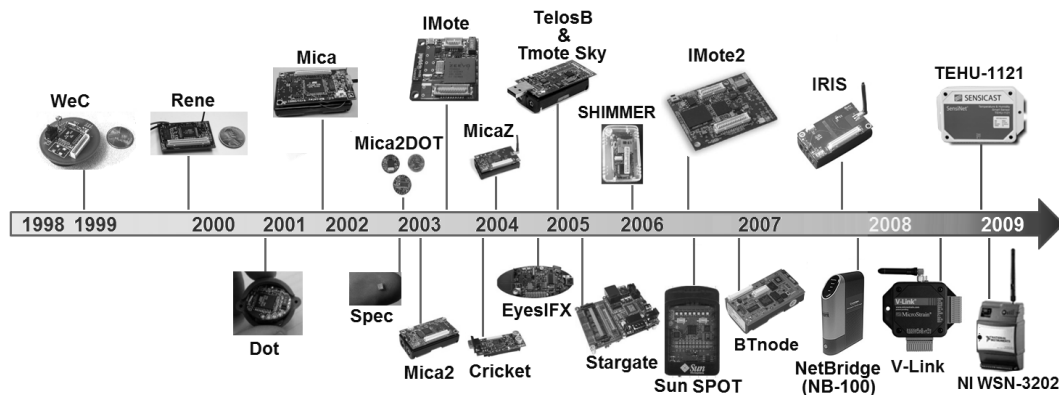
Senzorski čvorovi, kao osnovni elementi *BSM* [Hil03], od hardverskih komponenti obično sadrže izvor napajanja, mikrokontroler sa internim i/ili eksternim memorijskim modulima, radio primopredajnik (*RF* modul) i senzore zajedno sa kolima za prilagođenje signala i *A/D* konverziju (slika 1.1).



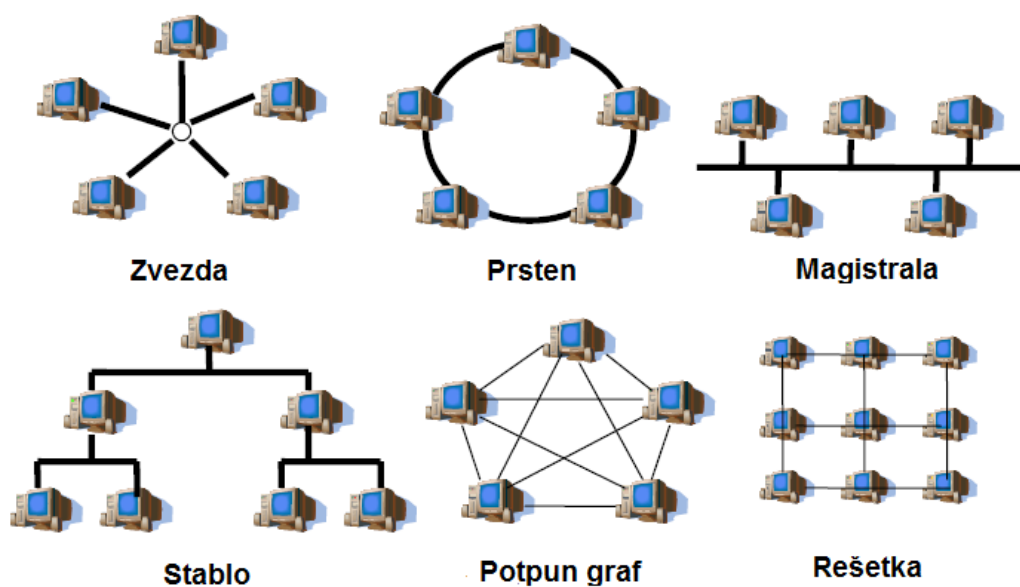
Slika 1.1: *Tipična arhitektura senzorskog čvora*

Ovakva arhitektura omogućava obavljanje osnovnih zadataka u koje spadaju akvizicija senzorskih podataka, njihovo čuvanje i obrada, kao i komunikacija sa drugim čvorovima. Veličina i cena senzorskih čvorova varira u zavisnosti od njihove namene i kompleksnosti. Na slici 1.2 prikazana je hronologija razvoja najčešće korišćenih senzorskih platformi [AV10][BCDV09]. Razvoj hardvera je bio na adekvatan način praćen i razvojem specijalizovanih operativnih sestama koji podržavaju ovakve platforme, među kojima su najčešće korišćeni TinyOS [LMP⁺05], LiteOS [CASH08], i Contiki [DGV04].

²Pojam "hop" odnosi se na komunikaciju u bežičnoj mreži u kojoj predajni i prijemni čvor ne komuniciraju direktno, nego preko jednog ili više posrednika. U ovom slučaju, broj hopova predstavlja ukupan broj čvorova u nizu, koji vrše (re)transmisiju prilikom prenosa poruke između predajnog i prijemnog čvora.



Slika 1.2: Hronologija razvoja bežičnih senzorskih čvorova (preuzeto iz [AV10])



Slika 1.3: Osnovne mrežne topologije (preuzeto iz [L⁺04])

U zavisnosti od broja, namene i prostornog rasporeda senzorskih čvorova, komunikacioni graf senzorske mreže može imati različite topologije, kao što su stablo, zvezda, prsten, magistrala, potpun mrežni graf ili rešetkasta topologija (slika 1.3) [SKK13][SX07][Mam12][L⁺04]. Kada je u pitanju način bežičnog prenosa podataka u BSM, u praksi se najčešće koriste sledeće standardizovane mrežne tehnologije [SMZ07]:

- IEEE 802.15.1 (PAN/Bluetooth) [MB01]
- IEEE 802.11a/b/g/n (Wi-Fi) [OP05]
- IEEE 802.15.4 (ZigBee) [BPC⁺07]
- IEEE 802.16 (WiMax) [GWAC05]

- IEEE 802.15.3 ultrawideband (UWB) [YG04]
- IEEE 1451.5 (Wireless Sensor Working Group) [SSNJ06]
- Mobilni IP [Per98]

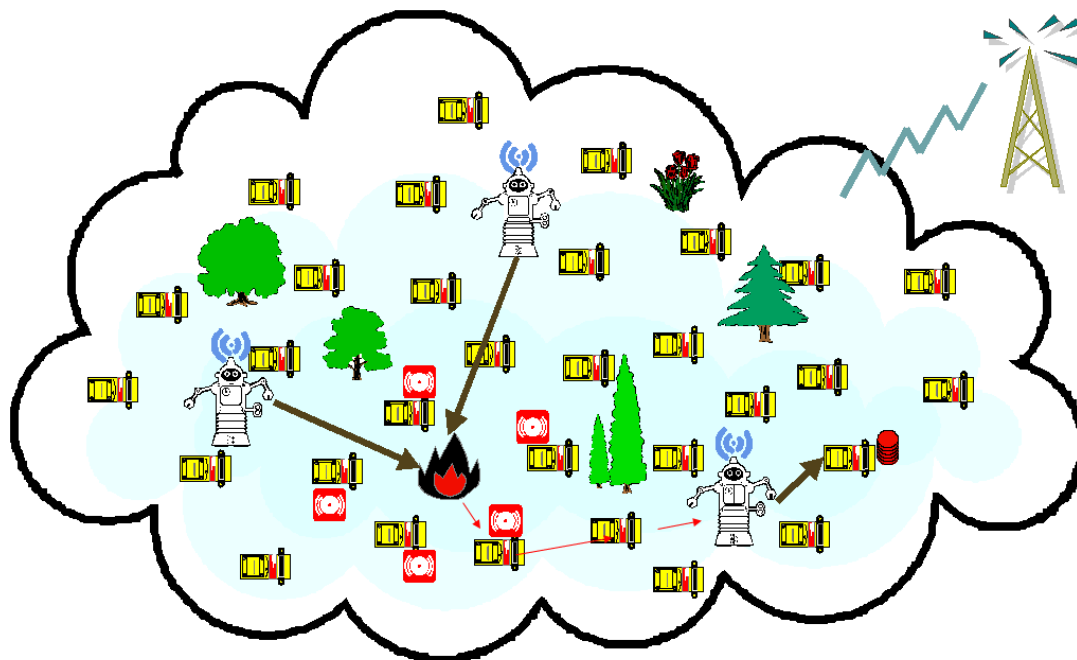
Istorijski gledano, razvoj bežičnih senzorskih mreža je u početku bio primarno motivisan primenama u vojne svrhe. Međutim, u današnje vreme postoji širok dijapazon primena ovakvih mreža, a ovde će biti pomenute samo neke od mogućih situacija u kojima se upotrebljavaju [AV10]:

- Vojne aplikacije - nadzor bojnog polja, otkrivanje snajpera, detekcija pokreta neprijateljskih trupa
- Nadzor parametara životne sredine - praćenje koncentracije zagađivača, detekcija požara, nadzor vulkanske aktivnosti, meteorološka merenja, praćenje kretanja životinja
- Medicina - daljinsko praćenje stanja pacijenta
- Primene u domaćinstvu - "pametne kuće"
- Nadzor u industriji - praćenje stanja mašina i opreme, praćenje koncentracije otpadnih hemikalija, merenje temperature/pritiska i drugih parametara tehnoloških procesa

1.2 Bežične mreže senzora i robota

Oblast bežičnih mreža senzora i robota (*BMSR*, engl. *WSRN = Wireless Sensor and Robot Network*³) [NS10][VDMC10][Xia08][AK04] predstavlja disciplinu koja povezuje problematiku iz oblasti *BSM* i multirobotskih sistema [GM04]. Tipičan *BMSR* sistem sačinjava veliki broj malih, jeftinih, baterijski napajanih statičnih senzorskih čvorova i manji broj mobilnih robota. Roboti su opremljeni uređajima koji omogućavaju automatizovano obavljanje zadataka bez upravljanja od strane čoveka, što je naročito od značaja u okruženjima koja mogu biti slabo naseljena, nepristupačna, ili opasna po život. Kada dođe do kritične promene parametara okruženja, što je situacija koja se naziva događajem, potrebna je dalja obrada koju obavlja neki od robota. Roboti su opremljeni sofisticiranijom (i skupljom) opremom koja omogućava detaljniji uvid u prirodu događaja i omogućava pravovremenu reakciju na događaj. Na primer, u slučaju sistema za detekciju i sprečavanje šumskih požara, roboti mogu da budu opremljeni vatrogasnim uređajima (slika 1.4). U ovom kontekstu, roboti imaju ulogu izvršilaca zadataka koji im se dodeljuju primenom odgovarajućeg algoritma.

³U literaturi se ovakve mreže nazivaju još i bežičnim mrežama senzora i aktuatora (engl. *WSAN = Wireless Sensor and Act(uat)or Network*).

Slika 1.4: *Bežična mreža senzora i robota*

Nakon detekcije događaja od strane statičkih senzorskih čvorova i prosleđivanja informacija o događajima robotima, potrebno je izvršiti koordinaciju robota na efikasan način. Koordinacija robota [SYTX06][FDS⁺02][KSV05] podrazumeva obavljanje dve akcije koje slede jedna za drugom. Prvo se odvija dodela (alokacija) zadataka robotima, nakon čega roboti prelaze na obavljanje zadataka koji su im dodeljeni. Dodela zadataka se vrši u skladu sa prirodom različitih scenarija [MLM14] koji se mogu javiti u praksi i koji su ukratko sumirani u tabeli 1.1.

Tipovi događaja	jednostruki višestruki sekvencijalni višestruki "grupa po grupa" višestruki paralelni
Dodela zadataka	1 zadatak - 1 izvršilac višestruki zadaci - 1 izvršilac 1 zadatak - višestruki izvršioci višestruki zadaci - višestruki izvršioci
Tip rešenja	centralizovano lokalizovano distribuirano tržišno

Tabela 1.1: *Aspekti problema koordinacije robota*

U jednom od ranijih istraživanja, proučavana je koordinacija robota u slučaju pojave pojedinačnog događaja, kojem se dodeljuje jedan izvršilac, po kriterijumu najmanje udaljenosti između lokacija događaja i izvršioca. Kao jedan od najvažnijih rezultata, u

radu [MLMS13], prikazan je k -SAAP protokol koji predstavlja rešenje koordinacije zasnovano na lokalizovanim aukcijama sa agregacijom podataka. U okviru ove disertacije u značajnoj meri se proširuju i uopštavaju ti rezultati, baveći se slučajem kada se konkurentno javlja više događaja kojima je potrebno dodeliti izvršioce.

U pogledu energetske efikasnosti [JSAC01], u takvim sistemima sa konkurentnom pojavom više događaja javljaju se dva ključna problema:

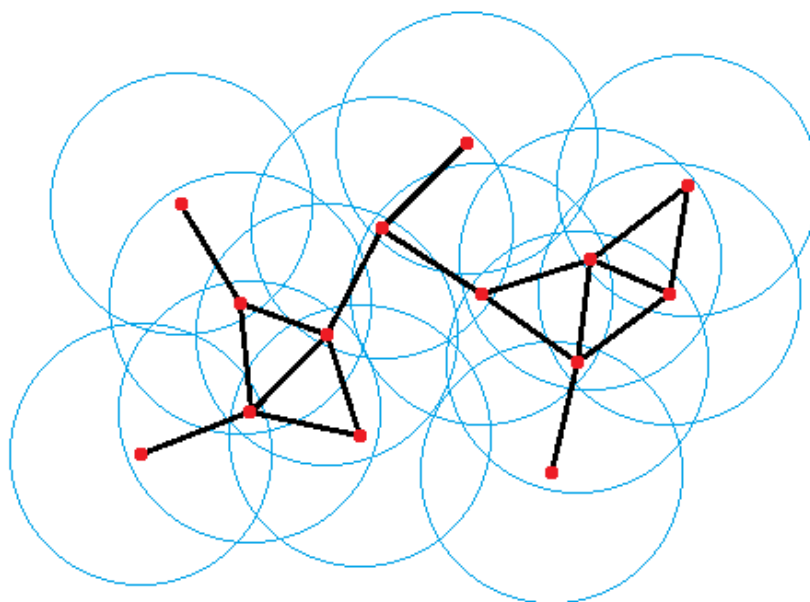
- Minimizacija ukupne dužine kretanja robota. Od zadataka koje robot obavlja (kretanje, komunikacija, obavljanje merenja, obrada podataka), kretanje je u energetsom smislu najzahtevnije. Stoga je od ključnog značaja za produženje vremena rada sistema optimizovati njihovo kretanje, kako bi ukupna dužina robotskih putanja bila minimalna, bez ugrožavanja funkcionalnosti sistema.
- Optimizacija komunikacije u mreži, odnosno smanjenje broja poruka koje razmenjuju senzorski čvorovi. Analogno kretanju kod robota, najveći deo energije napajanja statičnih senzorskih uređaja odlazi upravo na uspostavljanje bežične komunikacije [WHY06], pa je od interesa da se komunikacija obavlja na što efikasniji način.

1.2.1 Komunikacioni model

Rani radovi iz oblasti koordinacije robota zanemarivali su komunikacione aspekte problema, polazeći od pretpostavke da je komunikacioni graf kompletan, odnosno da je moguće ostvariti direktnu komunikaciju između bilo koja dva čvora, ili da su komunikacioni troškovi zanemarljivi, pa se informacije prenose tako što se mreža svaki put "preplavljuje" (engl. *Flooding*), pri čemu svaki od čvorova u idealnom slučaju vrši tačno jednu retransmisiju⁴. Obe pretpostavke postaju izrazito problematične u slučaju mreža koje sadrže veliki broj (stotine, ili čak hiljade) čvorova, pošto se javlja problem sa skalabilnošću [AE⁺09]. Čvorovi u okviru ovakve mreže mogu biti raspoređeni po relativno velikoj površini, sa međusobnim rastojanjima koja uveliko prevazilaze domet emitovanja bežičnog signala. Samim tim, javlja se potreba za realizacijom efikasnijih mehanizama komunikacije. Novija istraživanja na ovu temu uzimaju u obzir tri različite komunikacione paradigme [AK04]:

- Komunikacija senzor-senzor
- Komunikacija senzor-robot
- Komunikacija robot-robot

⁴U ovom slučaju su ukupni komunikacioni troškovi $O(s)$, gde je s broj čvorova u mreži.



Slika 1.5: Jedinični Disk Graf

Kao model komunikacije, u ovom i u većini drugih radova koristi se Jedinični Disk Graf (JDG, engl. *UDG = Unit Disk Graph*) [CCJ90]. U pitanju je jedan od najjednostavnijih modela, u kojem se domet bežičnog emitovanja uzima kao poluprečnik jediničnog diska sa centrom u mrežnom čvoru. Komunikaciju sa čvorom mogu da ostvare svi i samo oni čvorovi koji su locirani unutar njegovog jediničnog diska (slika 1.5). U terminologiji *BSM*, uobičajeno je da se takvi čvorovi nazivaju 1-hop susedima datog čvora.

U disertaciji je usvojena dodatna pretpostavka, koja je bitna u komunikacionim protokolima koji koriste geografsko prosleđivanje poruka (rutiranje) [YGE01][MWH01]. Pretpostavlja se da svi mrežni entiteti (statički senzorski čvorovi i roboti) u svakom trenutku raspolažu informacijom o sopstvenoj lokaciji. To je moguće postići ako su svi čvorovi opremljeni sistemom za geografsku lokalizaciju kao što je *GPS*⁵, *GLONASS*⁶ ili *Galileo*⁷, ili ako su statični čvorovi postavljeni od strane agenta koji je opremljen takvim uređajem [YA08]. Takođe, pretpostavlja se da pri inicijalizaciji mreže svaki od čvorova emituje pozdravnu poruku, kojom obaveštava neposredne susede o svom prisustvu i poziciji. Nakon ovog koraka, čvorovi raspolažu lokalizovanim znanjem o svom 1-hop okruženju.

Konačno, neophodno je da svi čvorovi u mreži budu vremenski sinhronizovani [SBK05]. Mehanizam putem kog se postiže sinhronizacija može biti neki od specijalizovanih protokola kao što je *FTSP* (engl. *Flooding Time Synchronization Protocol*), koji je opisan u radu [MKSL04].

⁵<http://www.gps.gov/>

⁶<http://glonass-iac.ru/en/>

⁷http://www.esa.int/Our_Activities/Navigation/The_future_-_Galileo/What_is_Galileo

1.2.2 Centralizovana ili distribuirana koordinacija?

Većina postojećih rešenja problema koordinacije robota koja stavljaju u drugi plan komunikacione troškove su po svojoj prirodi centralizovani algoritmi. Centralizovano rešenje podrazumeva da jedan od čvorova u mreži (bilo da je u pitanju statički senzorski čvor ili robot), poseduje sve relevantne informacije i donosi odluke o raspodeli zadataka. Iako je teorijski u ovim uslovima moguće pronaći optimalno rešenje⁸, centralizovani pristup sam po sebi može da nametne visoke komunikacione troškove, kao i da zahteva značajne resurse u pogledu kompleksnosti izračunavanja i da proizvede neprihvatljivo velika kašnjenja, što u značajnoj meri ograničava upotrebnost ovakvih algoritama u praksi. U tu svrhu, težnja je da se donošenje odluka odvija na distribuiran način, korišćenjem lokalne komunikacije u okolini mesta događaja. Osnovne karakteristike ovakvih rešenja su brzo vreme reagovanja, niži troškovi po pitanju komunikacionih i računarskih resursa u odnosu na odgovarajuća centralizovana rešenja, kao i veća robusnost, skalabilnost i tolerancija na kvarove. Sa druge strane, rešenja dobijena na ovaj način su po svojoj prirodi uvek sub-optimalna, budući da se baziraju na ograničenoj količini informacija, odnosno na lokalizovanom znanju ograničenom na nekoliko hopova u robotskoj mreži.

I pored navedenih nedostataka, centralizovani algoritmi mogu da predstavljaju korisnu referencu za ocenu efikasnosti distribuiranih rešenja, koja uprkos tome što su sub-optimalna, po efikasnosti mogu da budu uporediva sa odgovarajućim centralizovanim rešenjima. Takođe, logika na kojoj su zasnovana određena centralizovana rešenja može predstavljati osnovu za razvoj odgovarajućih distribuiranih algoritama.

1.3 Formalna definicija problema

Neka je $E = \{E_1, E_2, \dots, E_m\}$ skup mesta događaja i neka je $R = \{R_1, R_2, \dots, R_n\}$ skup robota. Cilj je svakom robotu R_i pridružiti listu DS_i koja predstavlja sekvencu lokacija događaja koje dati robot treba da obiđe, tako da se njihova unija poklapa sa skupom svih događaja E . Lokacije su zadate tačkama u Euklidskoj ravni. Roboti su vremenski sinhronizovani i opslužuju događaje u rundama jednakog trajanja. Pretpostavka je da je priroda događaja takva da je za svaki događaj tokom jedne runde potrebno i dovoljno da bude posećen od strane jednog robota. Svaka runda se sastoji iz dve faze:

- Statička (komunikaciona) faza, tokom koje se roboti ne kreću. U ovoj fazi prikupljaju se informacije o broju i lokacijama događaja, nakon čega se izvršava algoritam kojim se formiraju liste obilazaka DS_i .

⁸U praksi, složenost algoritma koji dovodi do optimalnog rešenja može da obesmisli njegovu primenu, te se napor usmerava ka nalaženju dovoljno kvalitetnog suboptimalnog rešenja.

- Dinamička faza, u kojoj roboti redom obilaze događaje koji su im dodeljeni. Lokacije na kojima su se roboti zatekli po okončanju runde predstavljaju početne lokacije na kojima ulaze u sledeću rundu.

U slučaju kada je robotima dozvoljeno da u toku runde obiđu više događaja, problem se naziva *s-dodela*(m, n)⁹. Ovakav scenario je neizbežan u slučaju da je broj događaja veći od broja robota ($m > n$).

Od značaja je i varijanta istog problema sa restrikcijom da je svakom robotu dozvoljeno da obiđe najviše jedan događaj, nazvana *m-dodela*(m, n)¹⁰. To odgovara slučaju kada je broj događaja manji ili jednak broju robota ($m \leq n$).

U oba slučaja, radi se o optimizacionim problemima u kojima je potrebno minimizovati ukupnu dužinu putanja svih robota tokom svake pojedinačne runde. Budući da je energija koju roboti troše na kretanje linearno proporcionalna pređenom putu, kriterijum optimizacije postavljen je u cilju maksimizacije ukupnog životnog vremena sistema, odnosno broja rundi koje roboti mogu da odrade pre nego što istroše energiju.

1.4 Algoritamska složenost problema dodele zadataka izvršiocima

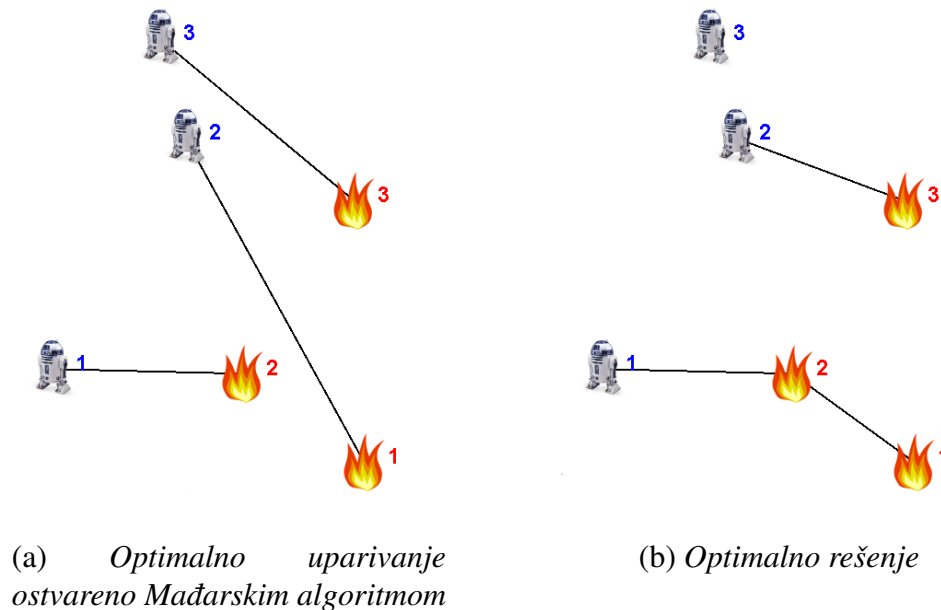
1.4.1 Analiza složenosti problema m-dodela(m,n)

Skup robota R i skup događaja E predstavljaju čvorove u kompletnom težinskom bipartitnom grafu [BM76]. Između svaka dva čvora R_i i E_j ($1 \leq i \leq n, 1 \leq j \leq m$) postoji ivica, kojoj je pridružena težina koja je jednaka njihovom međusobnom rastojanju $|R_i E_j|$. Ukoliko je $m = n$, problem se može svesti na uparivanje robota sa događajima (engl. *Matching problem*), uz zahtev da ukupna težina ivica bipartitnog grafa pridruženih izabranim parovima bude minimalna (engl. *Assignment problem*). Formalno, ako su zadata dva skupa R i E jednake kardinalnosti ($|R| = |E| = n$), zajedno sa težinskom funkcijom $w : R \times E \rightarrow \mathbb{R}$, potrebno je pronaći bijektivno preslikavanje $f : R \rightarrow E$, koje minimizuje funkciju cene:

$$C = \sum_{i=1}^n w(R_i, f(R_i))$$

⁹Naziv "s-dodela" upućuje na to da se robotima dodeljuju sekvence događaja.

¹⁰Naziv "m-dodela" potiče od toga što se u ovom slučaju vrši uparivanje robota s događajima (engl. *matching*).



Slika 1.6: Uparivanje robota sa događajima ne mora predstavljati optimalno rešenje u pogledu minimizacije ukupnog pređenog puta

Za problem uparivanja u bipartitnom grafu za koje je ukupna težina ivica minimalna postoji algoritam koji dovodi do optimalnog rešenja u polinomijalnom vremenu $O(n^3)$ (Mađarski algoritam, engl. *Hungarian algorithm* [Kuh55]). Međutim, to rešenje i dalje ne mora biti optimalno po pitanju minimizacije ukupnog puta koji roboti prelaze. Suštinski nedostatak je u tome da je svođenjem problema m -dodela(m,n) na *Assignment problem* uvedena pretpostavka da će svakom događaju biti dodeljen različit robot. U situaciji kao na slici 1.6, prikazan je slučaj u kojem je ukupan pređeni put minimalan ukoliko robot R_1 obiđe dva događaja (E_2 i E_1), a nasuprot tome robot R_3 ostaje na svojoj poziciji. Ovo predstavlja jednu instancu NP-teškog optimizacionog problema s -dodela(m,n) koji je opisan u sledećem odeljku i koji omogućava postizanje optimalnog rešenja.

1.4.2 Analiza složenosti problema s -dodela(m,n)

Problem s -dodela(m,n), za razliku od problema m -dodela(m,n), polazi od osnovne pretpostavke da svakom robotu može biti dodeljen proizvoljan broj događaja za obilazak tokom pojedinačne runde. Formalno, ako je $DS_i = \{E_{i1}, E_{i2}, \dots, E_{ik_i}\}$ lista događaja dodeljenih robotu R_i , događaj E_{ij} je j -ti po redosledu obilaska, a k_i je njihov ukupan broj, problem se svodi na minimizaciju funkcije:

$$C = \sum_{i=1}^n (|R_i E_{i1}| + \sum_{j=1}^{k_i-1} |E_{ij} E_{i,j+1}|)$$

Problem s -dodela(m, n) kao specijalni slučaj sadrži Problem trgovačkog putnika (engl. $TSP = Traveling Salesman Problem$) za $n = 1$. Pošto je TSP NP-težak optimizacioni problem, to je s -dodela(m, n) takođe NP-težak problem.

Opisani problem s -dodela(m, n) uključuje slučaj kada je broj događaja veći od broja robota, tj. $m > n$, ali nije ograničen samo na njega. Za ovaj slučaj u radu [WPT10] predložene su heuristike bazirane na podeli događaja u n grupa (klastera), nakon čega se roboti uparuju sa klasterima. Redosled obilaska mesta događaja u okviru klastera dobija se primenom TSP heuristike.

1.4.3 Osvrt na ciljeve istraživanja

Potrebno je formirati liste događaja koje će obići svaki od robota za zadatih n robota i m mesta događaja. U slučaju kada je $m \leq n$, moguće je izvršiti uparivanje robota sa događajima, pri čemu se svakom robotu dodeljuje najviše po jedan događaj za obilazak. Postoji polinomijalni algoritam koji ovakvo uparivanje vrši na optimalan način (Mađarski algoritam), ali on na žalost ne garantuje da je rešenje optimalno po pitanju minimizacije ukupnog pređenog puta svih robota. Uz pretpostavku da svaki robot sme da obiđe proizvoljan broj događaja u jednom krugu, moguće je postići optimalno rešenje u opštem slučaju (za proizvoljan odnos između m i n), ali tada problem postaje NP-težak. Stoga je glavni cilj istraživanja bio usmeren u pravcu nalaženja heuristike koja bi bila efikasnija od ostalih do sada ponuđenih rešenja, a zatim i demonstracije te efikasnosti.

U tom smislu, cilj ovog istraživanja usmeren je na razvoj centralizovanih i lokalizovanih algoritama koji pokrivaju slučajeve sa proizvoljnim brojem događaja i robota. To uključuje kako varijante uparivanja robota sa događajima, tako i varijante dodele robotima sekvenci događaja za obilazak, što predstavlja generalizaciju problema iz rada [MLMS13]. Stoga, pojedinačni izvršilac može preuzeti odgovornost za pronalaženje izvršilaca koji će opslužiti višestruke događaje koji se javljaju istovremeno. Roboti kao izvršioc i takođe mogu učestvovati u više konkurentnih aukcija. Kao dalja generalizacija, razmatran je i slučaj kada se javljaju događaji koji zahtevaju intervenciju od strane više izvršilaca.

Kada su u pitanju centralizovana rešenja, akcenat je na njihovoj efikasnosti u pogledu minimizacije dužina putanja koje prelaze roboti. Sa druge strane, kod distribuirane koordinacije moraju se uzeti u razmatranje i komunikacioni aspekti. Mehanizmi komunikacije koji obezbeđuju uslove za funkcionisanje predloženih distribuiranih rešenja trebaju da budu efikasni u pogledu količine mrežnog saobraćaja, u cilju

smanjenja potrošnje energije. Takođe, poželjno je da komunikacija bude lokalizovana, što doprinosi skalabilnosti sistema.

Prilikom evaluacije predloženih rešenja, izvršena je analiza složenosti algoritama i procena hardverskih resursa potrebnih za njihovo izvršavanje u realnom okruženju. Potom su obavljene opsežne simulacije, čime je na statistički relevantan način pokazana superiornost performansi novih algoritama u odnosu na do sada poznata rešenja.

1.5 Pregled literature

U ovom odeljku izložen je pregled literature, uz osvrt na aktuelno stanje u oblasti koja je vezana za tematiku ove disertacije. Pregled najznačajnijih radova iz oblasti sistema sa više robota prikazan je u odeljku [1.5.1](#), sa naglaskom na problem koordinacije. Tržišni pristup rešavanju problema koordinacije obrađivan je u radovima čiji najvažniji rezultati su sumirani u odeljku [1.5.1.2](#). Odeljak [1.5.2](#) bavi se komunikacionim aspektima koordinacije robota, sa akcentom na lociranje izvršilaca. Relevantni radovi i rešenja u vezi sa lociranjem izvršilaca izloženi su u odeljku [1.5.2.1](#).

1.5.1 Koordinacija u sistemima sa više robota

Detaljna analiza stanja u oblasti sistema sa više autonomnih mobilnih robota (engl. *multirobot systems*) prikazana je u poglavlju [[Par08](#)]. Istaknuta je motivacija za implementaciju ovakvih sistema:

- Složenost zadatka može biti prevelika da mi mogao biti uspešno rešen od strane pojedinačnog robota.
- Zadatak može biti inherentno distribuirane prirode.
- Izrada više robota sa ograničenim resursima može biti lakša i ekonomičnija od izrade jednog moćnog robota.
- Korišćenje paralelizma može dovesti do bržeg i efikasnijeg rešenja.
- Upotreba više robota povećava robusnost sistema uvođenjem redundantnosti.

Radovi [[DJMW96](#)] i [[Ver05](#)] takođe daju uvid u stanje i taksonomiju oblasti sistema sa više robota. Sistemi koji mogu biti sačinjeni od homogenih ili heterogenih robota se u opštem slučaju odlikuju zajedničkim osobinama, u koje spadaju kooperacija, svest o prisustvu drugih entiteta u sistemu, komunikacija i koordinacija. Studija [[CFK97](#)]

prepoznaje kolektivno ponašanje kao jedno od ključnih svojstava sistema sa više robota. Kolektivno ponašanje je generalna karakteristika koja se ispoljava u sistemima koji imaju više od jednog agenta. Kooperativno ponašanje je podklasa kolektivnog ponašanja, u čija osnovna svojstva spadaju: preduzimanje akcija od strane pojedinačnih agenata usmerenih ka postizanju kolektivnog cilja [BG91], interakcija između robota koja je obično bazirana na komunikaciji i udruživanje koje doprinosi povećanju performansi, ili uštedi vremena [PY90].

Problemi koje je potrebno rešiti tokom projektovanja sistema zavise od prirode zadatka i od senzorskih/aktuatorskih mogućnosti robota koji su na raspolaganju. Međutim, koordinacija između robota je jedno od ključnih pitanja koje se javlja u svim takvim sistemima. U radovima [RS08][BA98][CSKC01] predstavljena je nekolicina arhitektura namenjenih rešavanju problema koordinacije robota. Te arhitekture su po svojoj prirodi centralizovane, bazirane na klasterima, decentralizovane, lokalizovane ili hibridne. Većina postojećih rešenja je centralizovana u smislu da jedan od robota predstavlja centralni entitet koji sakuplja sve relevantne informacije i donosi odluke. Pri tome, retko se uzimaju u obzir komunikacioni troškovi koji proističu iz procesa prikupljanja informacija. Pošto se najčešće ne spominju nikakvi detalji vezani za komunikacione troškove, implicitno se podrazumeva postojanje kompletnog komunikacionog grafa, što znači da se bilo koja dva robota nalaze jedan u dometu drugog, čime im je omogućena direktna komunikacija.

Centralizovana rešenja najčešće svode problem koordinacije na linearno programiranje [NNY89][PA05]. Ključna prednost centralizovanog pristupa je u tome što je teoretski moguće pronaći optimalno rešenje. Sa druge strane, ovakva rešenja mogu da nametnu visoke troškove, kako po pitanju komunikacije, tako i u pogledu resursa potrebnih za njihovo izračunavanje. U tom kontekstu, često se neopravdano zanemaruju komunikacioni troškovi. Takođe, ostaje nejasan mehanizam komunikacije u slučajevima kada mrežni graf nije kompletan, ili ako nije povezan. Centralizovana rešenja se takođe odlikuju slabom robusnošću, pogotovo u slučaju bilo kakvog otkaza koordinatorskog čvora.

Hijerarhijske arhitekture zasnovane na klasterima [DGF⁺91] mogu biti implementirane samo pod određenim uslovima. Izvršiocu se grupišu u klastere, pri čemu se po jedan od njih u svakom klasteru bira za koordinatora (engl. *cluster head*). Izvršiocu na nižem nivou hijerarhije mogu komunicirati sa svojim koordinatorima, dok koordinatori komuniciraju među sobom u cilju donošenja odluka. Ovakva arhitektura je skalabilna, ali i dalje ima slab imunitet na otkaze od strane koordinatora. Ideja lokalizovane koordinacije u senzorskim mrežama zasnovane na udruživanju u klastere izložena je u članku [EGHK99]. Formiranje klastera omogućava postizanje globalnih ciljeva putem lokalne koordinacije, čime se postiže skalabilnost, robusnost i efikasna upotreba resursa u mnogim distribuiranim scenarijima senzorske koordinacije. Komunikacioni model koji je korišćen za opisivanje lokalnih interakcija nazvan je u tom radu direktnom difuzijom.

Decentralizovane kontrolne arhitekture [CFK97] tipično zahtevaju da izvršioци donose odluke na osnovu lokalnog znanja. Ovakav pristup je robustan i otporan na kvarove, pošto nijedan izvršilac nije odgovoran za kontrolu drugih izvršilaca. Međutim, postizanje globalne efikasnosti može predstavljati poteškoću u takvim sistemima, pošto viši ciljevi moraju biti inkorporirani u proces donošenja odluka kod svakog od izvršilaca. Ukoliko se neki od tih ciljeva promene, može biti komplikovano revidirati ponašanje pojedinačnih izvršilaca.

Hibridne kontrolne arhitekture [Par98] kombinuju lokalno donošenje odluka sa kontrolnim strategijama višeg nivoa, kako bi se postigla robusnost, dobra skalabilnost i mogućnost uticaja na akcije pojedinačnih izvršilaca promenom globalnih ciljeva.

Formalna analiza i klasifikacija problema dodele zadataka *MRTA* (engl. *Multi Robot Task Assignment*) u sistemima sa više robota izložena je u radu [GM04]. Analiza je prikazana iz perspektive operacionih istraživanja i kombinatorne optimizacije. Pri opisivanju *MRTA* problema, mogu se javiti sledeće situacije:

- Roboti mogu biti u stanju da simultano izvršavaju jedan zadatak (engl. *single-task robots*), ili višestruke zadatke (engl. *multi-task robots*).
- Zadaci mogu da zahtevaju intervenciju od strane jednog robota (engl. *single-robot tasks*), ili više robota (engl. *multi-robot tasks*).
- Dodeljeni zadaci se mogu izvršavati trenutno (engl. *instantaneous assignment*) ili odloženo (engl. *time-extended assignment*).

Softverska platforma koja omogućava koordinaciju višestrukih robota u nestrukturiranim i nepoznatim okruženjima opisana je u radu [FDS⁺02] i nazvana je *CHARON* (engl. *Coordinated Control, Hierarchical Design, Analysis, and Run-Time Monitoring of Hybrid Systems*). U pitanju je programski jezik visokog nivoa sa formalnom semantikom, dizajniran za primenu u umreženim sistemima sa više robota, sa više različitih načina upravljanja i estimacije ponašanja robota. Uz to, predviđeno je i korišćenje diskretnih komunikacionih protokola. Ova arhitektura omogućava razvoj sistema sa više robota sa kompleksnim ponašanjem, putem hijerarhijske i sekvencijalne dekompozicije. Funkcionalnost softverske platforme je demonstrirana na nekoliko primera u laboratorijskom okruženju, pri čemu su korišćeni mobilni autonomni roboti.

Primena koordinacije robota u cilju istraživanja nepoznatog terena razmatrana je u radu [BMSS05]. U tom radu predstavljen je algoritam čiji cilj je maksimizacija funkcije koristi (odnosno pokriva terena), uz istovremenu minimizaciju preklapanja između oblasti koje pokrivaju različiti roboti. Sličan problem razmatran je i u radu [SYTX06], gde je formulisana na sledeći način: n identičnih robota su zaduženi za istraživanje nepoznatog terena. Svaki robot opremljen je sensorima, uređajima za

lokalizaciju, mapiranje i komunikacionim uređajima ograničenog dometa. Autori su za rešavanje problema predložili distribuirani algoritam za koordinirano istraživanje terena. Taj algoritam zasnovan je na distribuiranom licitiranju i podrazumeva da svi roboti kreću sa početnih pozicija koje su međusobno bliske (u komunikacionom dometu) i da su njihove relativne pozicije poznate. Svaki robot kreće u istraživanje oblasti na takav način da maksimizuje informacionu dobit, uz minimizaciju cene koja je predstavljena rastojanjem između početne i ciljne pozicije.

U radu [KSV05] izložena je ideja koordinacije u dinamičkim sistemima sa više robota, korišćenjem grafova koordinacije. Pri tome je komunikacija između robota isključena, što se nadomešta metodama predikcije ponašanja drugih robota. Autori rada su efektno demonstrirali efikasnost predložene metodologije pobedom na međunarodnom takmičenju u fudbalu između timova humanoidnih robota *RoboCup 2003* [Pol04].

Problem pokrivanja terena (engl. *coverage*) [AAI09] je fokusiran na maksimizaciju površine terena sa koje se prikupljaju informacije, uz istovremenu minimizaciju preklapanja oblasti delokruga senzorskih uređaja¹¹. Ova problematika je intenzivno proučavana u istraživačkoj zajednici, uz značajan broj objavljenih publikacija. Strategije raspoređivanja mobilnih agenata u cilju efikasnog pokrivanja terena se generalno mogu klasifikovati u tri kategorije: pokrivanje zasnovano na virtuelnim silama [HMS02][ZC03], pokrivanje zasnovano na rešetkastim strukturama [SCS06][CIQC02][XTH06] i pokrivanje zasnovano na strukturama iz analitičke geometrije [MKPS05], u koje spadaju Voronoi dijagrami [Aur91] i Delaunay triangulacija [Che89]. Više reči o primenama Voronoi dijagrama u oblasti *BSM* biće u sledećem odeljku.

1.5.1.1 Primena Voronoi dijagrama u BSM

Protokol baziran na licitacijama i Voronoi dijagramima predstavljen je u radu [WCLP06]. Cilj protokola je maksimizacija površine koju pokrivaju senzori putem relokacije mobilnih senzora, tako da se minimizuju prazne površine (odnosno "rupe" u smislu pokrivenosti). Teren se deli na Voronoi ćelije sa po jednim sensorom u svakoj ćeliji. Senzori izračunavaju svoje Voronoi ćelije na osnovu lokalizovanog znanja pozicija svojih 1-hop i 2-hop suseda, a zatim sprovode postupak licitacije. Algoritmi koji kreću od sličnih pretpostavki prikazani su u radovima [AIRX08] i [ZJW13]. Još jedan rezultat koji se bavi distribuiranim Voronoi particionisanjem prikazan je u [GD12]. U ovom radu je opisan algoritam za lokalizovano kreiranje Voronoi dijagrama koji koristi komunikaciju sa susedima koji su udaljeni maksimalno dvostruko u odnosu na poluprečnik kruga koji je pokriven dometom senzora. Još jedan rad u kojem se sledi paradigma jedan senzor - jedna Voronoi ćelija je [SY13].

¹¹Prepostavka je da se koriste senzorski uređaji koji ne obavljaju merenja isključivo u svom neposrednom okruženju, nego pokrivaju oblast u određenom radijusu oko svoje trenutne pozicije.

Algoritmi pomenuti u prethodnom pasusu uglavnom su fokusirani na problem pokrivanja terena. Jedan od problema vezanih za njihovu upotrebu ogleda se u tome što kao pretpostavke za svoje funkcionisanje uzimaju stroge uslove u pogledu dometa komunikacije i prostorne distribucije čvorova, koji u opštem slučaju ne moraju biti ispunjeni. Uz to, ovi algoritmi sami po sebi ne obezbeđuju komunikacione mehanizme kojim bi bilo omogućeno detektovanje pozicija robota od strane statičkih senzorskih čvorova.

Algoritam koji omogućava robotima izračunavanje Voronoi dijagrama na distribuiran način prikazan je u članku [CH02]. U ovom slučaju sami roboti predstavljaju generatore ćelija¹². Ovaj algoritam se zasniva na pretpostavci da je svaki robot svestan pozicija drugih robota koji su udaljeni od njega do određene razdaljine na kojoj se može primeniti kriterijum stopiranja, koji prekida dalje izvršenje algoritma. Time se implicitno podrazumeva korišćenje komunikacije dugog dometa među robotima, što je pretpostavka koja se značajno razlikuje od one na kojoj počiva *vCell* protokol koji smo predstavili u radu [LM15], o čemu je detaljno elaborirano u odeljku 2.3. Naime, *vCell* se oslanja isključivo na komunikaciju kratkog dometa. Kao što će kasnije biti detaljnije objašnjeno, umesto konstrukcije Voronoi dijagrama kao takvog, *vCell* formira komunikacionu i informacionu strukturu među statičnim senzorskim čvorovima koja se sastoji iz ćelija koje svojim oblikom podražavaju Voronoi ćelije. Uz to, u ovom slučaju roboti ne moraju raspolagati informacijom o pozicijama, pa čak ni o postojanju drugih robota.

Nedavno je objavljeno još nekoliko interesantnih radova koji se bave primenama Voronoi dijagrama u različitim aplikacijama iz oblasti *BSM*. Šema za podešavanje dubine u podvodnim 3D mrežama je prikazana u radu [WWL13]. U radu [CPG⁺14] predložena je težinska lokalizaciona šema (*W-VBLS*), kao ekstenzija *VBLS* lokalizacione šeme zasovane na Voronoi dijagramima. U ovoj šemi se težinski mrežni graf formira na osnovu merenja jačine primljenog signala (*RSSI* indikator). Klasterovanje senzora u cilju agregacije podataka tema je studije izložene u [NK13]. U ovom radu predstavljeno je centralizovano multi-hop klasterovanje bazirano na fazi logici (*V-FCM*) algoritam.

Jedno od rešenja problema energetski efikasnog raspoređivanja mobilnih senzora po površini terena prikazano je u radu [HV05]. Ovde je akcenat na organizaciji i optimizaciji mrežnih resursa u cilju produženja životnog vremena sistema kao i migracije inteligencije sa globalnog na lokalni nivo. Distribuirani algoritam za raspoređivanje baziran je na Voronoi dijagramima.

¹²Pojam generatora je detaljnije objašnjen u odeljku 2.1.

1.5.1.2 Tržišne metode koordinacije robota

U poslednjih nekoliko godina, sve su popularnije tržišne metode koordinacije robota, čija formulacija, pregled i analiza su izloženi u radovima [GM02a][DZKS06]. Da bi mehanizam koordinacije između robota mogao biti smatran tržišno orjentisanim, od njega se očekuje da ispunjava sledeće kriterijume:

- Robotskom timu je postavljen zadatak koji može biti rastavljen na manje celine koje izvršavaju individualni roboti, ili podgrupe. Da bi uspešno odradio posao, tim ima na raspolaganju ograničen skup resursa koji je raspodeljen među članovima tima.
- Globalna funkcija cilja kvantifikuje preference dizajnera sistema, na osnovu čega se može uspostaviti hijerarhija među svim mogućim rešenjima.
- Individualne funkcije koristi za svaki od robota kvantifikuju preference tih robota u pogledu korišćenja resursa i njihovih individualnih doprinosa globalnom cilju. Izračunavanje vrednosti ove funkcije ne sme da zahteva globalnu informaciju o trenutnom stanju tima ili stanju ukupnog cilja. Preference na nivou podgrupe se kvantifikuju kombinovanjem individualnih koristi njenih članova.
- Definisana je veza između ciljne funkcije celog tima i funkcija koristi na pojedinačnom nivou ili nivou podgrupe. Ova veza pokazuje na koji način se individualna proizvodnja/potrošnja resursa i individualni napredak prema globalnom cilju odražavaju na ukupno rešenje.
- Resursi i individualni ciljevi, odnosno ciljevi na nivou podgrupe mogu biti preraspodeljeni korišćenjem mehanizma kao što su aukcije. Ovaj mehanizam na ulazu ima individualne funkcije koristi i izračunava izlaz na način koji maksimizuje korist agenta koji kontroliše sam mehanizam. U dobro dizajniranom mehanizmu, maksimizacija koristi kontrolnog agenta rezultuje poboljšanjem vrednosti ukupnog cilja.

Teoretska analiza performansi aukcijskih metoda koordinacije timova robota prikazana je u radu [LMK⁺05]. Matematički model izložen u radu [LJGM06] omogućava implementaciju mehanizma dinamičke dodele zadataka u sistemima sa više robota.

Centralizovani i distribuirani algoritmi za dodelu zadataka zasnovani na aukcijama koji su prikazani u radu [CV10] kao infrastrukturu za dodelu zadataka koriste minimalno stablo razapinjanja konstruisano nad mrežom statičkih senzorskih čvorova.

Pristup koji je korišćen u radu [TP07] podrazumeva da roboti prvo formiraju koalicije na nižem nivou, u cilju rešavanja pojedinačnog zadatka. Zatim, koalicije i eventualno individualni roboti ulaze u nadmetanje za zadatke na višem nivou hijerarhije.

Ekonomski pristup koordinaciji robota zasnovan na konceptu slobodnog tržišta izložen je u radu [DS00]. Tržište je definisano kroz funkcije prihoda i cene, koje zavise od prirode zadatka koji je potrebno rešiti. Zadatak se deli na potprobleme, a robotima je dozvoljeno da se nadmeću i da pregovaraju oko njihovog izvršenja. Kooperacija i nadmetanje se odvijaju kao posledica težnje robota da izvrše zadatak maksimizujući pri tome svoje individualne profite.

Višeslojna distribuirana arhitektura za upravljanje i koordinaciju u sistemima sa više robota opisana je u članku [GCD⁺03]. U pitanju je proširenje tradicionalne troslojne arhitekture koja omogućava robotima direktnu komunikaciju na sva tri sloja (nivoa): na bihejvioralnom nivou kreiraju se distribuirane kontrolne petlje, na izvršnom nivou sinhronizuje se izvršenje zadataka i konačno, na nivou planiranja koriste se tržišne tehnike za raspodelu zadataka i alokaciju resursa. Tržišno orjentisani sloj zadužen za planiranje, ima dve osnovne komponente. Prva komponenta je "trgovac" koji učestvuje u licitacijama, a druga je "raspoređivač", koji određuje izvodljivost zadatka, zadaje cene trgovcu i vrši interakciju sa izvršnim slojem, koji je zadužen za izvršenje zadataka. Ovakva arhitektura je primenjena i eksperimentalno proverena u scenariju istraživanja Marsa, gde je cilj bio prikupljanje i karakterizacija različitih vrsta kamenja sa površine ove planete.

U radu [GM02b] predstavljen je aukcijski metod dinamičke dodele zadataka robotima nazvan *MURDOCH*. U pitanju je varijanta *CNP* protokola [DS83] koja daje distribuiranu aproksimaciju globalnog optimuma, kada je u pitanju upotreba resursa. Primarni doprinos ovog rada je empirijska demonstracija distribuiranih mehanizama pregovaranja u multirobotским sistemima. Uz to, ovo je jedan od fundamentalnih i najčešće citiranih radova iz oblasti tržišne koordinacije robota.

Markovljevi procesi odlučivanja *MDP* (engl. *Markov Decision Process*) su kombinovani sa tržišnim arhitekturama u radu [BGT03]. Predstavljen je algoritam koji za klasu Markovljevih procesa odlučivanja kreira tržišnu arhitekturu koja dovodi do toga da decentralizovani sistem sa više robota konvergira ka konzistentnom rešenju. Takođe, pokazano je da je tako dobijeno rešenje identično onom koje bi bilo dobijeno primenom odgovarajućeg centralizovanog algoritma planiranja¹³. Primena algoritma je prikazana na primerima zadataka vuče objekta od strane više robota, planiranja robotskih putanja u uslovima ograničene količine energije za kretanje i koordinisanog timskog ponašanja u igri "*Paintball*".

Jedno od rešenja problema istraživanja terena od strane robotskog tima, o kojem je bilo reči u prethodnom odeljku, izloženo je u članku [ZSDT02]. Koordinacija među robotima se postiže tržišnim pristupom, koji podrazumeva da roboti u kontinuitetu pregovaraju među sobom, unapređujući pri tome svoje trenutne planove i deleći informacije o tome koji regioni su već pokriveni. Takav pristup je funkcionalan i u uslovima neidealne, ili čak nepostojeće komunikacije. Uz to, iako postoji centralni

¹³Kao referentno centralizovano rešenje se uvek uzima neka od varijanti linearnog programiranja.

agent, sistem se ne oslanja isključivo na njega i nastavlja da funkcioniše čak i kada ostali roboti nisu u mogućnosti da komuniciraju s njim. Centralni agent ima ulogu posrednika između robotskog tima i ljudskog operatora, što znači da ovakvo rešenje nije u potpunosti automatizovano.

Još jedan tržišni pristup usmeren ka problemu istraživanja terena je prikazan u radu [BHK⁺03]. Ovaj pristup se zasniva na kombinatornim aukcijama u kojima se roboti nadmeću oko grupa zadataka, umesto oko pojedinačnih zadataka. Prednost kombinatornih aukcija dolazi do izražaja u sistemima u kojima postoji tzv. pozitivna sinergija između zadataka, koja se očituje time da je kombinovana vrednost grupe ciljeva veća od suma njihovih individualnih vrednosti. Međutim, mehanizam grupisanja zadataka je problematičan, pošto se kreće od pojedinačnih zadataka, zatim se razmatraju sve moguće grupe od 2, 3 i više zadataka, što dovodi do eksponencijalne složenosti algoritma. Predloženo je nekoliko strategija za prevazilaženje ovog problema, koje dovode do sub-optimalnog particionisanja skupa zadataka u polinomijalnom vremenu.

Studija [TLJK05] predstavlja pokušaj sistematskog iznalaženja metoda za automatsko izvođenje pravila licitiranja koja se primenjuju u aukcijskim metodama koordinacije. Predložena metodologija je demonstrirana na primerima tri različita timska cilja u okviru scenarija istraživanja terena pomoću robotskog tima: minimizacija ukupne dužine putanja svih robota, minimizacija dužine maksimalne pojedinačne robotske putanje i minimizacija prosečnih troškova vezanih za obilazak pojedinačnog cilja. Prvi cilj, odnosno minimizacija ukupne dužine putanja, poklapa se sa ciljem koji je postavljen u okviru teme ove disertacije.

1.5.2 Komunikacioni aspekti koordinacije

Kao što je naglašeno u odeljku 1.5.1, većina razmatranih pristupa koordinaciji robota zanemaruje komunikacione aspekte. Uticaj ograničenja koja su nametnuta bežičnom mrežnom komunikacijom na koordinaciju razmatran je u radu [RB07]. Predložen je centralizovani algoritam u kome roboti komuniciraju sa baznom stanicom, čime se implicitno podrazumeva zvezdasta topologija mreže. Bazna stanica donosi odluke o narednom potezu robota, na osnovu svih raspoloživih mogućnosti, korišćenjem funkcije koristi.

U radu [MPGA07], predstavljen je model komunikacione infrastrukture za koordinaciju između senzora i robota, u skladu sa paradigmom podele mreže zasnovane na događajima. Sensorski čvorovi se grupišu u klastere koji se formiraju na osnovu stabala koja se koriste za prosleđivanje podataka izvršiocima. Razmatrana su dva potproblema: koordinacija sensor-izvršilac i koordinacija izvršilac-izvršilac. Za prvi potproblem prikazano je optimalno rešenje zasnovano na celobrojnom linearnom programiranju (engl. *ILP*) [RMO93], kao i distribuirana varijanta rešenja. Kada je u

pitanju koordinacija izvršilac-izvršilac, problem je formulisan kao mešovito celobrojno nelinearno programiranje (engl. *MINLP*) [CMM98], a prikazano je i distribuirano rešenje bazirano na aukcijama. U oba slučaja, koristi se matematičko programiranje kako bi se dobilo rešenje koje odgovara višestrukim kriterijumima optimizacije kao što su minimizacija potrošnje energije i propagacionog kašnjenja pri komunikaciji, uzimajući u obzir pouzdanost linkova, troškove prosleđivanja poruka i sl.

Aukcijski pristup koordinaciji u slučaju jednog događaja i višestrukih izvršilca, sa naglaskom na komunikaciju u robotskoj mreži razmatran je u radu [MMS10]. Osnovni protokol nazvan je *SAP* (engl. *Simple Auction Protocol*) i zasnovan je na protokolu iz rada [MPGA07]. Unapređenja se dobijaju na osnovu aukcijske agregacije podataka, uz ograničenje komunikacije na k hopova u svrhu postizanja lokalnosti. Predloženo je pet novih protokola nazvanih k -*SAP* (engl. *k-hops Simple Auction Protocol*), *SAAP* (engl. *Simple Auction Aggregation Protocol*), k -*SAAP* (engl. *k-hops Simple Auction Aggregation Protocol*), k -*AAP* (engl. *k-hops Auction Aggregation Protocol*) i *RFT* (engl. *Routing with Face Traversal*). Vezano za tematiku ove disertacije, najznačajniji je k -*SAAP*, koji je kombinovan sa *iMesh* protokolom za lociranje izvršilaca u radu [MLMS13].

1.5.2.1 Lociranje izvršilaca

U nedavno objavljenom preglednom radu [CD13] izložena je problematika lociranja robota kao izvršilaca zadataka iz dve različite perspektive. Detekcija statičkih događaja od strane robota nazvana je propitivanjem (engl. *querying*), dok je detekcija robota od strane statičkih čvorova nazvana praćenjem (engl. *tracking*). Postojeća rešenja su po svojoj prirodi kategorizovana u više grupa: rešenja zasnovana na stablima, hijerarhijska rešenja zasnovana na klasterima, geometrijska i mešovita rešenja. Poželjne osobine protokola za propitivanje i/ili praćenje su osetljivost na razdaljinu, energetska efikasnost, skalabilnost i otpornost na otkaze.

Problematika lociranja izvršilaca (engl. *location service, service discovery*) analizirana je u radu [LNS10]. Klasifikacija protokola je zasnovana na onoj koja je predstavljena u radu [DPH07]. Algoritmi za lociranje izvršilaca svrstavaju se u tri kategorije: algoritmi zasnovani na preplavljanju [APL99][FK05][SRL06][KFHM02], algoritmi zasnovani na kvorumu [PHS97][KAS98][SLJ08] i algoritmi zasnovani na baznoj regiji (engl. *home region*) [Sto99][LJDC⁺00b]. Problem lociranja izvršilaca podrazumeva da izvršioc (roboti) oglašavaju svoje pozicije, dok korisnici usluge (senzorski čvorovi koji detektuju događaje) pokušavaju da saznaju trenutne pozicije izvršilaca. Neophodno je minimizovati ukupne troškove u smislu broja poslatih poruka, uz istovremenu maksimizaciju uspeha u pronalaženju optimalnog (najčešće najbližeg) raspoloživog izvršioca. U prethodnih nekoliko godina predloženo je više algoritama za lociranje izvršilaca [MBB09][VP08]. Postojeća rešenja mogu biti direktno primenjena u oblasti bežičnih mreža senzora i robota, ali nisu uvek primenljiva u *BMSR* sa

ograničenim resursima, budući da su neka od tih rešenja skopčana sa velikim komunikacionim troškovima i/ili neravnomernom raspodelom opterećenja među mrežnim čvorovima u pogledu memorijskog prostora. Pored već pomenute klasifikacije algoritama za lociranje izvršilaca, postoji i druga, koja ih kategorizuje u dve grupe: algoritmi koji koriste direktorijum i algoritmi bez direktorijuma [ZMN05]. Algoritmi koji koriste direktorijum zasnivaju se na specijalizovanim informacionim strukturama u kojima su smeštene relevantne informacije o izvršiocima, koje omogućavaju njihovo otkrivanje. Uz to, ti algoritmi obično zahtevaju i globalnu komunikaciju/procesiranje u cilju formiranja klastera i dominantnog skupa [YKR06], kreiranja i održavanja informacionog direktorijuma. Algoritmi koji ne koriste direktorijum ne moraju da održavaju informacionu strukturu, nego se zasnivaju na periodičnom oglašavanju prisustva izvršilaca. Njihovo izvršenje često uključuje operacije ograničenog preplavljanje mreže.

Dok je problem lociranja izvršilaca orjentisan na korisnike usluge, kao što je već napomenuto na početku odeljka, isti problem može biti posmatran iz perspektive izvršilaca. Propitivanje [CD13] je proces detekcije statičkih događaja od strane mobilnih izvršilaca. Glavne komponente sistema koji koristi ovakav pristup su čvor koji zadaje upit (u ovom slučaju robot), poruke kojim se vrši propitivanje i čvor koji detektuje događaj. Robot inicira propagaciju upita kroz mrežu, kako bi došao do informacije o prisustvu događaja. Poruka sadrži polje koje definiše koja vrsta informacije se prenosi kroz mrežu. Izvor podataka, odnosno čvor koji detektuje događaj čuva i prosleđuje dalje u mrežu podatke o događaju. Kada poruka koja sadrži upit dosegne čvor koji skladišti relevantnu informaciju o događaju, generiše se odgovor koji se prosleđuje nazad robotu koji je inicirao upit. Ako se koristi centralizovani pristup, upiti se prosleđuju baznoj stanici koja prikuplja informacije o događajima. Korišćenje jedne centralne bazne stanice za sve upite može biti neefikasan pristup, pogotovo u slučaju ako se događaj desio u blizini robota koji je pokrenuo upit. Uz to, korišćenje bazne stanice za skladištenje i prosleđivanje podataka može dovesti do pojave komunikacionih uskih grla. Stoga je tendencija da se rešenja baziraju na pronalaženju kratkih (lokalnih) komunikacionih putanja između robota i čvorova koji detektuju događaje.

Neki skorašnji radovi [VDA13], [VDdA08] bave se problemom prikupljanja podataka. Mobilni kolektori (roboti) prikupljaju podatke statičkih senzora kretanjem duž putanja koje optimizuju pokriće terena koji se posmatra. Ovo se razlikuje od problematike o kojoj je reč u ovoj disertaciji, gde se podrazumeva da se prikupljanje podataka obavlja na statički način, što isključuje kretanje robota.

GLS (engl. *Grid Location Service*) protokol vrši podelu terena rešetkastom strukturom i konstruiše kvad-stablo¹⁴ nad ćelijama rešetke [LJDC⁺00a]. Korišćenjem *hash* funkcije kreirane na osnovu kvad-stabla, svakom od čvorova dodeljuje se jedinstven

¹⁴Kvad-stablo je graf sa strukturom stabla u kome svi čvorovi osim onih na poslednjem nivou imaju tačno 4 potomka.

podskup čvorova koji se nazivaju lokacionim serverima. Svaki čvor obaveštava lokacione servere o svojoj trenutnoj lokaciji. Čvor saznaje lokaciju proizvoljnog drugog čvora propitivanjem nekog od njegovih lokacionih servera. Ovaj protokol zahteva da je unapred poznata topologija mreže, kako bi mogla biti izvršena njena podela na ćelije. Mana ovog protokola je u tome što može da izazove velike komunikacione troškove, budući da se propagacija upita vrši duž izlomljenih linija.

Zonsko klasterovanje robota [KKC⁺06] je metoda kontrole topologije sa ciljem ujednačavanja potrošnje energije među senzorskim čvorovima, bez generisanja izolovanih senzorskih čvorova. Princip klasterovanja zasnovan je na podeli mreže na zone, u zavisnosti od rastojanja od pojedinačnog robota, koji ima ulogu informacionog kolektora. U svakoj zoni bira se odgovarajući broj lokalnih koordinatora, u zavisnosti od veličine zone. Podaci sa senzora prosleđuju se lokalnim koordinatorima, koji obavljaju agregaciju podataka i time dodatno smanjuju količinu mrežnog saobraćaja.

GHT (engl. *Geographic Hash Table*) šema za smeštanje podataka predstavljena je u radu [RKY⁺02]. Čvor smešta podatke u *hash* tabelu na način koji je određen tipom podataka i prosleđuje te podatke korišćenjem kombinovanog *GFG* protokola [BMSU01][FS06]. Osnovni problem sa ovom šemom je to što dovodi do neželjenih ne-lokalnih upita, što znači da može da se desi da podaci prevale veliki broj hopova da bi dostigli čvor koji je lociran relativno blizu izvora podataka. Takođe, moguća je pojava komunikacionih uskih grla, ukoliko se neki tipovi podataka često generišu ili potražuju.

GCLP (engl. *Geography based Content Location Protocol*) koristi servere, koji periodično oglašavaju svoje prisustvo u 4 geografska pravca [TV04]. Čvorovi koji prime poruke kojima serveri oglašavaju prisustvo postaju lokacijski serveri. Ako lokacijski server primi obaveštenja od strane više servera, dalje se prosleđuje samo ono koje potiče od najbližeg servera. Ovakav način prosleđivanja informacije predstavlja neformalnu definiciju blokirajućeg pravila, koje je korišćeno pri kreiranju *iMesh* strukture [LSS09]. Usled periodičnog razglašavanja lokacija, *GCLP* generiše velike troškove u pogledu mrežne komunikacije.

Šema za smeštanje i preuzimanje podataka bazirana na orjentirima (engl. *landmarks*) predstavljena je u radu [FGG06]. U okviru ove šeme, generišu se stabla koja predstavljaju najkraće informacione putanje do unapred definisanih čvorova koji imaju ulogu orjentira. Izvor podataka smešta podatke u *hash* tabelu tako da se u zavisnosti od tipa podataka oni šalju određenom orjentirskom čvoru, a podaci se raspedeljuju korišćenjem generisanog stabla, ukorenjenog u orjentirskom čvoru. Čvor koji zahteva podatak šalje upit putem iste strukture stabla. Potencijalan problem kod ovakve šeme je zatrpavanje pojedinih čvorova velikom količinom podataka i korišćenje (skupih) globalnih operacija. Takođe, ovakvo rešenje je energetski neefikasno i nije skalabilno.

Ideja slična onoj izloženoj u prethodnom pasusu je nedavno predstavljena u radu [TJZ⁺13]. Predmet interesovanja je šema za lokalizaciju u 2D i 3D mrežama

nepravilnog oblika. Ključna ideja u osnovi predloženog *CATL* protokola je detekcija isturenih čvorova, koji se nalaze na mestima gde informacione putanje naglo menjaju pravac i gde udaljenost bazirana na broju hopova počinje značajno da odstupa od Euklidske udaljenosti. Razvijen je iterativni protokol u kojem se koristi multilateralni mehanizam lokalizacije koji izbegava isturene čvorove. Iako ovaj pristup ne zahteva skup lokalizacioni hardver, njegov glavni nedostatak se upravo ogleda u nemogućnosti realne fizičke lokalizacije čvorova (umesto toga, dobijaju se virtuelne koordinate). Uz to, zahteva se slanje značajno većeg broja poruka po čvoru u odnosu na druge lokalizacione šeme koje su ovde pomenute. U pojedinim istraživanjima, virtuelne koordinate su kombinovane sa dekompozicijom mreže, u svrhu rutiranja u mrežama nepravilnog oblika [TBK09][KT10].

Lociranje izvršilaca naziva se još i informacionim posredovanjem (engl. *information brokerage*) [LKL12], gde je glavno pitanje razmena informacija između korisnika usluge i izvršilaca. Postoje dva trivijalna rešenja problema. Prvo je da upit preplavljuje mrežu, pa se čeka na odgovore izvršilaca. Drugi trivijalni pristup je da se informacije o izvršiocima prosleđuju svim čvorovima u mreži, čime korisnici usluge u startu raspolazu potrebnim informacijama, bez potrebe za prosleđivanjem upita.

U radu [LKL12], predstavljen je nov protokol za informaciono posredovanje (prosleđivanje informacija o izvršiocima korisnicima njihovih usluga), nazvan *RDRIB*. Korisnici usluga dolaze do podataka za čije držanje i repliciranje je zadužen određen skup čvorova. Za razliku od drugih sličnih šema, *RDRIB* garantuje uspešno pronalaženje informacije bez korišćenja algoritama za detekciju ivice i saznavanje geografskog položaja izvršilaca, dobijenog korišćenjem lokalizacionog hardvera (*GPS*).

Šema za informaciono posredovanje izložena u radu [SZG09] smešta informacije o izvršiocima duž krive linije, umesto u jednom ili više izolovanih čvorova. Korisnici putuju duž druge krive koja se garantovano seče sa pomenutom krivom. Ova šema se prirodno naslanja na *GHT*, uz poboljšanu lokalnost upita.

Metod informacionog posredovanja koji koristi hijerarhijsku dekompoziciju predstavljen je u radu [FGNW06]. Ovde se među čvorovima uspostavlja hijerarhija klastera, u kojoj svaki čvor pripada po jednom klasteru na svakom nivou. Informacije o izvršiocima repliciraju se u *hash* čvorovima u svim susednim klasterima na svim nivoima. Podaci se dobijaju propitivanjem *hash* čvorova u klasterima u kojima obitavaju izvršioci, u rastućem poretku hijerarhije nivoa, sve dok se ne pronađe *hash* čvor koji poseduje informaciju od interesa. Hijerarhijska dekompozicija garantuje uspešno pronalaženje informacije. Međutim, to se ostvaruje na račun značajnih troškova u pogledu komunikacije i potrošnje memorije, budući da je neophodno globalno znanje topologije.

Metode lociranja izvršilaca bazirane na kvorumu [SLJ08] i iMesh strukturi [LSS09] biće detaljnije opisane u narednom odeljku, budući da su od posebnog značaja za tematiku ove disertacije.

1.6 Postojeća rešenja

U ovom odeljku biće detaljnije opisana relevantna postojeća rešenja problema izloženih u prethodnim odeljcima. Neka od njih poslužiće kao osnova čijim unapređenjem će biti izvedeni novi algoritmi, dok će druga služiti kao referenca za ocenu performansi. Poređenje njihovih performansi sa novim rešenjima opisanim u poglavljima 2 i 3 putem simulacija biće prikazano u poglavlju 4.

1.6.1 Uparivanje robota sa događajima

Sledi prikaz postojećih rešenja koja se odnose na problem uparivanja robota sa događajima (problem m -*dodela*(m, n), definisan u odeljku 1.3). Ova rešenja su po svojoj prirodi centralizovana, budući da je za njihovo izvršenje neophodno globalno znanje o robotima i događajima. Pri tome se zanemaruju komunikacioni aspekti procesa prikupljanja informacija, o kojima će biti reči u odeljku 1.6.2.

1.6.1.1 Uparivanje u bipartitnom grafu sa minimalnom dužinom ivica - Mađarski algoritam

Poznati problem uparivanja u bipartitnom grafu sa minimalnom ukupnom težinom ivica, moguće je u kontekstu *BMSR* formulisati na sledeći način: težina pridružena ivici $w(i, j)$ je "cena" koju je potrebno platiti da bi se robot R_i pomerio na mesto događaja E_j . Na raspolaganju je više različitih opcija kada je u pitanju metrika na osnovu koje se određuju težine ivica u grafu. Među tim metrikama, najočigledniji izbor je rastojanje između robota i mesta događaja. Ako postoji n robota i n događaja, svakom robotu dodeljuje se tačno jedan događaj, na takav način da je ukupna težina ivica koje odgovaraju izabranim parovima minimalna. Problem m -*dodela*(m, n) je lako svesti na upravo opisani problem. Ukoliko je $m < n$, u skup događaja se dodaje još $n - m$ događaja, takvih da je težina ivica koja ih povezuje sa svim robotima jednaka nuli. Zatim se izvršava algoritam za optimalno uparivanje, pri čemu dodatni "virtuelni" događaji ne doprinose ukupnoj težini, budući da je svakom od njih pridružena nulta težina ivice. Nakon što je izvršeno optimalno uparivanje robota sa događajima, oni roboti koji su upareni sa nepostojećim događajima dodatim u svrhu izjednačavanja kardinalnosti ostaju na svojim pozicijama, dok ostali roboti odlaze na mesta događaja sa kojima su upareni.

Kao što je već napomenuto u poglavlju 1, *Mađarski algoritam*, u literaturi poznat još i kao *Mađarski metod* (algoritam 1), vrši optimalno uparivanje po kriterijumu minimizacije ukupne težine ivica, u vremenu $O(n^3)$. Ovo rešenje je originalno predložio Jacobi (posthumno je objavljeno 1890. godine), a kasnije je nezavisno otkriveno i od strane dvojice Mađarskih matematičara (König i Egervary). Detaljna analiza i predlozi poboljšanja predloženi su od strane Kuhna [Kuh55]. Takođe, postoje mnogi algoritmi koji daju sub-optimalna rešenja, kao što je distribuirani algoritam prikazan u radu Bertsekasa [Ber88].

Kompletan bipartitni težinski graf koji predstavlja ulaz algoritma predstavljen je kvadratnom matricom $W_{n \times n}$, u kojoj je $w_{i,j} = |R_i E_j|$:

$$W_{n,n} = \begin{pmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,n} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n,1} & w_{n,2} & \cdots & w_{n,n} \end{pmatrix}$$

Algoritam 1 Mađarski algoritam [Kuh55]

Ulaz: matrica W

Izlaz: skup uređenih parova M

- 1: u svakoj vrsti matrice W oduzmi minimalni element vrste od svih elemenata
 - 2: u svakoj koloni matrice W oduzmi minimalni element kolone od svih elemenata
 - 3: $l \leftarrow n + 1$
 - 4: **while** $l \neq n$ **do**
 - 5: duž vrsta/kolona matrice W povuci minimalan broj linija koje pokrivaju sve nulte elemente; $l \leftarrow$ broj potrebnih linija
 - 6: **if** $l < n$ **then**
 - 7: $k \leftarrow$ najmanji ne-nulti element matrice W
 - 8: oduzmi k od svakog nepokrivenog elementa i dodaj k svakom elementu koji je pokriven dvostruko (horizontalnom i vertikalnom linijom)
 - 9: **end if**
 - 10: **end while**
 - 11: izaberi n nultih elemenata matrice W , tako da se u svakoj vrsti i svakoj koloni nalazi tačno po jedan takav element; na osnovu indeksa izabranih elemenata konstruiši skup uređenih parova M
-

Izlaz algoritma je skup uređenih parova $M = \{(R_i, E_j) : R_i \in R \wedge E_j \in E\}$, tako da $(\forall R_i \in R)(\exists E_j \in E)((R_i, E_j) \in M)$ i $(\forall (R_i, E_j) \in M)(\forall (R_k, E_l) \in M)(i \neq k \Rightarrow j \neq l)$. Drugim rečima, skup M sadrži n disjunktivnih ivica koje pripadaju kompletnom bipartitnom grafu konstruisanom nad skupovima R i E . Pri tome važi uslov:

$$\sum_{(R_i, E_j) \in M} w_{i,j} = \min$$

1.6.1.2 Problem stabilnih brakova

Akkaya, Guneydas i Bicak u svom radu [AGB10] razmatrali su problem uparivanja izvršilaca sa događajima. Zadati su skupovi od n žena i n muškaraca, pri čemu svaka osoba ima listu preferenci u okviru koje svakoj osobi suprotnog pola dodeljuje jedinstveni rang od 1 do n . Potrebno je upariti žene sa muškarcima na takav način da ne postoji par žena-muškarac koji jedno drugo preferiraju više nego svoje trenutne partnere.

U kontekstu uparivanja robota sa događajima, roboti preuzimaju ulogu muškaraca, a događaji ulogu žena¹⁵. Pri tome se liste preferenci formiraju na osnovu Euklidskog rastojanja između robota i mesta događaja. Algoritam stabilnih brakova *SB*[GS62] (algoritam 2) obezbeđuje stabilno uparivanje i kao takav predstavlja centralizovano rešenje problema *m-dodela(m,n)*, za slučaj $n \geq m$. Vremenska složenost ovog algoritma je $O(n^2)$.

Algoritam 2 Algoritam SB [AGB10][GS62]

Ulaz: skup robota R i skup događaja E

Izlaz: skup uređenih parova M koji predstavljaju stabilne brakove

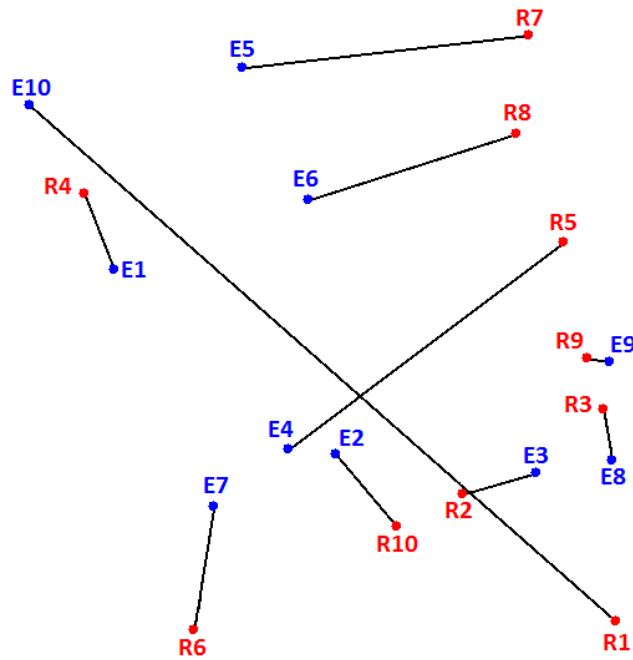
```

1: svi roboti  $r \in R$  i svi događaji  $e \in E$  se inicijalizuju kao slobodni;  $M = \emptyset$ 
2: while postoji slobodan robot  $r$  do
3:    $e$  je događaj najbliži robotu  $r$ 
4:   if  $e$  nije uparen ni sa jednim robotom then
5:     formiraj par  $(r, e)$  i dodaj ga u  $M$ 
6:   else
7:     već postoji neki par  $(r', e) \in M$ 
8:     if  $|re| < |r'e|$  then
9:       u skupu  $M$  par  $(r, e)$  zamenjuje prethodno formiran par  $(r', e)$ 
10:       $r'$  postaje slobodan
11:    else
12:      par  $(r', e)$  ostaje u skupu  $M$ 
13:      udaljenost  $(r, e)$  se postavlja na  $\infty$ 
14:    end if
15:  end if
16: end while

```

Na slici 1.7 prikazan je primer formiranja parova robota i događaja pomoću *SB* algoritma, za slučaj kada je $n = 10$. Iako je dobijeno rešenje stabilno, ono ne mora biti i optimalno. Ključni nedostatak ovakvog rešenja je pojava dugačkih ivica, kao što je $R_1 - E_{10}$ na slici. Ideja za unapređenje je u tome da se dozvoli da neki od parova robot-događaj razmene partnere, ako to dovodi do smanjenja ukupne dužine ivica, odnosno putanja koje roboti trebaju da pređu. Ovaj princip predstavlja osnovu za *PDM* algoritam koji će biti opisan u poglavlju 3.

¹⁵Ove uloge mogu i da se zamene, što ne utiče na stabilnost rešenja.



Slika 1.7: Uparivanje robota sa događajima pomoću SB algoritma

1.6.1.3 Maksimalno uparivanje

Wang, Peng i Tseng u radu [WPT10] izložili su algoritam koji vrši uparivanje robota sa događajima, vodeći pri tome računa o tri kriterijuma optimizacije:

- Broj parova je maksimalan.
- Suma težina pridruženih ivicama (tj. njihovih dužina) treba da bude što je moguće manja (na ovaj način minimizuje se ukupna količina energije koja se troši na kretanje robota).
- Standardna devijacija dužina ivica treba da bude što je moguće manja (ujednačavanjem dužine ivica postiže se ravnomernija raspodela opterećenja među robotima).

Budući da je ulaz algoritma kompletan bipartitni graf, prvi kriterijum je uvek ispunjen pošto se formira tačno m parova¹⁶. Što se tiče dva preostala kriterijuma, može se desiti da su zahtevi međusobno suprotstavljeni, tako da najčešće nije moguće postići optimum po oba kriterijuma. Uprkos tome, predloženo rešenje koje je prikazano u algoritmu 3 i koje će ovde biti označeno skraćenicom *MM* (engl. *Maximum Matching*), osmišljeno je sa idejom da u što većoj meri zadovolji oba prethodno pomenuta kriterijuma optimizacije.

¹⁶Autori [WPT10] su uveli pretpostavku da je $n \geq m$, tako da algoritam svakom događaju dodeljuje po jedan robot, a u slučaju kada je $n > m$, $n - m$ robota neće biti upareni s događajima.

Algoritam 3 Algoritam $MM(\beta)$ [WPT10]**Ulaz:** skup robota R , skup događaja E i parametar β **Izlaz:** skup uređenih parova M

- 1: svaki događaj E_j sortira sve robote na osnovu udaljenosti od njih (u rastućem poretku)
- 2: $B_j \leftarrow |R_i E_j|$, tako da je R_i robot na poziciji β u listi preferenci događaja E_j ; ova vrednost predstavlja granicu na osnovu koje se određuje koji roboti dolaze u obzir za uparivanje
- 3: izabrati proizvoljan događaj E_j kojem nije dodeljen robot i pokušati uparivanje sa prvim robotom R_i u okviru njegove liste preferenci
- 4: **if** R_i je već uparen sa E_k **then**
- 5: **if** $(B_j > B_k) \vee ((B_j = B_k) \wedge (w(i, j) < w(i, k))) \vee ((B_j = B_k) \wedge R_i \text{ je jedini kandidat za } E_j, \text{ ali ne i za } E_k)$ **then**
- 6: E_j se uparuje sa R_i
- 7: E_k postaje neuparen
- 8: **end if**
- 9: **end if**
- 10: **if** E_j je neuparen **then**
- 11: ponoviti istu proceduru za preostale robote, dok ne budu uspešno upareni (što je garantovano, pošto se vrednost granica usput povećava usled brisanja robota iz lista preferenci)
- 12: **end if**

U algoritmu 3 vrednost B_j koja se odnosi na događaj E_j naziva se granicom. Značenje granice je takvo da njena vrednost određuje da li robot koji je kandidat za uparivanje može biti uparen sa dotičnim događajem. Ukoliko ne može, biće pokušano uparivanje sa drugim robotom čije rastojanje od događaja je jednako granici. Pošto je jedan od ciljeva ujednačavanje potrošnje energije među robotima sprečavanjem uparivanja senzora sa robotima koji su previše udaljeni, podizanje granica se odlaže što je moguće duže. Inicijalno, broj kandidata u listama preferenci i vrednosti granica određuju se na osnovu parametra β .

U radu [WPT10] pokazano je da je složenost algoritma MM $O(m \cdot n \cdot \log n)$.

1.6.2 Komunikacioni protokoli za otkrivanje izvršilaca

Centralizovani algoritmi koordinacije robota kao neophodan uslov za svoje funkcionisanje pretpostavljaju posedovanje informacija o lokacijama svih događaja i svih robota. Ove informacije se razmenjuju komunikacijom u bežičnoj mreži [NS10]. Naivni pristup razmeni informacija zasnovan na preplavlivanju mreže je problematičan, jer nameće velike komunikacione troškove, koji se ogledaju u neprihvatljivo velikom broju poslatih poruka. Svaka nova informacija koja se prenosi kroz mrežu proizvodi $O(s)$ poruka, gde je s ukupan broj senzorskih čvorova. U ovom odeljku biće reči o sofisticiranijim komunikacionim protokolima koji postižu isti cilj na znatno efikasniji način. Pre razmatranja samih protokola, prvo će ukratko biti opisani mehanizmi prosleđivanja poruka na kojima se zasniva njihovo funkcionisanje.

1.6.2.1 Prosleđivanje poruka u BSM sa garantovanom isporukom

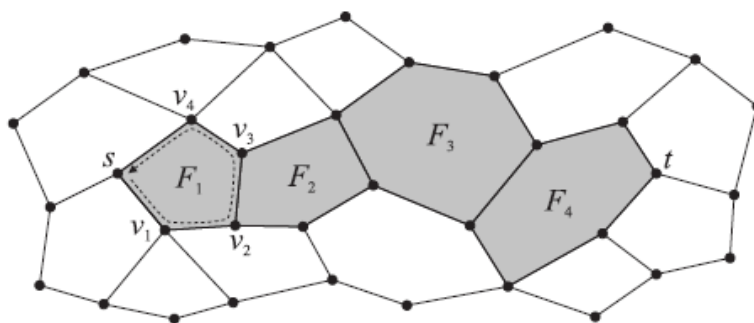
Algoritmi za prosleđivanje (rutiranje) poruka u *BSM* se generalno mogu podeliti u dve kategorije:

- **Pohlepni (engl. *greedy*) algoritmi**, koji koriste heuristike za konstruisanje putanje propagacije poruka zasnovane na izboru lokalnog optimuma u svakom koraku [LS03]. Primer pohlepne strategije rutiranja je algoritam koji u svakom koraku prosleđuje poruku onom susedu koji je najbliži odredišnom čvoru¹⁷. Principijelni nedostatak ovakvih algoritama ogleda se u tome što ne garantuju isporuku, u slučaju da postoje neregularnosti u topologiji mreže.
- **Algoritmi zasnovani na preplavlivanju**, koji koriste mehanizam kontrolisanog umnožavanja poruka, kako bi svaki čvor u mreži primio bar po jednu kopiju originalne poruke. Iako je u ovom slučaju isporuka garantovana, pod pretpostavkom da je mrežni graf povezan, problem je u velikim komunikacionim troškovima i lošoj skalabilnosti ovakvih rešenja [AE⁺09].

Bose et al. u radu [BMSU01] predstavili su algoritam za rutiranje koji garantuje isporuku poruke prijemnom čvoru uz pretpostavku da je komunikacioni graf povezan. Pri tom se ne zahteva globalno znanje, nego je dovoljno da predajni čvor raspolaže podatkom o lokaciji odredišta, kao i da svi čvorovi znaju sopstvenu poziciju i pozicije svojih 1-hop suseda. Algoritam se izvršava nad planarnim grafom, koji se konstruiše kao podstruktura mrežnog grafa. U planarnom grafu dozvoljeno je da se ivice ukrštaju samo u čvorovima [W⁺01]. Jedan od predstavnika klase planarnih grafova koji može

¹⁷Pojam najbližeg čvora može da se odnosi na minimalnu fizičku udaljenost, pri čemu se uzima u obzir Euklidsko rastojanje između čvorova. Međutim, ukoliko se čvorovima dodeljuju virtuelne koordinate zasnovane na udaljenostima u hopovima od unapred određenog skupa orjentira [WSRABD07], tada se kao metrika udaljenosti obično uzima Hamingovo rastojanje.

biti konstruisan na lokalizovan način je Gabrijelov graf [GS69]: proizvoljan čvor u zadržava sve i samo one izlazne ivice (u, v) koje ispunjavaju uslov da se unutar kruga sa prečnikom $|uv|$ koji prolazi kroz u i v ne nalazi nijedan drugi čvor. Ivice Gabrijelovog grafa vrše podelu ravni na nekoliko unutrašnjih i jedan spoljašnji poligon, koji se nazivaju licima grafa (engl. *face*). Osnovna ideja rutiranja baziranog na planarnom grafu je prosleđivanje poruke duž ivica sekvence lica koja se naslanjaju jedno na drugo i koja obezbeđuju progres ka odredišnom čvoru. U primeru prikazanom na slici 1.8, poruka se od izvora s prosleđuje ka odredištu t preko sekvence lica F_1, F_2, F_3, F_4 . Prosleđivanje poruka duž ivica lica obavlja se korišćenjem pravila leve (ili alternativno desne) ruke. Po prijemu poruke, čvor je prosleđuje dalje duž sledeće ivice planarnog grafa koja je prva u smeru suprotnom od kazaljke na satu u odnosu na ivicu duž koje je poruka primljena. Na primer, primenom ovog pravila prilikom obilaska lica F_1 na slici 1.8 dobija se ciklus $s - v_4 - v_3 - v_2 - v_1 - s$.

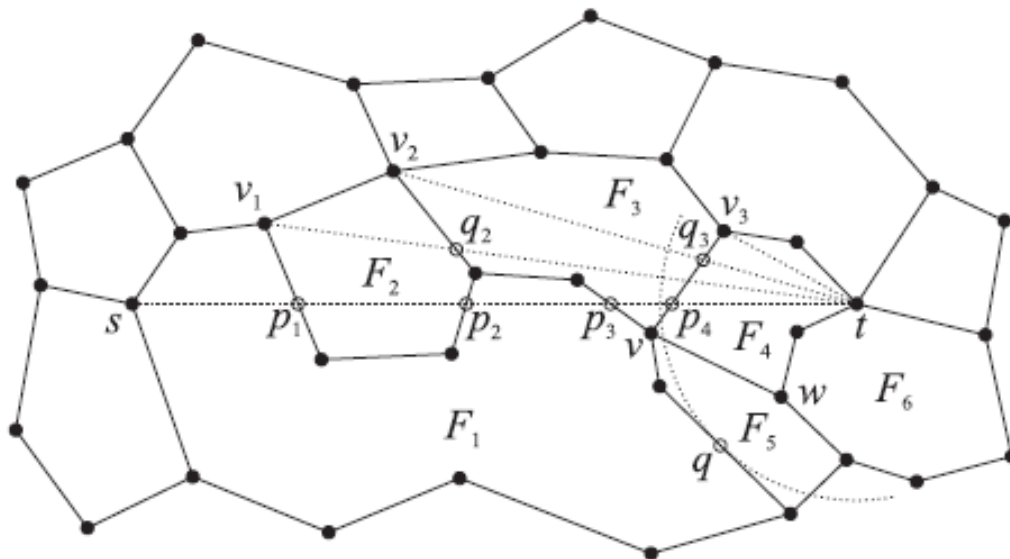


Slika 1.8: Lica u planarnom grafu duž čijih ivica se vrši rutiranje (preuzeto iz [FS06])

Rutiranje zasnovano na licima u planarnom grafu (engl. *face routing*) implementirano je u nekoliko varijanti koje se međusobno razlikuju po kriterijumu na osnovu kojeg se prilikom obilaska jednog lica grafa prelazi na sledeće. *GFG* šema (engl. *Greedy-Face-Greedy*), predložena od strane Freya i Stojmenovića [FS06], prati pravu liniju koja povezuje izvor i odredište (čvorovi s i t na slici 1.9). Kada se tokom propagacije poruke naiđe na ivicu koja preseca liniju st u tački p , prelazi se na lice koje se seče sa segmentom linije pt . Pri tome se u razmatranje uzimaju samo one presečne tačke koje su bliže odredištu od presečne tačke koja je izazvala prethodnu promenu lica.

U primeru prikazanom na slici 1.9 poruka koja potiče od čvora s prosleđuje se odredišnom čvoru t . Prilikom obilaska ivice lica F_1 u skladu sa pravilom leve ruke, poruka nailazi na presečnu tačku p_1 kao prvi presek sa linijom st , usled čega se prelazi sa lica F_1 na lice F_2 . Pri obilasku lica F_2 , p_2 će biti jedina presečna tačka koja je locirana bliže cilju od tačke p_1 , odakle je započeo obilazak lica F_2 . Po stizanju do tačke p_2 , ponovo se prelazi na lice F_1 . Obilazak lica F_1 traje do nailaska na presečnu tačku p_3 . Pošto je p_3 bliže cilju od p_2 , ispunjen je uslov za prelazak na lice F_3 . Istom logikom, nakon nailaska na tačku p_4 prelazi se na lice F_4 , čijim obilaskom se konačno stiže do odredišnog čvora t . Dakle, redosled obilazaka lica je F_1, F_2, F_1, F_3, F_4 . Još je potrebno napomenuti da su pri prelascima između dva uzastopna lica moguće dve

strategije nastavka obilaska. Prva je da se ne prelazi na drugu stranu linije st , nego da obilazak sledećeg lica kreće od čvora koji dolazi neposredno pre tačke ukrštanja i u tom slučaju se sledeće lice obilazi u istom smeru kao i prethodno. Druga opcija je da se u obilazik uključi i ivica koja se seče sa linijom st . U tom slučaju, obilazak sledećeg lica kreće neposredno po prelasku na drugu stranu, pri čemu se smer obilaska menja u odnosu na onaj koji je bio aktuelan prilikom obilaska prethodnog lica.

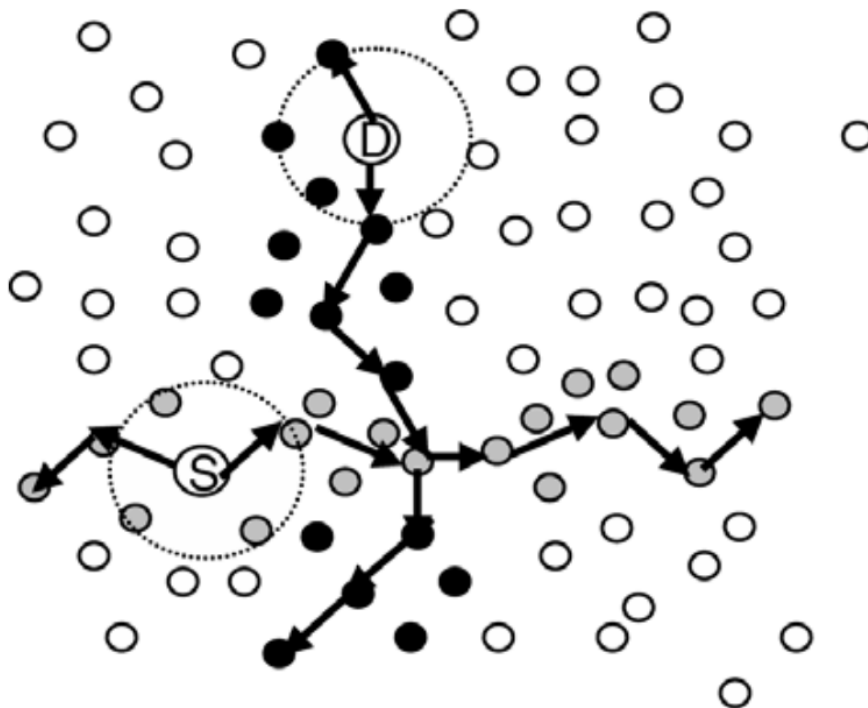


Slika 1.9: GFG rutiranje (preuzeto iz [FS06])

1.6.2.2 Pravougaoni kvorum

Pravougaoni kvorum koji su predložili Stojmenović, Liu i Jia [SLJ08] predstavlja komunikacionu strukturu koja funkcioniše na sledeći način. Kada je potrebno da robot (obebežen slovom "D" na slici 1.10) prosledi informaciju o svom prisustvu i poziciji, on to čini slanjem poruke koja se propagira duž "kolone" u mreži, prema severu i jugu, do dostizanja granica mreže. Svi čvorovi koji usput primaju ovu poruku čine kolonu koja se pruža pravcem sever-jug. Kada neki od čvorova (obebežen slovom "S" na slici) zahteva informaciju o robotu (ili o više njih), pretraga se vrši slanjem upita koji se propagira prema istoku i zapadu. Svi usputni čvorovi koji primaju upit sačinjavaju "vrstu" duž pravca istok-zapad. Povratna informacija se formira na mestu preseka vrste i kolone i šalje se nazad ka čvoru koji je inicirao upit. Na ovaj način je osigurano da sigurno dolazi do preseka vrsta sa kolonama, čime se garantuje otkrivanje željene informacije od strane zainteresovanih mrežnih entiteta.

Svaka od tri moguće operacije (oglašavanje, upit i ažuriranje lokacije) zahtevaju slanje $O(\sqrt{s})$ poruka, gde je s ukupan broj čvorova u mreži. Manjkavost protokola zasnovanog na kvorumu je to što operacije ažuriranja lokacije i pretrage moraju da se obavljaju po dužini (odnosno širini) cele mreže kako bi bilo osigurano da se vrste i kolone uvek

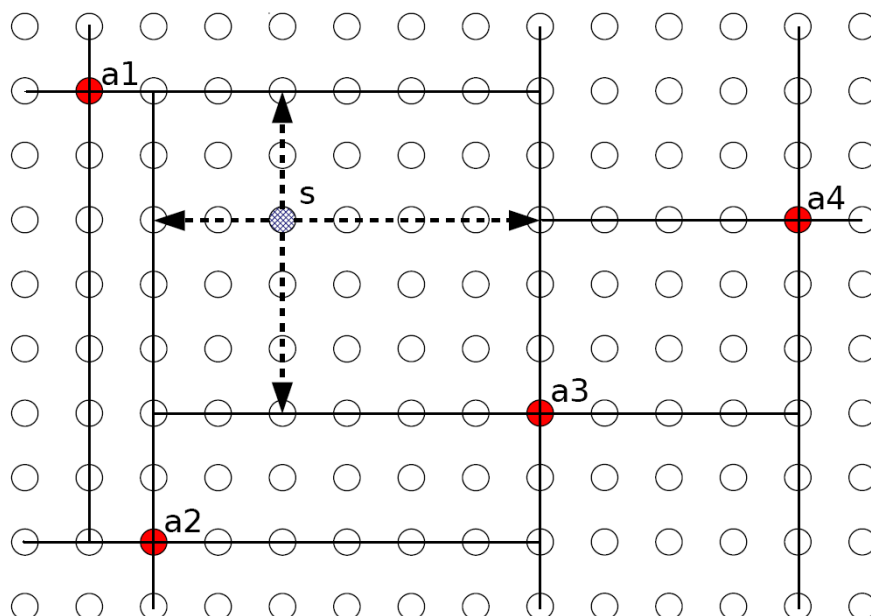


Slika 1.10: Formiranje kvoruma (preuzeto iz [SLJ08])

preseku. Uz to, dodatni problem je u tome što npr. ako su svi roboti kolinearni duž iste kolone, svi čvorovi u okviru kolone moraju da čuvaju informacije o svim robotima, što predstavlja opterećenje u pogledu ograničenih memorijskih resursa kojim ovi čvorovi raspolažu.

1.6.2.3 iMesh struktura

Strukturu nazvanu *iMesh* predložili su Li, Santoro i Stojmenović [LSS09]. Protokol kojim se formira informaciona struktura predstavlja modifikaciju *GCLP* protokola koji su Tchakarov i Vaidya predstavili u svom radu [TV04]. Polazeći od svakog robota, kreiraju se *registracione poruke* koje sadrže informacije o robotima i propagiraju se od strane senzorskih čvorova na četiri strane sveta (sever, jug, istok i zapad). Propagacija se zaustavlja u skladu sa *blokirajućim pravilom*, kada se naiđe na čvor koji je već primio informaciju o bližem robotu u vertikalnom ili horizontalnom pravcu. Primenom ovog pravila informaciona struktura dobija osobinu lokalnosti, a dodatna prednost je što je dovoljno da svaki čvor u okviru strukture memoriše informaciju o samo jednom (najbližem) robotu. Proizvoljan senzorski čvor (označen slovom "s" na slici 1.11), dolazi do informacije o obližnjem robotu sprovodeći tzv. *ukrštenu pretragu*. Proces pretrage se odvija slanjem upita u sva 4 geografska pravca, dok se ne naiđe na čvor u okviru *iMesh* strukture, pri čemu se generiše odgovor koji se šalje nazad čvoru koji je inicirao upit.

Slika 1.11: *iMesh* struktura sa 4 robota i senzorom koji vrši pretragu

Osnovna varijanta *iMesh* protokola analizirana je u radu [LSS09] za slučaj mreže sa rešetkastom strukturom, kao što je ona koja je prikazana na slici 1.11. Ovdje je topologija mreže pojednostavljena pretpostavkom da svaki čvor može da komunicira samo sa svoja četiri najbliža suseda (odnosno tri ili dva, ako je lociran na obodu mreže). Izložena je takođe ideja implementacije protokola u mreži sa proizvoljnom topologijom, gde se mehanizam formiranja informacione strukture zasniva na *GFG* rutiranju [BMSU01][FS06]. Unapređenje ovog protokola, kao jedan od doprinosa ove disertacije, prikazano je u poglavlju 2.

1.6.3 Koordinacija među robotima u slučaju višestrukih događaja

1.6.3.1 Centralizovana koordinacija izvršilaca u slučaju višestrukih događaja zasnovana na klasterima

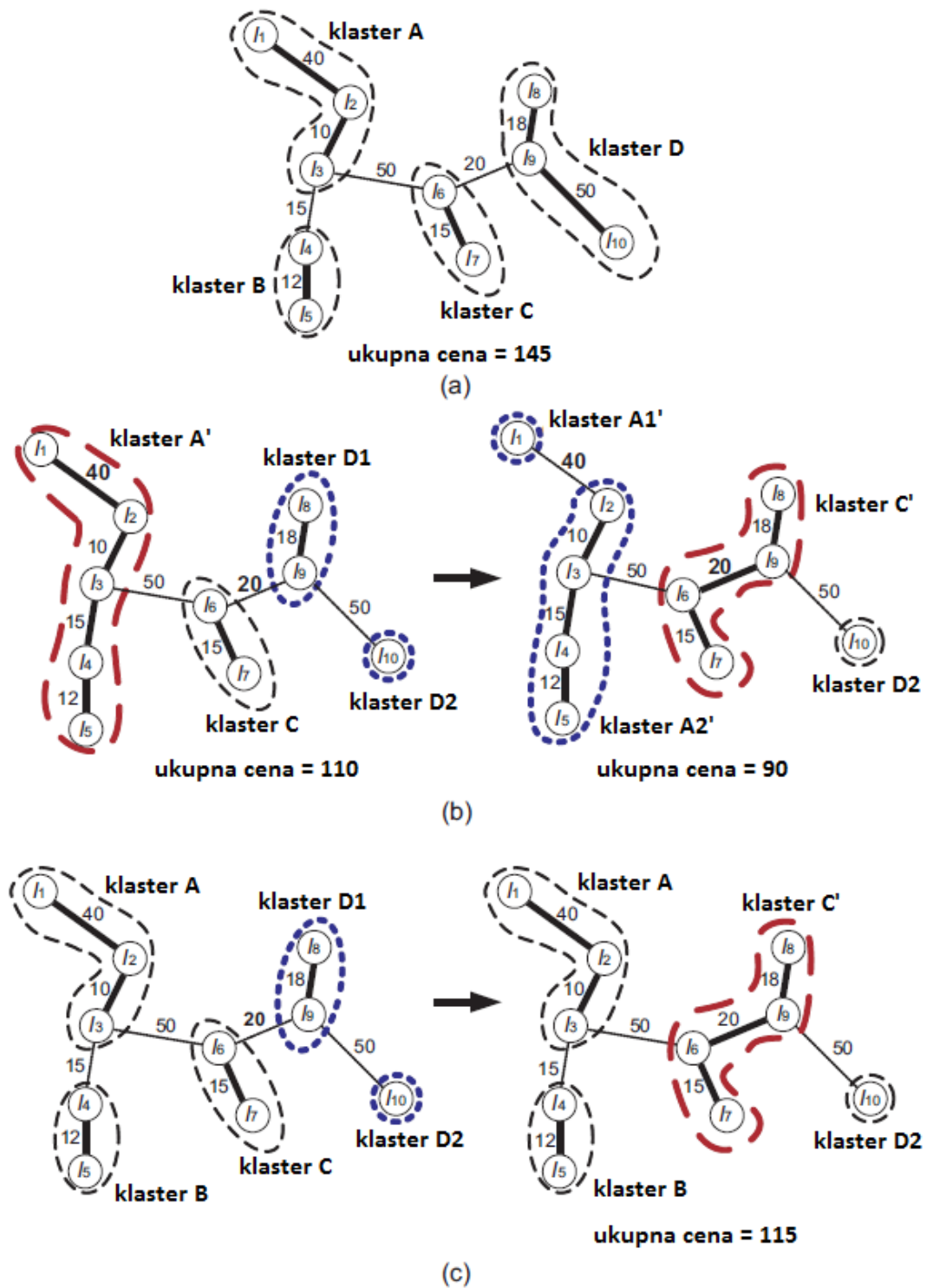
Wang et al. u radu [WPT10] predstavili su centralizovanu i distribuiranu verziju rešenja problema koordinacije robota kao izvršilaca u *BMSR*. U centralizovanom pristupu razlikuju se dva slučaja: $n = |R| < |E| = m$ i $|R| \geq |E|$. Kada je $|R| \geq |E|$, problem se rešava specijalizovanim algoritmom uparivanja (*MM*, engl. *Maximum Matching*) u težinskom bipartitnom grafu. Kada je pak $|R| < |E|$, skup događaja E deli se na $|R|$ klastera, primenom jedne od tri predložene metode:

- ***K-means* klasterovanje** [HKP06]: Ova šema grupiše mesta događaja u skladu sa njihovim relativnim udaljenostima, tako da su bliski događaji grupisani zajedno. Inicijalno se skup E deli na slučajan način na n nepraznih klastera. Zatim se

sprovodi iterativni proces. U svakoj iteraciji računaju se centralne tačke klastera. Ako je skup lokacija unutar klastera zadat njihovim koordinatama $\{(x_1, y_1), (x_2, y_2), \dots, (x_p, y_p)\}$, centralna tačka klastera se definiše kao $(\frac{1}{p} \sum_{i=1}^p x_i, \frac{1}{p} \sum_{i=1}^p y_i)$. Nakon što su izračunate centralne tačke, skup E se reparticioniše po principu da se lokacije koje su najbliže istoj centralnoj tački stavljaju u isti klaster. Proces se nastavlja sve dok je dalja relokacija događaja moguća. Primer *K-means* klasterovanja je prikazan na slici 1.12 (a).

- **MaxMin klasterovanje:** Kao manu *K-means* klasterovanja, autori rada [WPT10] navode neefikasnost u slučaju kada je raspodela mesta događaja neravnomerna ili retka. Stoga je predložena *MaxMin* šema, koja polazi od rezultata dobijenog *K-means* šemom, a zatim iterativno rastavlja i spaja pojedine klasterove. Intuitivno, klasteri sa raštrkanim lokacijama događaja trebaju da budu podeljeni. U svakoj iteraciji, prvo se konstruiše minimalno stablo razapinjanja [CLR⁺01] za svaki klaster. Neka je w_{max}^{intra} najveća dužina ivice, gledano među svim klasterima, a w_{min}^{inter} minimalna udaljenost između dva klastera, pri čemu se pod udaljenošću između klastera podrazumeva udaljenost između dva najbliža čvora koja pripadaju tim klasterima. Klaster koji sadrži ivicu dužine w_{max}^{intra} deli se na dva, uklanjanjem te ivice. Zatim se među novonastalom grupom od $n + 1$ klastera spajaju dva koja su međusobno udaljena w_{min}^{inter} . Postupak se ponavlja sve dok ne bude ispunjen uslov $w_{max}^{intra} \leq w_{min}^{inter}$. Primer *MaxMin* klasterovanja prikazan je na slici 1.12 (a) i (b).
- **balansirano klasterovanje:** Iako *MaxMin* klasterovanje smanjuje ukupnu cenu klastera, ono može dovesti do nebalansirane raspodele lokacija. Stoga je ideja da se pokuša da se istovremeno smanji ukupna cena klastera i standardna devijacija cena među različitim klasterima. Slično kao kod *MaxMin* klasterovanja, polazi se od rešenja dobijenog primenom *K-means* šeme, a zatim se klasteri iterativno rastavljaju i spajaju. U svakoj iteraciji se klaster c sa maksimalnom cenom deli na dva klastera, tako da je razlika u njihovim cenama minimalna. Tada se među $n + 1$ klastera spajaju dva, tako da cena klastera dobijenog njihovim spajanjem bude minimalna. Postupak se nastavlja dok je moguće postići smanjenje ukupne cene klastera. Primer *balansiranog* klasterovanja prikazan je na slici 1.12 (a) i (c).

Nakon formiranja klastera, roboti se uparuju sa njima, kako bi svaki robot posetio tačno jedan klaster. Uparivanje robota sa klasterima se vrši pomoću *MM* algoritma opisanog u odeljku 1.6.1.3. Algoritam uparivanja *MM* za slučaj $n \leq m$ koristi parametar β , koji uzima vrednost 4 ili 5. Nakon što su roboti upareni sa klasterima, redosledi obilazaka se dobijaju pomoću *TSP* heuristike.



Slika 1.12: Metode formiranja klastera (preuzeto iz [WPT10])

1.6.3.2 Pokrivanje grafa stablima sa korenjem u skupu R

U neusmerenom grafu $G = (V, D)$ sa pozitivnim težinama ivica w , *pokrivač grafa stablima* TC (engl. *tree cover*) je skup T stabala $\{T_i\}$, takav da je $V = \cup_{i=1}^k V(T_i)$. Težina stabla se definiše kao $w(T) = \sum_{e \in T} w(e)$. Cena pokrivanja T je $\max_{T_i \in T} w(T_i)$. Stabla u okviru skupa T mogu da imaju zajedničke čvorove, ili čak zajedničke ivice. Rešavanje problema $MMkTC$ (engl. *Min-Max k-Tree Cover*) ima za cilj pronalaženje pokrivača grafa stablima sa minimalnom cenom, uz ograničenje maksimalnog dozvoljenog broja stabala na k .

Neka $R \subset V$ označava skup korena. $RRTC$ (engl. *R-rooted tree cover*) grafa $G = (V, D)$ je pokrivač grafa stablima T , gde svako od stabala $T_i \in T$ ima jedinstven koren u skupu R . Stabla u $RRTC$ skupu T mogu da dele čvorove i ivice. Takođe, koren stabla T_i može da pripada stablu T_j , ali koreni stabala T_j i T_i moraju biti različiti. Dakle, ako su zadati težinski graf G i skup korena R , *min-max RRTC* problem podrazumeva nalaženje pokrivača grafa G stablima sa korenjem u skupu R , tako da pokrivanje ima minimalnu cenu. Even et al. u radu [EGK⁺04] pokazali su da je $RRTC$ NP-kompletan problem i prikazali polinomijalni aproksimativni algoritam sa faktorom aproksimacije $(4 + \epsilon)$ (algoritam 4).

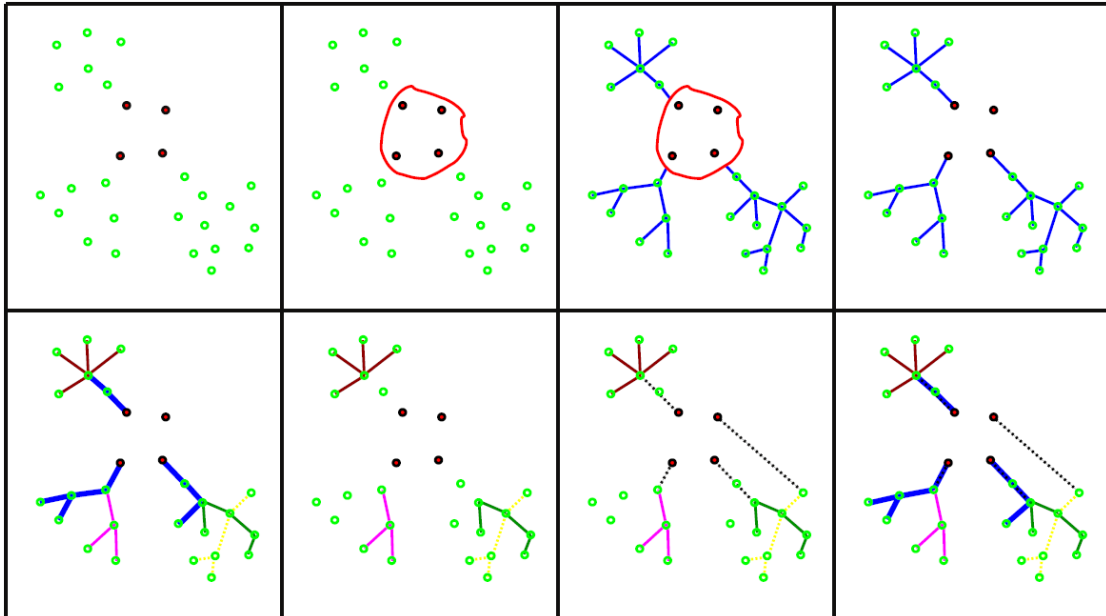
Algoritam 4 Algoritam $RRTC$ [EGK⁺04]

Ulaz: Graf G , skup korena R , parametar B

Izlaz: Pokriće grafa G stablima sa korenjem u skupu R , sa cenom ne većom od $4B$

- 1: Uklanjaju se sve ivice čija težina je veća od B .
 - 2: $M \rightarrow MSR$ dobijeno iz grafa G kontrakcijom svih korena iz skupa R u jedinstveni čvor.
 - 3: $\{T_i\}_i \rightarrow$ šuma dobijena iz M dekontrakcijom korena iz R .
 - 4: Vršiti se ivična dekompozicija svakog stabla T_i na stabla $\{S_j^i\}_j + L_i$ tako da $w(S_j) \in [B, 2B)$, za svako j i $w(L_i) < B$.
 - 5: Pokušati uparivanje stabala $\{S_j^i\}_{i,j}$ sa korenima, uz ograničenje da stablo S_i može biti upareno samo sa korenima udaljenim ne više od B od njega.
 - 6: Ako nije uspelo uparivanje, vraća negativan odgovor: "Vrednost B je suviše mala".
 - 7: Ako je svako stablo upareno sa korenom, vraća pozitivan odgovor: skup stabala, pri čemu se svako stablo sastoji od S_j , korena r sa kojim je upareno, i eventualno rezidualnim stablom L (ako postoji), koje sadrži koren r .
-

Uparivanje korena sa stablima u okviru pomenutog algoritma je instanca problema uparivanja čvorova u bipartitnom grafu sa minimalnom težinom i rešava se primenom Mađarskog algoritma, koji daje optimalno rešenje. Algoritam se kombinuje sa binarnim pretraživanjem, u cilju pronalaženja minimalne vrednosti parametra B , pri kojoj algoritam vraća pozitivan odgovor. U kontekstu problema *s-dodela(m,n)*, skup robota R predstavlja skup korena, a stabla koja se dobijaju kao izlaz algoritma predstavljaju klastere događaja koje roboti obilaze i čiji redosled obilaska se određuje primenom *TSP* heuristike.



Slika 1.13: Primer izvršenja RRTC algoritma (preuzeto iz [EGK⁺04])

Na slici 1.13 prikazane su različite faze izvršenja RRTC algoritma (s leva na desno):

1. Ulaz algoritma. Koreni su predstavljeni potpunim tačkama.
2. Kontrakcija korena.
3. Minimalno stablo razapinjanja kontrahovanog grafa.
4. De-kontrakcija grafa.
5. Ivična dekompozicija stabala. Rezidualna stabla $\{L_i\}$ su označena tamnim debelim ivicama.
6. Ne-rezidualna stabla $\{S_j^i\}$.
7. Uparivanje ne-rezidualnih podstabala sa korenima.
8. Konačna stabla se sastoje od rezidualnog stabla, stabla uparenog s korenom i ivice duž koje je izvršeno uparivanje.

1.6.3.3 Delimično distribuirani algoritmi koordinacije izvršilaca

Pojam *delimično distribuiranih* algoritama odnosi se na rešenja koja se izvršavaju distribuirano, iako su po svojoj suštini centralizovana. Kod takvih rešenja, roboti prvo komuniciraju među sobom, uz eventualnu asistenciju statičnih senzorskih čvorova, kako bi prikupili informacije o svim robotima i događajima. Na osnovu takvog globalnog znanja, oni izvršavaju isti algoritam nad istim ulaznim podacima i dolaze do

istog izlaza, oko kojeg postoji opšti konsenzus. Ovakav pristup odstupa od osnovne ideje distribuiranog donošenja odluka na osnovu lokalnog znanja i kao takav nije skalabilan, budući da troši značajno veće resurse sa povećanjem broja čvorova i dimenzija mreže.

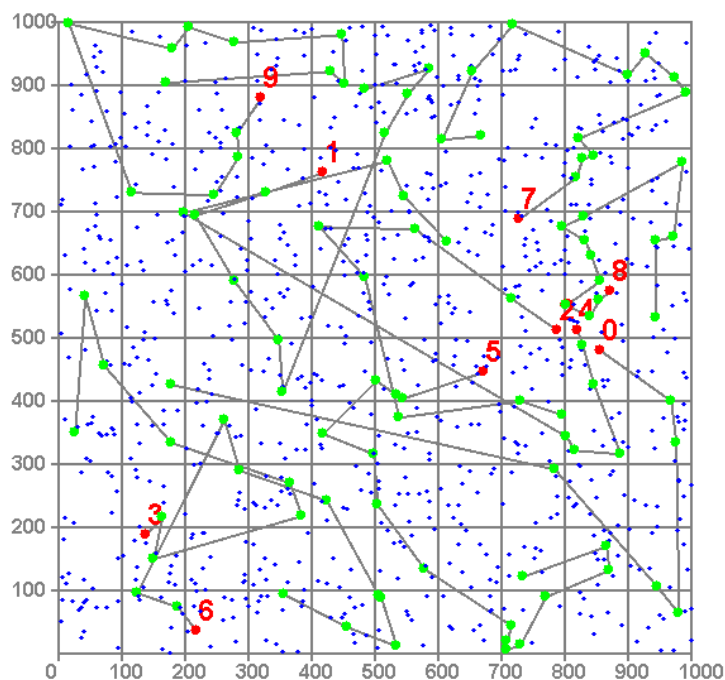
Centralizovani algoritam u radu [AGB10] proširen je tako da vodi ka delimično distribuiranom rešenju, koje će ovde biti označeno skraćenicom *AGB*. Ukoliko robotska mreža čini povezan graf, tada se centralizovani algoritam izvršava na način opisan u odeljku 1.6.1.2. U suprotnom, međusobno povezane komponente učestvuju u kreiranju podmreža. Izolovani događaji koji ne pripadaju ni jednoj od ovih podmreža oglašavaju svoje prisustvo duž kolona u senzorskoj mreži, dok roboti oglašavaju svoje prisustvo duž vrsta¹⁸. Na taj način, čvorovi koji se nalaze na preseku vrsta i kolona mogu da dodele izolovane događaje nekoj od podmreža. Uparivanje događaja sa robotima se dalje vrši nezavisno u svakoj podmreži. Nakon toga, u podmrežama u kojima broj robota i događaja nije bio jednak, pojedini roboti ili događaji ostaju neupareni. Svi neupareni događaji i roboti oglašavaju svoje prisustvo duž vrsta/kolona, pri čemu pored prosleđivanja informacije o sebi, takođe primaju povratnu informaciju o drugim događajima/robotima. Nakon ovog koraka, svaki od robota ima isto znanje, pa se donose identične odluke izvršenjem *SB* algoritma [GS62] od strane svih robota.

U svom distribuiranom rešenju (koje će ovde biti skraćeno označeno kao *WPT* algoritam) [WPT10], Wang, Peng i Tseng su podelili mrežu u pravougaonu rešetku. U svakoj ćeliji rešetke bira se po jedan senzorski čvor koji preuzima ulogu koordinatora (engl. *grid head*). Algoritam se odvija u tri faze:

- **Pripremna faza:** Tokom ove faze svaki koordinator prikuplja informacije o prisustvu robota i događaja unutar svoje ćelije. Zatim se prisustvo robota oglašava duž kolona rešetke (komunikaciona struktura čini pravougaoni kvorum), kako bi koordinatori ćelija koje sadrže događaje mogli da se informišu o njima šaljući upite ortogonalno, tj. duž vrsta rešetke.
- **Faza takmičenja:** Koordinatori ćelija koje sadrže događaje nadmeću se za robote slanjem poziva duž vrsta rešetke. Po prijemu (višestrukih) poziva, roboti biraju "pobedničke" ćelije koje uvrštavaju u svoje liste za obilazak i obaveštavaju ih o tome, dok ostalima šalju poruku da ih odbijaju. Način na koji se biraju pobedničke ćelije je po svojoj prirodi sličan centralizovanom algoritmu uparivanja *MM*, pomenutom u odeljku 1.6.3.1. Nakon konačnog broja iteracija, roboti imaju formirane liste ćelija za obilazak, pri čemu ne može da se desi da neka od ćelija sa događajima ostane izostavljena, a algoritam kojim se ćelije sa događajima dodeljuju robotima je dizajniran tako da se svakom od robota dodeljuje približno jednak broj ćelija za obilazak, bez obzira na to koliko je događaja prisutno u pojedinim ćelijama.

¹⁸Ovakav mehanizam komunikacije se naziva *kvorum* i detaljnije je opisan u odeljku 1.6.2.2.

- **Izvršna faza:** U ovoj fazi roboti sekvencijalno obilaze mesta događaja u okviru ćelija koje su im dodeljene tokom prethodne faze, a redosled obilazaka se određuje dvostepenom *TSP* šemom. Primer robotskih putanja u slučaju 10 robota i 100 događaja prikazan je na slici 1.14.

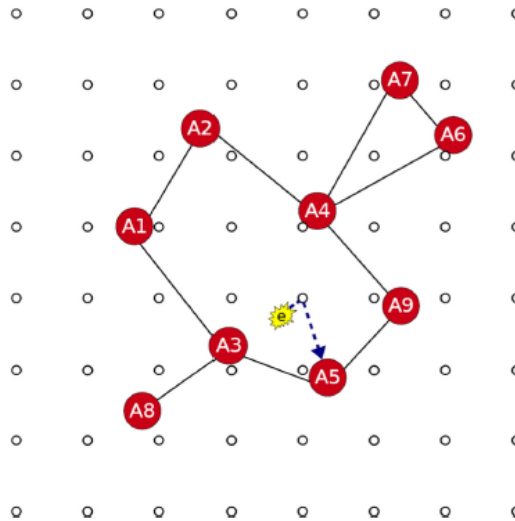


Slika 1.14: Putanje robota pri izvršenju distribuirane verzije WPT algoritma

Pored suboptimalne dodele zadataka, oba delimično distribuirana algoritma (*AGB* [AGB10] i *WPT* [WPT10]) imaju kao manu neefikasnost u komunikacionom smislu, budući da zahtevaju razmenu velikog broja poruka u okviru senzorske mreže.

1.6.3.4 Aukcijska agregacija u slučaju pojedinačnog događaja

U radu [MLMS13] analiziran je problem koordinacije robota u cilju izbora najpogodnijeg (tj. najbližeg) robota kao izvršioca koji opslužuje događaj koji je prijavljen nekom od robota, u svrhu eventualnog pronalaženja robota koji je bliži mestu događaja među njegovim susedima u robotskoj mreži. Koordinacija robota je realizovana primenom aukcijskog agregacionog protokola nazvanog *k-SAAP* (engl. *k-hop Simple Auction Aggregation Protocol*, slika 1.15). U ovom protokolu, kao mehanizam otkrivanja obližnjeg robota od strane senzorskog čvora koji je detektovao događaj koristi se *iMesh* struktura. Robot kojem je prijavljen događaj organizuje aukciju unutar robotske mreže, pri čemu se informacija o događaju prosleđuje susedima udaljenim do k hopova, formirajući usput strukturu stabla. Roboti na i -tom

Slika 1.15: k -SAAP protokol (preuzeto iz [MLMS13])

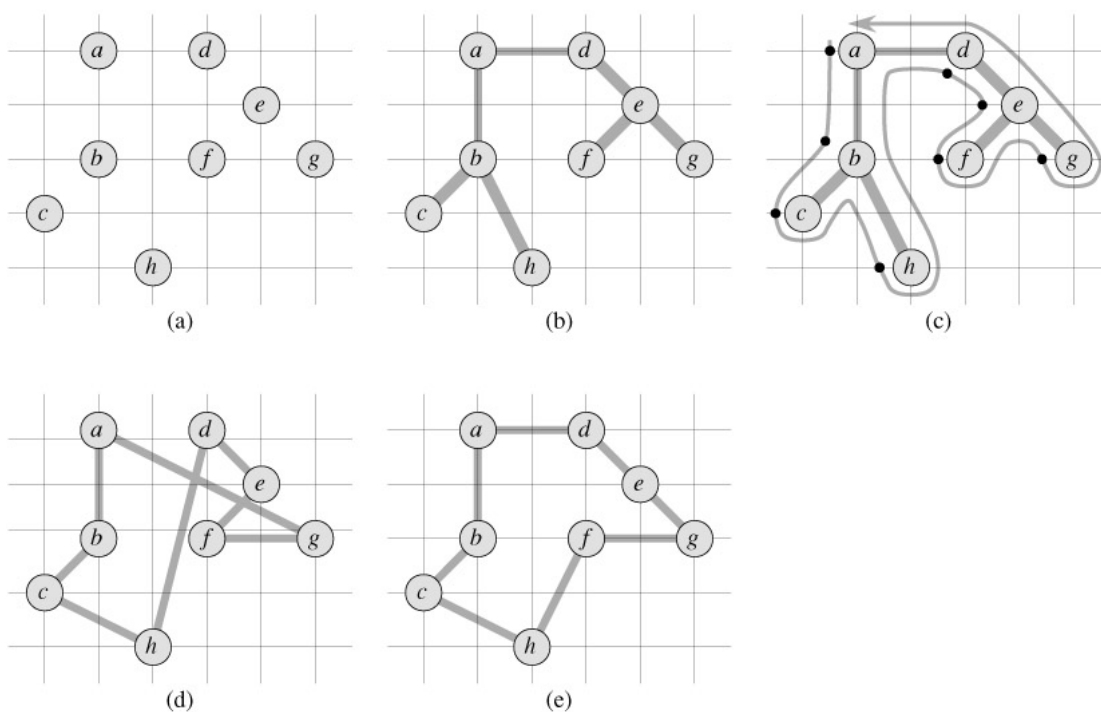
nivou stabla primaju odgovore od svojih direktnih potomaka na $i + 1$ -om nivou. Na osnovu znanja sopstvene pozicije i na osnovu primljenih poruka od strane potomaka, svaki od robota na i -tom nivou vrši agregaciju podataka tako što svom direktnom pretku na $i - 1$ -om nivou šalje samo po jednu poruku koja sadrži informaciju o robotu u njegovom podstablu koji je najbliži mestu događaja. Na kraju, robot koji je inicirao aukciju (tj. koren stabla) raspolaže informacijom o tome koji je robot u njegovom susedstvu do k -hopova najbliži događaju i na osnovu toga dodeljuje tom robotu zadatak obilaska događaja.

1.6.4 Problem trgovačkog putnika

Problem trgovačkog putnika *TSP* (engl. *Traveling Salesman Problem*) jedan je od najpoznatijih i najintenzivnije proučavanih problema iz oblasti kombinatorne optimizacije. U svojoj osnovnoj varijanti, problem je formulisan na sledeći način: ako je zadat skup od n gradova i poznate su udaljenosti između bilo koja dva grada, potrebno je naći najkraću moguću putanju (*TSP* ciklus, koji ujedno predstavlja Hamiltonov ciklus minimalne dužine), koja povezuje sve gradove i vraća se u mesto polaska, koje je jedan od n zadatih gradova. U teoriji složenosti algoritama, *TSP* je poznat kao NP-težak problem [Pap03], dok varijanta ovog problema u kojoj se zahteva odgovor na pitanje da li postoji *TSP* ciklus koji je kraći od zadate dužine spada u klasu NP-kompletnih problema.

Kada je u pitanju problematika koja je obrađivana u ovoj disertaciji, od značaja je varijanta *TSP* u kojoj su gradovi predstavljeni tačkama u Euklidskoj ravni. U potpunom simetričnom neusmerenom težinskom grafu $G = (V, (V \times V))$ koji predstavlja ulaznu instancu, skup čvorova V reprezentuje gradove, a težine ivica

odgovaraju udaljenostima između njih. U pitanju je specijalna podklasa *TSP* problema u kojoj važi nejednakost trougla, odnosno $(\forall u, v, w \in V)(|uv| + |vw| > |uw|)$. Za *TSP* sa nejednakošću trougla razvijeno je mnoštvo sub-optimalnih algoritama (heuristika) i aproksimativnih rešenja¹⁹. Ukoliko se konstruiše minimalno stablo razapinjanja T grafa G i ukoliko je $W = \frac{1}{2} \sum_{(u,v) \in T} |uv|$ težina minimalnog stabla razapinjanja, lako se pokazuje da W predstavlja donju, a $2W$ gornju granicu za dužinu optimalnog *TSP* ciklusa²⁰. Slika 1.16 ilustruje 2-aproksimativni algoritam koji formira *TSP* ciklus na osnovu obilaska minimalnog stabla razapinjanja u dubinu: (a) skup čvorova (gradova), (b) minimalno stablo razapinjanja, (c) redosled obilaska čvorova sa polaznom tačkom "a", (d) dobijeni *TSP* ciklus, (e) optimalno rešenje, koje je za 23% kraće od dobijenog aproksimativnim algoritmom.



Slika 1.16: 2-aproksimativni *TSP* algoritam (preuzeto iz [CLR⁺01])

Dodatna pretpostavka koja čini razliku u odnosu na klasični *TSP* problem je da se ne zahteva povratak u početnu tačku. U pogledu složenosti, radi se o problemu koji je i dalje NP-težak. Heuristika koja kao izlaz daje acikličnu *TSP* putanju sa zadatom početnom tačkom (što je u kontekstu problema koordinacije robota početna pozicija robota tokom pojedinačne runde), zasnovana je na iterativnim poboljšanjima rešenja dobijenog gore opisanim 2-aproksimativnim algoritmom. Poboljšanja se postižu tako što se u svakoj iteraciji pronalaze parovi ivica koje se uklanjaju iz ciklusa i zamenjuju

¹⁹Aproksimativni algoritmi koji se izvršavaju u polinomijalnom vremenu funkcionišu samo pod pretpostavkom da važi nejednakost trougla. U suprotnom, ne postoje aproksimativni algoritmi sa konstantnim faktorom aproksimacije, osim ako je $P=NP$.

²⁰Dok je W "tvrda" donja granica, $2W$ predstavlja konzervativnu procenu gornje granice. U radu [Chr76] prikazan je $O(n^3)$ aproksimativni algoritam sa faktorom aproksimacije $\frac{3}{2}$, što znači da u najgorem slučaju pronalazi *TSP* ciklus koji nije duži od $\frac{3}{2}W$.

se novim ivicama koje ponovo povezuju segmente ciklusa, tako da ukupna dužina rezultujućeg ciklusa bude kraća od prvobitnog²¹. Na primer, ako se krene od ciklusa pikazanog na slici 1.16 (d), uklanjanjem ivica (h, d) i (g, a) dobijaju se međusobno nepovezani segmenti putanje $a - b - c - h$ i $d - e - f - g$. Ubacivanjem ivica (h, g) i (d, a) dobija se ciklus $a - b - c - h - g - f - e - d - a$, koji je kraći od prvobitnog ciklusa $a - b - c - h - d - e - f - g - a$. Iterativni postupak se nastavlja sve dok postoje parovi ivica kod kojih opisani pospupak razmene dovodi do skraćanja ukupne dužine ciklusa. Na kraju, suboptimalna aciklična TSP putanja se dobija uklanjanjem iz ciklusa duže od dve ivice incidentne sa početnom tačkom.

1.7 Doprinosi i pregled disertacije

Sva istraživanja vezana za ovu doktorsku disertaciju finansirana su od strane Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije kroz projekat "Inovativne elektronske komponente i sistemi bazirani na neorganskim i organskim tehnologijama ugrađeni u robe i proizvode široke potrošnje", (evidencioni broj TR32016, 2011-2015), kao i od strane Evropske komisije kroz projekat APOSTILLE "Reinforcement of research potentials of the Faculty of Technical Sciences in the field of post silicon electronics", (EC FP7 REGPOT 256615, 2010-2013).

Glavni doprinosi ove disertacije objavljeni su u dva međunarodna časopisa sa impakt faktorom, u jednom poglavlju monografije međunarodnog značaja i prezentovana su na dve međunarodne konferencije:

[LM15] *M. Lukic and I. Mezei, "Localised querying and location update service in wireless sensor and robot networks with arbitrary topology", International Journal of Ad Hoc and Ubiquitous Computing, 2015, (in press). (kategorija M23)*

[LBS15] *M. Lukic, A. Barnawi, and I. Stojmenovic, "Robot coordination for energy-balanced matching and sequence dispatch of robots to events", Computers, IEEE Transactions on, vol. 64, no. 5, pp. 1416-1428, May 2015. (kategorija M21)*

[MLM14] *I. Mezei, M. Lukic, and V. Malbasa, "Wireless Sensor and Robot Networks", World Scientific, 2014, ch. 3: "Robot-Robot Coordination", pp. 51-69. (kategorija M14)*

²¹Ista ideja izmene ivica među parovima čvorova primenjena je u *PDM* algoritmu opisanom u poglavlju 3.

[LM12] M. Lukic and I. Mezei, "Distributed distance sensitive iMesh based service discovery in dense WSN", *Ad-hoc, Mobile, and Wireless Networks*, pp. 435-448, 2012. **(kategorija M33)**

[LS13] M. Lukic and I. Stojmenovic, "Energy-balanced matching and sequence dispatch of robots to events: Pairwise exchanges and sensor assisted robot coordination", in *Mobile Ad-Hoc and Sensor Systems (MASS), 2013 IEEE 10th International Conference on. IEEE, 2013*, pp. 249-253. **(kategorija M33)**

U nastavku su takođe navedeni radovi iz oblasti *BSM* koji su indirektno vezani za tematiku obrađivanu u ovoj disertaciji, a koji su takođe asocirani gore navedenim projektima:

[JLM11] N. Janicijevic, M. Lukic, and I. Mezei, "Routing protocol for low-power and lossy wireless sensor networks", in *Telecommunications Forum (TELFOR), 2011 19th. IEEE, 2011*, pp. 234-237. **(kategorija M33)**

[LPMS09] M. Lukic, B. Pavkovic, N. Mitton, and I. Stojmenovic, "Greedy geographic routing algorithms in real environment", in *Mobile Ad-hoc and Sensor Networks, 2009. MSN'09. 5th International Conference on. IEEE, 2009*, pp. 86-93. **(kategorija M33)**

[LJM11] M. Lukic, N. Janicijevic, and I. Mezei, "Improved decision making in WSN based on localized auctions and fuzzy logic," in *Telecommunications Forum (TELFOR), 2011 19th. IEEE, 2011*, pp. 238-241. **(kategorija M33)**

[MLMS13] I. Mezei, M. Lukic, V. Malbasa, and I. Stojmenovic, "Auctions and iMesh based task assignment in wireless sensor and actuator networks", *Computer Communications*, vol. 36, no. 9, pp. 979-987, 2013. **(kategorija M22)**

Sledi kratak pregled najvažnijih doprinosa disertacije. U poglavlju 2 opisana su sledeća nova rešenja vezana za komunikacione protokole za otkrivanje izvršilaca:

- *iMeshC* i *iMeshD*, kao unapređenja postojećeg *iMesh* protokola (doprinos predstavljen u radu [LM12]).
- Novi *vCell* protokol (doprinos predstavljen u radovima [LM15] i [MLM14]).
- Novi algoritmi za ažuriranje lokacije u okviru *iMesh* i *vCell* protokola (doprinos predstavljen u radu [LM15]).

U poglavlju 3 izloženi su doprinosi disertacije koji su predstavljeni u radovima [LS13], [LBS15] i [MLM14]. U pitanju su novi algoritmi za centralizovanu i distribuiranu dodelu zadataka izvršiocima, pri čemu roboti preuzimaju ulogu izvršilaca zadataka pod kojima se podrazumeva obilazak mesta događaja:

- Algoritmi za centralizovano uparivanje robota s događajima (*PDM* i *PRM*).
- Algoritmi za centralizovanu dodelu sekvenci događaja izvršiocima (*SQD* i *SRD*).
- Algoritam za centralizovanu dodelu višestrukih izvršilaca događajima (*GSQD*).
- Algoritam za centralizovanu dodelu sekvenci događaja izvršiocima sa vremenskim ograničenjima (*SQDT*).
- Algoritmi za distribuiranu dodelu sekvenci događaja izvršiocima (*MAD* i *MRD*).
- Algoritam za distribuiranu dodelu višestrukih izvršilaca događajima (*GAD*).

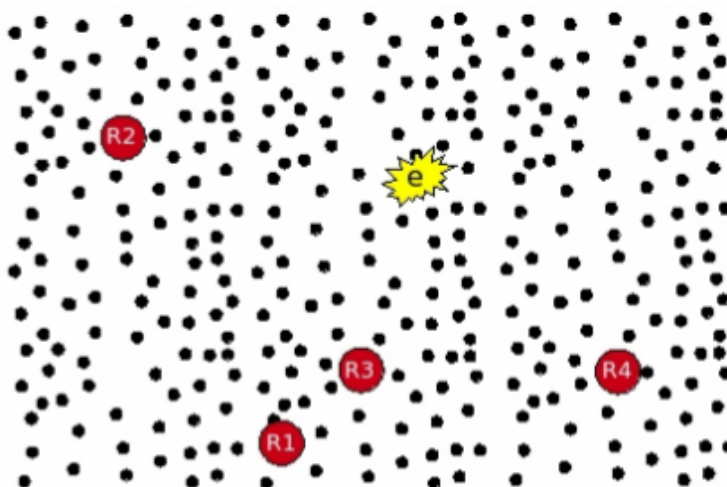
Konačno, u poglavlju 4 prikazani su i komentarisani rezultati simulacija kojima su potvrđena poboljšanja performansi novih u odnosu na postojeća rešenja.

Poglavlje 2

Komunikacioni protokoli za otkrivanje izvršilaca

U ovom poglavlju izložen je doprinos disertacije kada su u pitanju komunikacioni protokoli za otkrivanje izvršilaca u bežičnim mrežama senzora i robota. Podsećanja radi, sledi kratak osvrt na postavku problema. Posmatra se teren sa senzorskim čvorovima postavljenim na slučajan način usled nemogućnosti da budu raspoređeni u skladu sa unapred određenom strategijom, na primer postavljeni su izbacivanjem iz aviona, ili je topologija mreže diktirana konfiguracijom terena. Svi senzorski čvorovi raspolazu informacijom o sopstvenoj poziciji, korišćenjem *GPS*-a ili nekog drugog odgovarajućeg lokalizacionog sistema. Roboti su takođe na slučajan način raspoređeni po terenu i njihov broj je značajno, dva do tri reda veličine manji u odnosu na broj statičkih senzorskih čvrova. Po detektovanju događaja pomoću senzora, potrebno je proslediti informaciju o njegovoj pojavi nekom od robota, koji će potom preuzeti ulogu izvršioca zaduženog za obilazak mesta događaja. Senzorski čvor koji detektuje događaj inicira proces otkrivanja izvršilaca, sa krajnjim ciljem lociranja i porosleđivanja zadatka najboljem, odnosno najbližem raspoloživom izvršiocu. Komunikacija na relaciji senzorski čvor-senzorski čvor i senzorski čvor-robot odvija se korišćenjem jedinstvenog bežičnog kanala. Mreža je modelovana kao jedinični disk graf (*JDG*), što znači da svaki mrežni entitet, bilo da je u pitanju senzorski čvor ili robot, može da komunicira sa svim i samo onim entitetima koji se nalaze unutar dometa emitovanja. Svi mrežni entiteti su vremenski sinhronizovani, čime se omogućava iterativni postupak kreiranja informacione strukture za otkrivanje izvršilaca. Ovaj postupak iniciraju roboti istovremenim slanjem registracionih poruka, čija se propagacija kroz mrežu nadalje odvija u pravilnim vremenskim intervalima. Primer *BMSR* sa jednim događajem i četiri robota prikazan je na slici [2.1](#)

Problem otkrivanja izvršilaca u *BMSR* posmatra se u datom vremenskom okviru kada se roboti ne pomeraju. Iako je mobilnost esencijalno svojstvo robota u takvim sistemima, postavlja se zahtev da se proces otkrivanja izvršilaca obavlja na statički



Slika 2.1: BMSR sa pojedinačnim događajem i 4 robota (preuzeto iz [MMS10])

način. Time se podrazumeva da kretanje robota ne spada u ovaj proces, nego se smatra delom akcija koje se sprovode u okviru izvršenja zadatka. U ovom poglavlju, roboti će biti posmatrani kao statički mrežni entiteti sa istim komunikacionim svojstvima kao i statički senzorski čvorovi.

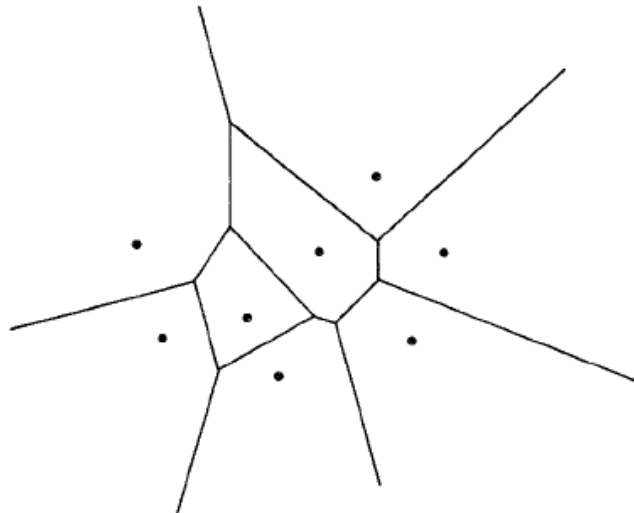
2.1 Karakteristike protokola za otkrivanje izvršilaca

Ako se govori o efikasnosti protokola za otkrivanje izvršilaca, postavljaju se sledeća ključna pitanja:

- Kako pronaći najpogodnijeg izvršioca kojem treba da bude dodeljen zadatak obilaska mesta događaja? Neophodno je osigurati pronalaženje najbližeg robota u cilju minimizacije kretanja, što sa sobom povlači smanjenje potrošnje energije, kao i vremena reakcije sistema. Prema tome, od protokola za otkrivanje izvršilaca se očekuje da omogući pouzdanu detekciju najbližeg, ili eventualno obližnjeg robota.
- Kako minimizovati količinu mrežnog saobraćaja izazvanog procesom otkrivanja izvršilaca, bez ugrožavanja kvaliteta samog procesa? Ovo je bitno u cilju produženja vremena funkcionisanja baterijski napajanih senzorskih čvorova, kao i smanjenja mogućnosti kolizija pri emitovanju.
- Kako na adekvatan način reagovati na promene u mrežnoj strukturi izazvane kretanjem robota, tako da informaciona struktura za otkrivanje izvršilaca bude ponovo uspostavljena, uz nametanje što je moguće manjih dodatnih komunikacionih troškova?

Sva ova pitanja mogu biti postavljena kao aspekti istog krajnjeg cilja, a to je produženje životnog vremena sistema unapređenjem energetske efikasnosti koje se postiže minimizacijom kretanja robota i broja poruka koje se šalju posredstvom bežičnog medijuma.

Pošto se u cilju skraćivanja putanja koje prelaze roboti najboljim izvršiocima smatraju oni roboti koji su najbliži mestima događaja, to se odražava i na svojstva koja bi trebalo da poseduje protokol za njihovo otkrivanje. Rešenje koje poseduje poželjnu osobinu osetljivosti na razdaljinu (engl. *proximity property*), implicitno podrazumeva konstrukciju Voronoi dijagrama [Aur91]. Voronoi dijagram, koji se još naziva i Voronoi teselacijom, dobija se particionisanjem ravni na oblasti (ćelije), na osnovu udaljenosti tačaka ravni od tačaka iz posebnog diskretnog podskupa. Tačke iz tog podskupa nazivaju se semenima (engl. *seed*), ili generatorima. Voronoi ćelija za zadati generator je skup svih tačaka ravni koje su bliže tom generatoru nego bilo kom drugom, kao što je ilustrovano slikom 2.2.



Slika 2.2: *Primer Voronoi dijagrama sa 8 tačaka*

U kontekstu problema otkrivanja izvršilaca, kao generatori Voronoi dijagrama uzimaju se pozicije robota u Euklidskoj ravni. Roboti distribuiraju informacije o sebi duž granica Voronoi poligona (ćelija). Ovo omogućava senzorskom čvoru koji je detektovao događaj da sprovede pretragu duž proizvoljnog pravca i da time sigurno dođe do informacije o najbližem robotu, pošto će upit svojom propagacijom pre ili kasnije dosegnuti granicu Voronoi ćelije. Iako ova ideja dovodi do optimalnog rešenja, problem je što zahteva globalno znanje. U praksi, potrebno je zameniti Voronoi dijagram lokalizovanom planarnom strukturom sa svojstvom osetljivosti na razdaljinu, čime se proces otkrivanja lokalizuje, odnosno ograničava na neposredno okruženje senzorskog čvora koji zahteva informaciju o robotu.

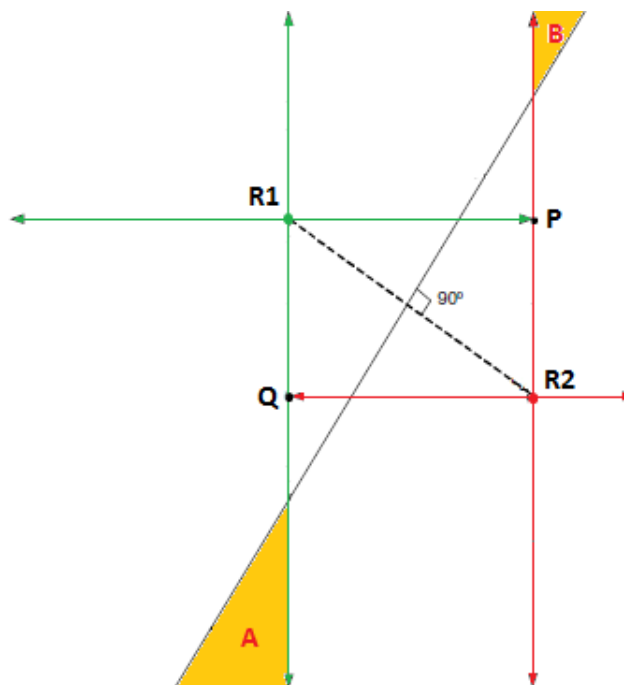
2.1.1 Analiza nedostataka postojećih rešenja

Jedna od ideja usmerena ka distribuiranom rešavanju problema otkrivanja izvršilaca podrazumeva zamenu Voronoi dijagrama informacionim direktorijumom, poznatim kao pravougaoni kvorum [SLJ08]. Detalji vezani za funkcionisanje kvoruma izloženi su u odeljku 1.6.2.2. Ukratko, roboti vrše propagaciju informacije o svom prisustvu duž vrsta (odnosno prema istoku i zapadu), dok se pretraga vrši slanjem upita duž kolona (pravcem sever-jug). Pravac propagacije upita ukršta se sa vrstama koje sadrže informacije o robotima. Kada upit dosegne ivicu terena, informacija prati povratnu putanju i vraća se ka senzorskom čvoru koji je inicirao upit. Principijelni problem vezan za ovakav metod otkrivanja izvršilaca, očituje se u mogućnosti pojave neravnomernog opterećenja među senzorskim čvorovima u pogledu memorijskih resursa, u slučaju da su roboti postavljeni kolinearno sa vrstom. Drugi problem je u tome što ovakva struktura ne poseduje svojstvo osetljivosti na razdaljinu, što posledično dovodi do toga da proces pretrage ne može biti lokalizovan.

Modifikacija pravougaonog kvoruma nazvana *iMesh* [LSS09] opisana je u odeljku 1.6.2.3. Protokol kojim se kreira *iMesh* struktura ujedno predstavlja generalizaciju *GCLP* protokola [TV04]. U pitanju je planarna struktura koja se konstruiše na lokalizovan način, zadržavajući pri tome poželjnu osobinu osetljivosti na daljinu. Prva modifikacija je u tome da roboti propagiraju informaciju o lokaciji slanjem registracionih poruka duž četiri geografska pravca. Druga modifikacija je korišćenje pravila blokiranja baziranog na razdaljini, čime se postiže ključna prednost u odnosu na kvorum, kako u pogledu osetljivosti strukture na razdaljinu, tako i po pitanju smanjenja komunikacionih troškova neophodnih za uspostavljanje informacione strukture. Pravilo blokiranja primenjuje se onda kada se dve *iMesh* ivice sretnu u istom čvoru. U tom slučaju, informacija koja potiče od bližeg robota nastavlja propagaciju, dok ona koja potiče od udaljenijeg robota biva blokirana. Rad [LSS09] sadrži teorijsku analizu *iMesh* strukture i pokazuje da takva struktura poseduje osobinu lokalnosti i dovodi do ravomerne raspodele memorijskog opterećenja među čvorovima. Teorijska analiza i simulacije koje potvrđuju teorijske nalaze u tom radu sprovedeni su pod pretpostavkom da su senzorski čvorovi postavljeni u pravougaonu rešetkastu strukturu, pri čemu je domet emitovanja takav da svaki čvor može da komunicira samo sa svojim neposrednim susedima u četiri geografska pravca i stoga ima dva, tri ili četiri suseda, u zavisnosti od svog položaja u rešetki. Po uspostavljanju informacione strukture, čvor koji je detektovao događaj sprovodi proces otkrivanja izvršioca putem ukrštene pretrage. Upiti se šalju u 4 geografska pravca i propagiraju se dok ne dostignu *iMesh* ivice, kada se generišu odgovori koji se šalju nazad prateći putanje upita.

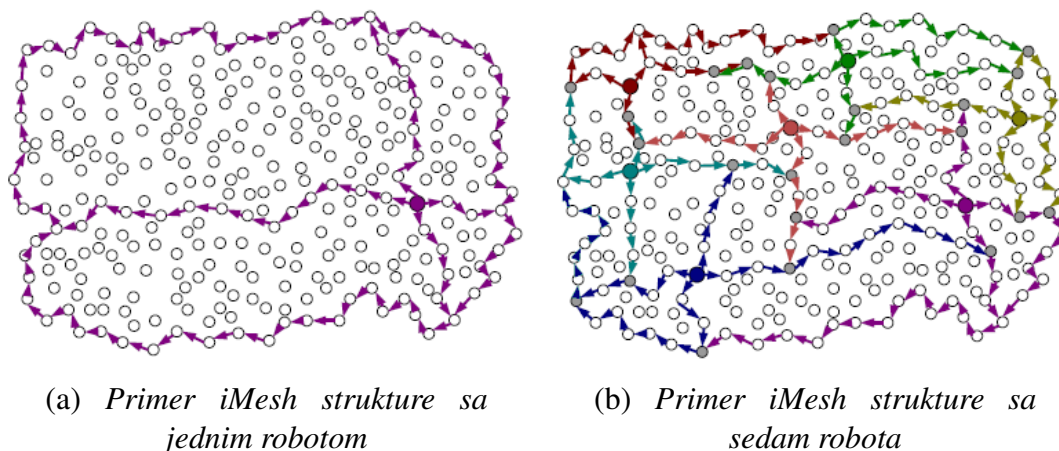
I pored toga što *iMesh* struktura poseduje poželjnu osobinu osetljivosti na razdaljinu, postoje izvesni slučajevi kada ovaj mehanizam nije u stanju da detektuje najbližeg izvršioca. U primeru prikazanom na slici 2.3 prikazana je šema *iMesh* strukture u slučaju postojanja dva robota, R_1 i R_2 . Simetrala duži koja povezuje R_1 i R_2 deli ravan

na dva dela koji sadrže tačke kojima je najbliži izvršilac R_1 , odnosno R_2 . Prilikom formiranja informacione strukture, pravilo blokiranja je primenjeno u čvorovima P i Q . Čvor P blokira informaciju o robotu R_2 , dok se informacija o robotu R_1 prosleđuje dalje ka jugu. U takvoj situaciji, ako neki od čvorova koji se nalazi u zoni A inicira pretragu, dobiće informaciju samo o robotu R_1 , iako je robot R_2 bliži. Slično razmatranje važi i za zonu B . U osnovnoj verziji protokola nazvanoj *iMeshA* primenjuje se samo pravilo blokiranja, dok je u verziji *iMeshB* uvedeno dodatno pravilo produženja, čime je gore navedeni nedostatak delimično kompenzovan. U slučaju kada postoji mogućnost komunikacije u robotskoj mreži, posredstvom bežičnog linka većeg dometa, zahtevi koji se postavljaju pred protokol za otkrivanje izvršilaca su u izvesnoj meri relaksirani. Tada nije neophodno naći najbližeg, nego je dovoljno locirati obližnjeg izvršioca, a eventualna korekcija se postiže aukcijskim protokolom kao što je *k-SAAP* [MLMS13], koji se izvršava u okviru robotske mreže.

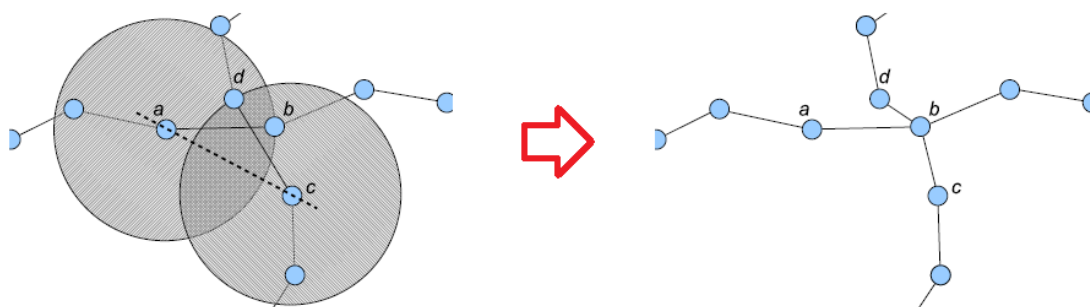


Slika 2.3: Slučajevi kada *iMesh* nije u stanju da otkrije najbližeg izvršioca

U slučaju kada su senzorski čvorovi raspoređeni na slučajan način, u radu [LSS09] se sugeriše da propagacione putanje poruka tokom uspostavljanja *iMesh* strukture slede *GFG* šemu prosleđivanja (rutiranja) poruka [BMSU01][FS06]. Konkretno, po prijemu registracione poruke, čvor iz nje izdvaja i memoriše informaciju o položaju robota koji je inicirao njenu propagaciju, a zatim koristeći informaciju o pravcu prostiranja, prosleđuje poruku sledećem čvoru. Kada registraciona poruka dosegne praznu oblast, koja može biti praznina u topologiji ili ivica terena, mehanizam prosleđivanja obezbeđuje praćenje ivice (engl. *face routing*), u pravcu kazaljke na satu (ili obratno), čime se obilazi prazan prostor.

Slika 2.4: *iMesh* struktura sa *GFG* rutiranjem (preuzeto iz [LSS07])

Ukoliko je u mreži prisutan samo jedan robot, usled nedostatka informacije o tome gde se nalaze granice mreže, kao i prirode *GFG* rutiranja, putanje registracionih poruka se nakon dolaska do najudaljenijih čvorova u sva četiri pravca nastavljaju duž ivica mreže, kao što je prikazano na slici 2.4a. U slučaju prisustva više robota, putanje njihovih registracionih poruka se međusobno presecaju unutar mreže i/ili se preklapaju na ivicama mreže. Dve registracione putanje koje se presecaju mogu da se nađu u situaciji da se susreću u istom čvoru, ili da se ukrštaju. U radu u kojem je preliminarno predstavljen *iMesh* protokol [LSS07], sugerise se da je slučaj kada se putanje ukrštaju moguće lokalno razrešiti transformacijom u slučaj kada se susreću u istom čvoru (slika 2.5), ali su izostavljeni detalji mehanizma pomoću kojeg bi se ova transformacija mogla realizovati. Dalje, potrebno je istaći da navedeno razmatranje važi samo u slučaju kada se primenjuje pohlepno rutiranje. U slučaju primene *GFG* šeme ne može doći do mimoilaženja registracionih putanja, budući da se rutiranje sprovodi nad planarnim grafom, čije ključno svojstvo je da se ivice susreću isključivo u čvorovima grafa.

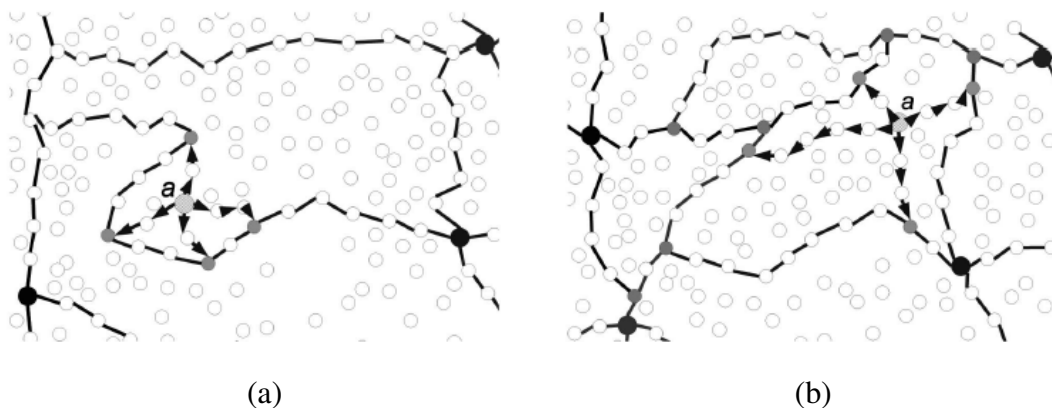


Slika 2.5: Ukrštanje putanja registracionih poruka (preuzeto iz [LSS07])

Dakle, pod pretpostavkom da se putanje uvek susreću u zajedničkom čvoru, u takvom čvoru se primenjuje pravilo blokiranja i pravilo produženja, na sličan način kao u mrežama sa rešetkastom strukturom. Na slici 2.4b prikazana je rezultujuća *iMesh*

struktura sa sedam robota. Očigledan nedostatak *GFG* šeme rutiranja ogleda se u komunikacionim troškovima vezanim za obilazak ivica terena. Principijelno, ovaj problem se javlja usled toga što *GFG* nije u stanju da razlikuje praznine u mrežnoj topologiji od ivica terena. Međutim, postoje još i suptilniji detalji koji se ne uočavaju tako lako na prvi pogled. U pitanju su situacije koje mogu da se jave tokom procesa ukrštene pretrage, pri čemu protokol ne obezbeđuje tačnu informaciju o poziciji najbližeg izvršioca.

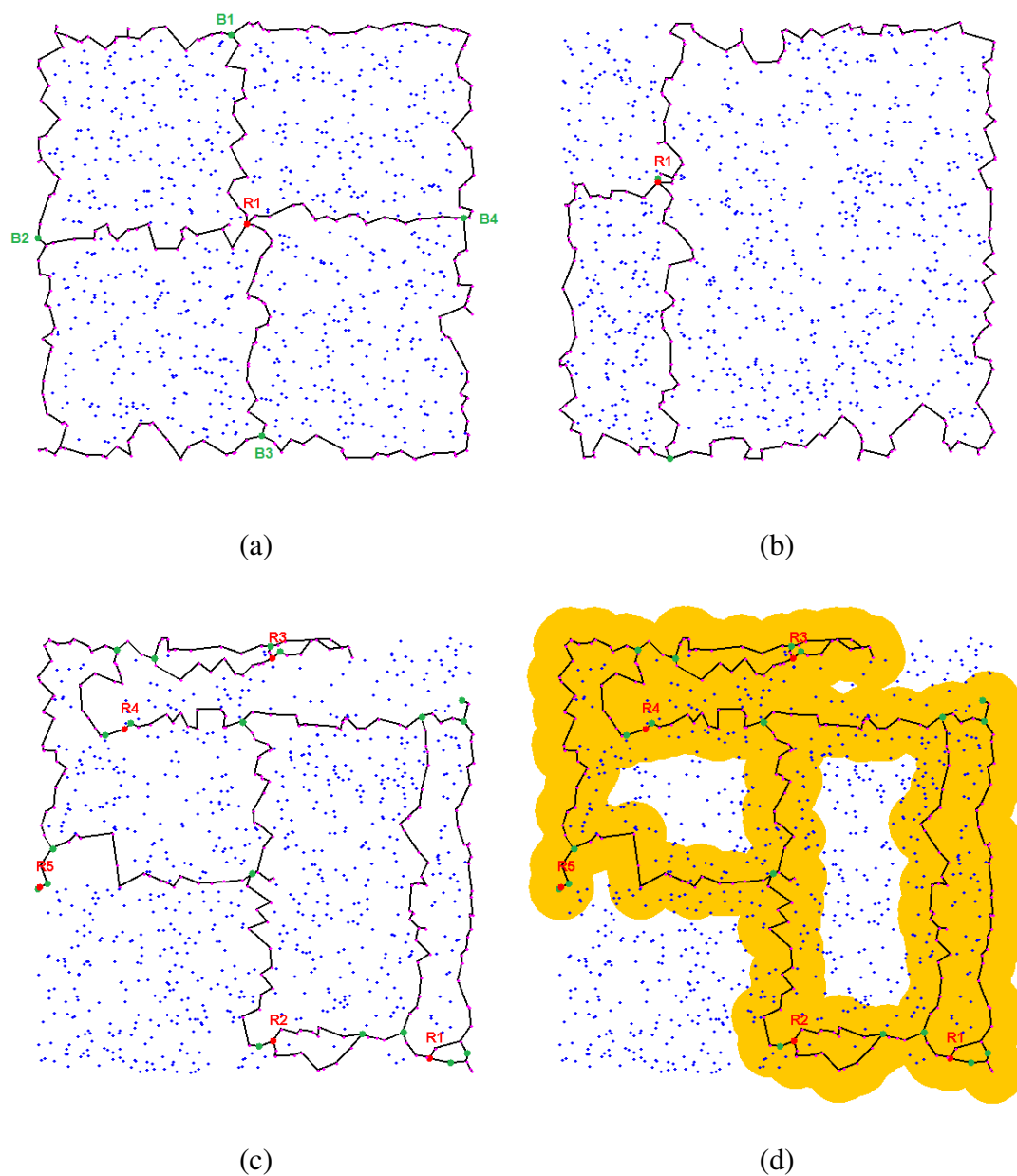
U prisustvu praznina u mrežnoj topologiji, poruke se prosleđuju duž njihovih ivica, što može rezultovati izlomljenim putanjama propagacije i posledičnim greškama u procesu pretrage. Slika 2.6, na kojoj linije sa strelicama pokazuju linije duž kojih se vrši pretraga, ilustruje dva primera kada se ukrštenom pretragom ne otkriva najbliži robot. U prvom slučaju (slika 2.6a), sve četiri putanje poruka za petragu dostižu jednu istu zakrivljenu ivicu. Na slici 2.6b prikazana je situacija kada je *iMesh* ćelija¹ unutar koje se nalazi čvor koji vrši pretragu ograničena sa pet ivica, a putanje pretrage su takve da ne dostižu najseverniju ivicu ćelije.



Slika 2.6: Slučajevi u kojima ukrštena pretraga ne otkriva najbližeg izvršioca (preuzeto iz [LSS07])

Do sada navedeni nedostaci osnovnog protokola nazvanog *iMeshA* uočeni su i istaknuti od strane autora rada [LSS07]. Međutim, nakon implementacije u simulacionom okruženju, uočeno je još problema, koji su ilustrovani situacijama prikazanim na slici 2.7. Slika 2.7a prikazuje očekivanu i željenu situaciju, kada je u mreži prisutan jedan robot. Informacija o njemu se propagira u sva četiri geografska pravca, prateći *GFG* šemu rutiranja bez prelaska pravca prostiranja (što je objašnjeno u odeljku 1.6.2.1). Po dolasku do ivice mreže, propagacione poruke obilaze spoljno lice, što znači da u nastavku prate ivice mreže u smeru suprotnom od kazaljke na satu. Propagacija se zaustavlja u čvorovima B_1, B_2, B_3 i B_4 , gde se primenjuje pravilo blokiranja. Ovde je bitno istaći sledeći detalj: pošto se prilikom *GFG* rutiranja dešava da putanja prelazi dva puta preko iste ivice, potrebno je osigurati da propagaciona putanja ne može da

¹Pod pojmom *iMesh* ćelije podrazumeva se oblast ograničena *iMesh* ivicama, odnosno putanjama registracionih poruka.

Slika 2.7: Ilustracija nedostataka algoritma *iMeshA*

blokira samu sebe. To se postiže tako što se pravilo blokiranja primenjuje ili po dolasku do čvora koji pripada *iMesh* putanji drugog robota, ili u ovom slučaju, po dolasku do čvora koji pripada *iMesh* putanji istog robota, ali je propagirana duž drugog geografskog pravca. Na primer, severna putanja propagacije dolazi do ivice terena u čvoru B_1 , dalje se propagira duž ivice terena suprotno od kazaljke na satu i konačno biva blokirana u čvoru B_2 , koji već pripada zapadnoj putanji.

Međutim, redukovanjem mrežnog grafa sa *JDG* na Gabrijelov graf, drastično se smanjuje broj neposrednih suseda čvorova u mreži, što može dovesti do pojave anomalija u *iMesh* strukturi. Jedan takav primer prikazan je na slici 2.7b, gde severni i

istočni pravac odmah bivaju izblokirani, tako da se propagacija dalje odvija samo duž zapadnog i južnog pravca. Teoretski, u ekstremno nepovoljnom slučaju kada je u mreži prisutan jedan robot koji ima samo jednog suseda u Gabrijelovom grafu, propagacija registracionih poruka vrši se samo u jednom pravcu. U tom slučaju javlja se problem nemogućnosti primene pravila blokiranja, što dovodi do kruženja propagacione poruke u beskonačnoj petlji.

Na slici 2.7c prikazan je primer *iMeshA* strukture sa pet robota. Ovde su uočljivi još neki fenomeni koji dodatno doprinose da struktura odstupa od željene, kakva se uspostavlja u mrežama gde su čvorovi raspoređeni u pravilnu pravougaonu rešetku. U ovom slučaju nijedan od pet robota koji su prisutni u mreži nije u mogućnosti da vrši propagaciju registracionih poruka u sva četiri pravca. Na primer, robot R_4 propagira informaciju u dva pravca (sever i istok), pri čemu je severna putanja izrazito deformisana. Robot R_3 propagira u tri pravca (južni je izblokirani), zapadna i severna putanja su približno paralelne usled blizine robota severnoj granici terena, a istočna putanja stiže do lokalnog ekstrema, nakon čega se okreće i vraća nazad prema R_3 , pošto propagaciona putanja ne sme da preseče pravu liniju koja se pruža od robota ka istoku. Robot R_5 ima još nepovoljniji položaj, budući da je u mogućnosti da vrši propagaciju samo ka istoku. Na početku ove putanje uočljiva je velika krivina, koja nastaje usled neravnomernosti distribucije čvorova, odnosno lokalne praznine u topologiji grafa. Na slici 2.7d prikazana je ista mreža, sa obeležnim jediničnim diskovima pri svakoj transmisiji, koji sačinjavaju osenčene oblasti. Uočava se da bi krivina u propagacionoj putanji mogla biti ispravljena, ukoliko bi se vršilo pohlepno rutiranje, koje koristi dugačke ivice, umesto mnoštva kratkih ivica od kojih je sačinjen Gabrijelov graf.

2.2 Modifikovani iMesh protokol u gustim mrežama sa uniformnom raspodelom čvorova

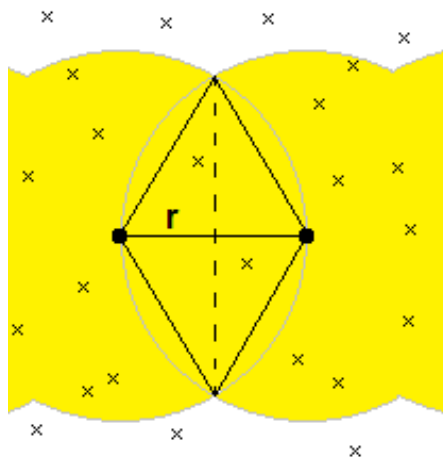
U radu [LM12] prikazali smo neke od načina na koje se osnovni *iMesh* protokol može adaptirati kako bi bio prilagođen primenama u gustim mrežama sa uniformnom slučajnom distribucijom senzorskih čvorova. Za razmatranja koja slede u nastavku ovog poglavlja, biće dovoljna intuitivna predstava pojma guste mreže, a to je mreža u kojoj se sa dovoljno velikom verovatnoćom može smatrati da unutrašnji čvorovi imaju susede u sva četiri geografska pravca. Diskusija koja je izneta u poglavlju 4 daje detaljniji uvid u uslove pod kojima se mreža može smatrati gustom.

U mrežama sa rešetkastom topologijom je relativno jednostavno sprovesti postupak kreiranja informacione strukture i ukrštene pretrage, budući da su sve propagacione putanje međusobno normalne ili paralelne. Što je još važnije, nema mimoilaženja, nego se putanje koje su međusobno normalne ukrštaju u čvorovima. Ako je raspodela

čvorova slučajna, ukrštanje putanja u čvorovima najčešće ne mora biti slučaj, te je stoga neophodno obezbediti mehanizam konstrukcije koji osigurava da se pravilo blokiranja i ukrštena pretraga mogu sprovesti na odgovarajući način.

Ukoliko ne postoje prazne oblasti u mrežnoj topologiji, odnosno ako je raspodela čvorova uniformna, problemi vezani za *GFG* rutiranje mogu da budu prevaziđeni primenom geografskog pohlepnog rutiranja. Na ovaj način smanjuju se komunikacioni troškovi, iz više razloga. Kao prvo, propagacija registracionih poruka prekida se po dostizanju ivica terena. Zatim, prilikom rutiranja obično se koriste duže ivice u odnosu na one koje se sačinjavaju Gabrijelov graf. Uz to, obebeđuje se da ćelije budu približno pravougaonog oblika, čime se poboljšava kvalitet procesa pretrage izbegavajući nepoželjne situacije prikazane na slici 2.6.

Druga modifikacija ogleda se u tome da se tokom propagacije registracionih poruka umesto *iMesh* ivica kreiraju *iMesh* pojasevi tako što svi čvorovi koji su 1-hop susedi (odnosno koji su unutar dometa emitovanja) čvorova koji vrše (re)transmisiju registracionih poruka, memorišu informaciju o robotu koji je inicirao proces registracije, bez obira na to da li treba da proslede poruku dalje ili ne. Na ovaj način se "tanke" ivice zamenjuju "debelim" oblastima, bez nametanja dodatnih komunikacionih troškova. Time se osigurava da međusobno normalne propagacione putanje ne mogu da se mimođu, što je ilustrovano slikom 2.8. U najgorem slučaju, ako su dva uzastopna čvora koja prosleđuju propagacionu poruku maksimalno udaljena (udaljenost im je jednaka dometu emitovanja r), tada širina pojasa na najužem mestu iznosi $r\sqrt{3}$.



Slika 2.8: Minimalna širina *iMesh* pojasa

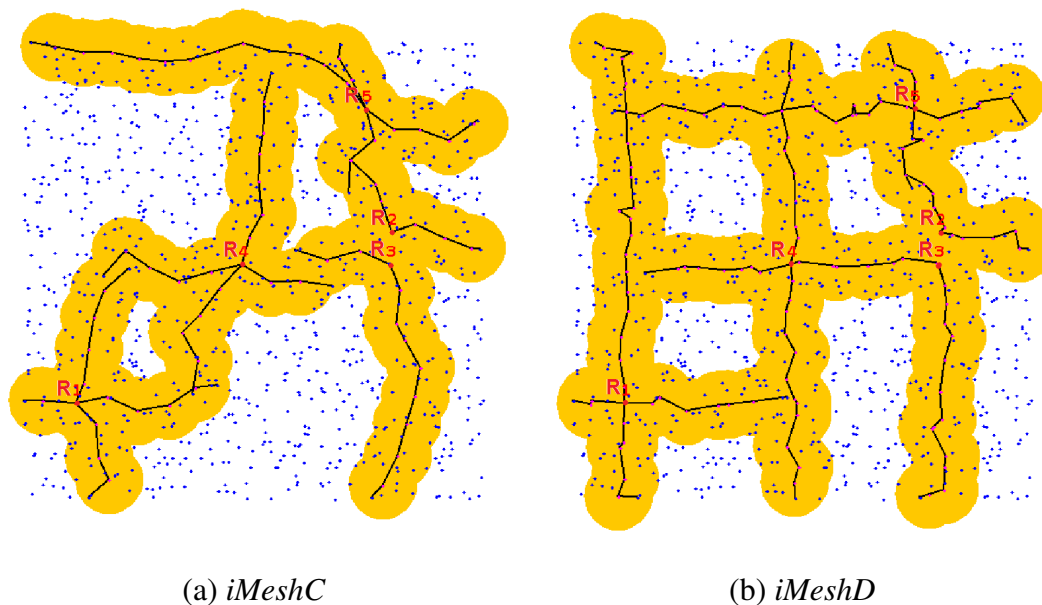
Pravilo blokiranja modifikovano je tako da se primenjuje kada registraciona poruka biva prosleđena bilo kom čvoru koji se već nalazi unutar nekog od postojećih *iMesh* pojaseva, čime se rešava problem koji nastaje ako se putanje registracionih poruka ne ukrštaju u istom čvoru. Isti princip primenjuje se i kod ukrštene pretrage, kada upit dođe do čvora u okviru *iMesh* pojasa.

2.2.1 Notacija

Mrežu sačinjavaju skup statičkih senzorskih čvorova $S = \{S_1, S_2, \dots, S_m\}$ i skup robota $R = \{R_1, R_2, \dots, R_n\}$. Svi uređaji opremljeni su lokalizacionim sistemom i stoga raspoložu informacijom o svojim koordinatama (x, y) u okviru dvodimenzionalnog terena koji se nadzire. Koristi se jedinstveni bežični komunikacioni kanal, a mreža je modelovana kao jedinični disk graf (*JDG*). Mrežni graf $G = (V, E)$, gde je $V = S \cup R$, u sebe uključuje statičke senzorske čvorove i robote. Između dva čvora u i v postoji ivica e ako i samo ako je Euklidsko rastojanje između njih manje ili jednako dometu emitovanja r (tj. $(e = (u, v) \in E) \Leftrightarrow (|uv| \leq r)$). Skup 1-hop suseda čvora u označava se sa $N(u)$, odnosno $(\forall u \in V)(\forall v \in V)(v \in N(u) \Leftrightarrow (u, v) \in E)$.

2.2.2 Formiranje modifikovane iMesh strukture

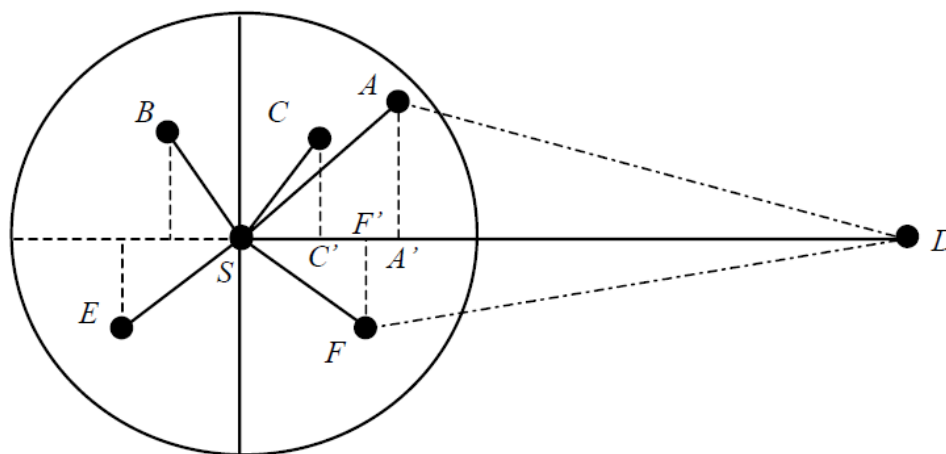
Proces otkrivanja izvršilaca odvija se u dve faze. U prvoj fazi kreira se informaciona struktura. Za početak, neophodno je da svaki čvor u mreži pošalje pozdravnu poruku, koja sadrži informaciju o njegovoj poziciji. Po prijemu pozdravnih poruka, čvorovi formiraju tabele susednosti koje sadrže liste njihovih neposrednih suseda. Slika 2.9 ilustruje kreiranje *iMesh* strukture u mreži sa 5 robota i 1000 senzorskih čvorova. Svaki od robota kreira maksimalno četiri registracione poruke, u zavisnosti od njihovog relativnog položaja. Na primer, roboti R_2 i R_3 su neposredni susedi (R_2 je severoistočno u odnosu na R_3), pa stoga robot R_2 generiše samo dve registracione poruke koje se propagiraju ka severu i istoku, dok robot R_3 šalje poruke u pravcu juga i



Slika 2.9: *iMesh* struktura

zapada. Registraciona poruka se uvek prosleđuje od pošiljaoca u ka odabranom prijemnom čvoru $v \in N(u)$, u skladu sa unapred određenim kriterijumom rutiranja. Crne linije na slici ilustruju propagacione putanje poruka. Kao što je već napomenuto, prilikom prosleđivanja registracione poruke od čvora u ka čvoru v , svi neposredni susedi čvora u ($w \in N(u)$) takođe primaju istu poruku, koja će biti prosleđena dalje samo od strane čvora v . Memorisanjem informacije o robotu koja je izdvojena iz primljene poruke, ovi čvorovi postaju deo *iMesh* informacione strukture, te se na taj način umesto tankih ivica kreiraju informacioni pojasevi, bez dodatnih komunikacionih troškova. Informacioni pojasevi označeni su žutom bojom na slici 2.9.

Slika 2.10 ilustruje različite strategije rutiranja od čvora S ka čvoru D . Poruka se uvek prosleđuje nekom od suseda koji obezbeđuje pozitivan napredak prema odredištu, što znači smanjenje ukupnog rastojanja do odredišta. Prema tome, u primeru prikazanom na slici 2.10, kandidati za prosleđivanje poruke su čvorovi A , C i F . Kada je u pitanju rutiranje registracionih poruka, odredište je virtuelni čvor u beskonačnosti, pa se kao kriterijum napretka posmatra rastojanje između predajnog čvora i projekcije prijemnog čvora na liniju SD . Rutiranje pri kojem se uvek bira čvor koji obezbeđuje najveći napredak (u primeru sa slike u pitanju je čvor A), obeležava se skraćenicom *MFR* (engl. *Most Forward Routing*). Druga strategija po kojoj se poruka prosleđuje čvoru koji obezbeđuje pozitivan napredak, a da se pritom minimizuje odstupanje od linije propagacije SD (u primeru sa slike u pitanju je čvor F), obeležava se skraćenicom *LDR* (engl. *Least Deviation from straight line Routing*).



Slika 2.10: Strategije rutiranja registracionih poruka

Formalno, kada robot S koji se nalazi na poziciji (x_S, y_S) inicira registracionu poruku, odredište je virtuelni čvor D na poziciji (x_D, y_D) , čije su koordinate zadate na sledeći način:

$$(x_D, y_D) = \begin{cases} (x_S, \infty) & , \text{ ako se poruka šalje na sever} \\ (\infty, y_S) & , \text{ ako se poruka šalje na istok} \\ (x_S, -\infty) & , \text{ ako se poruka šalje na jug} \\ (-\infty, y_S) & , \text{ ako se poruka šalje na zapad} \end{cases}$$

Prilikom rutiranja, čvor u na poziciji (x_u, y_u) prosleđuje poruku čvoru $v \in N(u)$ na poziciji (x_v, y_v) . Pri *MFR* rutiranju, čvor v bira se na sledeći način:

- $(y_v > y_u) \wedge (\forall w \in N(u))(y_v - y_u \geq y_w - y_u)$, ako se poruka šalje na sever
- $(x_v > x_u) \wedge (\forall w \in N(u))(x_v - x_u \geq x_w - x_u)$, ako se poruka šalje na istok
- $(y_v < y_u) \wedge (\forall w \in N(u))(y_u - y_v \geq y_u - y_w)$, ako se poruka šalje na jug
- $(x_v < x_u) \wedge (\forall w \in N(u))(x_u - x_v \geq x_u - x_w)$, ako se poruka šalje na zapad

Sa druge strane, prilikom *LDR* rutiranja, kriterijumi za izbor čvora v kojem se prosleđuje poruka su sledeći:

- $(y_v > y_u) \wedge (\forall w \in N(u))(|x_v - x_D| \leq |x_w - x_D|)$, ako se poruka šalje na sever
- $(x_v > x_u) \wedge (\forall w \in N(u))(|y_v - y_D| \leq |y_w - y_D|)$, ako se poruka šalje na istok
- $(y_v < y_u) \wedge (\forall w \in N(u))(|x_v - x_D| \leq |x_w - x_D|)$, ako se poruka šalje na jug
- $(x_v < x_u) \wedge (\forall w \in N(u))(|y_v - y_D| \leq |y_w - y_D|)$, ako se poruka šalje na zapad

Postoje dva uslova pod kojima se prekida dalja propagacija registracione poruke koja je inicirana od strane robota r :

- Poruka je stigla do čvora koji nema suseda u pravcu prostiranja poruke, što znači da je dostignuta ivica terena.
- Prijemni čvor v već pripada u *iMesh* pojasu drugog robota r' . Ako je $|vr'| < |vr|$, prekida se dalja propagacija, a u suprotnom se nastavlja. Ovo je modifikacija pravila blokiranja iz [LSS09].

Dodatni detalj koji treba istaći je da u slučaju kada registraciona poruka koja potiče od robota r dođe do čvorova koji su već u *iMesh* pojasu drugog robota r' , ako se ispostavi da im je robot r bliži nego r' , potrebno je da ažuriraju informaciju o najbližem izvršiocu koordinatama robota r .

Modifikovane protokole nazvali smo *iMeshC* i *iMeshD*, budući da su protokoli koji su predstavljeni u radu [LSS09] nazvani *iMeshA* and *iMeshB*. Ako se primenjuje *MFR* rutiranje, dobija se *iMeshC* struktura prikazana na slici 2.9a. Uočava se da propagacione putanje poruka imaju tendenciju da se savijaju, usled lokalnih nepravilnosti u distribuciji čvorova. Sa druge strane, primenom *LDR* rutiranja koje umesto najudaljenijeg čvora poruku prosleđuje čvoru koji obezbeđuje minimalno odstupanje od zamišljene prave linije, dobija se *iMeshD* struktura prikazana na slici 2.9b. Kao što se uočava na slici, *iMeshD* se odlikuje ravnijim propagacionim linijama, što rezultuje približno pravougaonim oblikom *iMesh* ćelija. Formiranje *iMesh* informacione strukture opisano je u algoritmu 5.

U oba slučaja, kao kriterijum da je dostignuta ivica terena uzima se nemogućnost daljeg prosleđivanja poruke u željenom pravcu. Ovo je prihvatljivo pod pretpostavkom da je mreža gusta, što potvrđuju rezultati simulacija izneti u poglavlju 4. U retkim mrežama, pošto se propagacija registracionih poruka u zaustavlja u slučaju postojanja praznina u topologiji, *iMeshC* i *iMeshD* ne moraju nužno biti dobro rešenje. Stoga, u mrežama koje nisu guste i/ili koje nemaju ravnomernu raspodelu senzorskih čvorova celom svojom površinom, bolji izbor predstavlja upotreba novog protokola koji je zasnovan na lokalizovanom preplavlivanju i predstavljen je u odeljku 2.3.

Algoritam 5 Formiranje *iMesh* strukture

```

1: for  $i = 1 \rightarrow m$  do
2:    $S_{i.najblizi\_robot} \leftarrow NULL$ 
3: end for
4: for  $i = 1 \rightarrow n$  do
5:    $R_i \in \mathcal{R}$  šalje registracione poruke susedima u 4 geografska pravca u skladu sa
      pravilom rutiranja; svaka poruka sadrzi informaciju o poziciji robota  $R_i$  i pravcu
      propagacije
6: end for
7: while postoje poruke koje su primljene, a nisu obrađene do
8:   if čvor  $S_i$  je primio poruku sa informacijom o robotu  $R_k$  then
9:      $R_c \leftarrow S_{i.najblizi\_robot}$ 
10:    if  $(R_c = NULL) \vee (|S_i R_k| < |S_i R_c|)$  then
11:       $S_{i.najblizi\_robot} \leftarrow R_k$ 
12:      if primljena poruka je adresirana na čvor  $S_i$  then
13:         $S_i$  u sledećem krugu prosleđuje poruku koja sadrži informaciju o robotu
           $R_k$  sledećem čvoru u pravcu propagacije, u skladu sa pravilom rutiranja
14:      end if
15:    else
16:      registraciona poruka se odbacuje
17:    end if
18:  end if
19:  čekaj sledeći krug
20: end while

```

2.2.3 Ukrštena pretraga

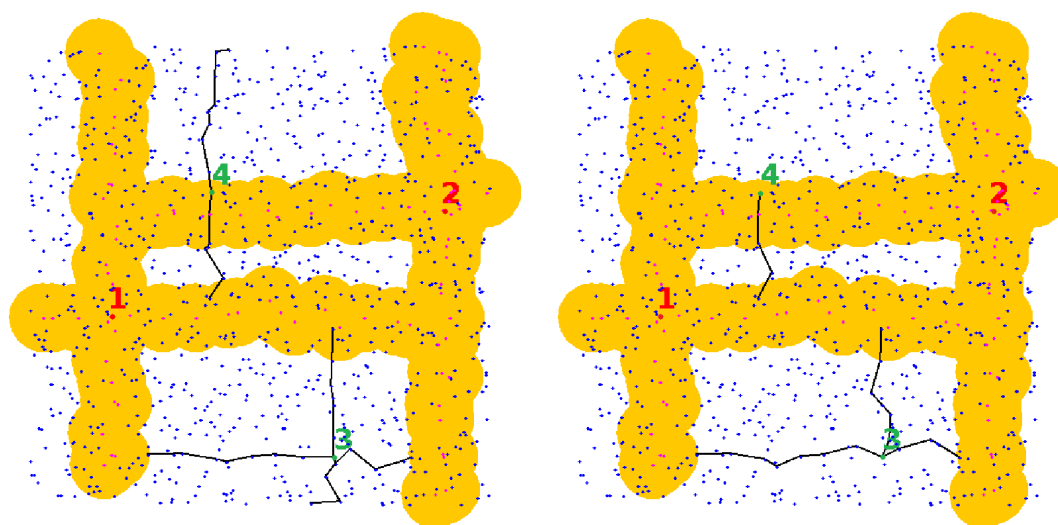
Kada je uspostavljena informaciona struktura, po detekciji događaja od strane nekog od senzorskih čvorova, sprovodi se pretraga u cilju pronalaženja lokacije najbližeg robota, kojem je potrebno dodeliti ulogu izvršioca. Proces pretrage poseduje svojstvo osetljivosti na razdaljinu, budući da je ograničen na ćeliju u kojoj je locirano mesto događaja. Senzor koji je detektovao događaj inicira slanje upita koji se propagiraju u četiri pravca, na isti način kao i registracione poruke, sve dok ne stignu do čvora koji je unutar nekog od *iMesh* pojaseva, ili do ivice terena. U prvom slučaju, prijemni čvor koji je unutar *iMesh* pojasa generiše odgovor koji sadrži informaciju o robotu i koji se šalje nazad čvoru koji je inicirao upit. Odgovor se uvek rutira korišćenjem *MFR* šeme, iz sledećih razloga:

- Na ovaj način izbegava se potreba za pamćenjem putanje prilikom prosleđivanja upita, budući da bi se u tom slučaju svakim hopom povećavala količina informacije koja se smešta u poruku (ili u memoriju čvorova koji je prosleđuju). Stoga je moguće da se putanja odgovora razlikuje od putanje upita.
- Ovaj način rutiranja odgovora može rezultovati manjim brojem hopova koje odgovor treba da pređe dok ne stigne do čvora koji je inicirao upit.

U drugom slučaju, kada upit stigne do ivice terena, a da usput nije naišao ni na jedan *iMesh* pojas, upit se odbacuje bez generisanja odgovora.

Čvor koji sprovodi pretragu generiše upite koji se prosleđuju na sve četiri strane, ukoliko se nalazi unutar *iMesh* ćelije, tj. ako ne pripada nijednom od postojećih *iMesh* pojaseva. U suprotnom, ako je lociran u nekom od pojaseva, na prvi pogled deluje logično da je proces pretrage izlišan, pošto senzorski čvor već raspolaže informacijom o robotu. Ipak, taj robot ne mora nužno biti najbliži mestu događaja, što je ilustrovano primerom na slici 2.11. Čvor 4 koji sprovodi pretragu nalazi se unutar *iMesh* pojasa robota 2, ali mu je bliži robot 1. Kako bi došao do informacije o bližem robotu, čvor 4 šalje upite samo u pravcu koji je normalan na pravac pružanja *iMesh* pojasa kojem pripada, a to je u ovom slučaju pravac sever-jug. Upit koji se prosleđuje ka severu stiže do ivice terena i stoga se odbacuje bez generisanja odgovora, dok upit koji je usmeren ka jugu dolazi do *iMesh* pojasa robota 1², pri čemu se generiše odgovor koji se prosleđuje nazad čvoru 4. Na ovaj način omogućeno je da čvor sazna informaciju o robotu 1, iako se nalazi u informacionom pojasu generisanom od strane drugog robota. Ukoliko pretragu sprovodi čvor 3 koji je lociran u unutrašnjosti *iMesh* ćelije, upiti se regularno prosleđuju u sva četiri pravca, kao što je prikazano na slici 2.11a, dok odgovori stižu iz tri pravca u kojima su upiti stigli do *iMesh* pojasa, što se uočava na slici 2.11b.

²Pri rutiranju južnog upita, ignoriše se uslov za generisanje odgovora sve dok upit ne izađe iz pojasa robota 2.



(a) Putanje propagacije upita

(b) Putanje propagacije odgovora

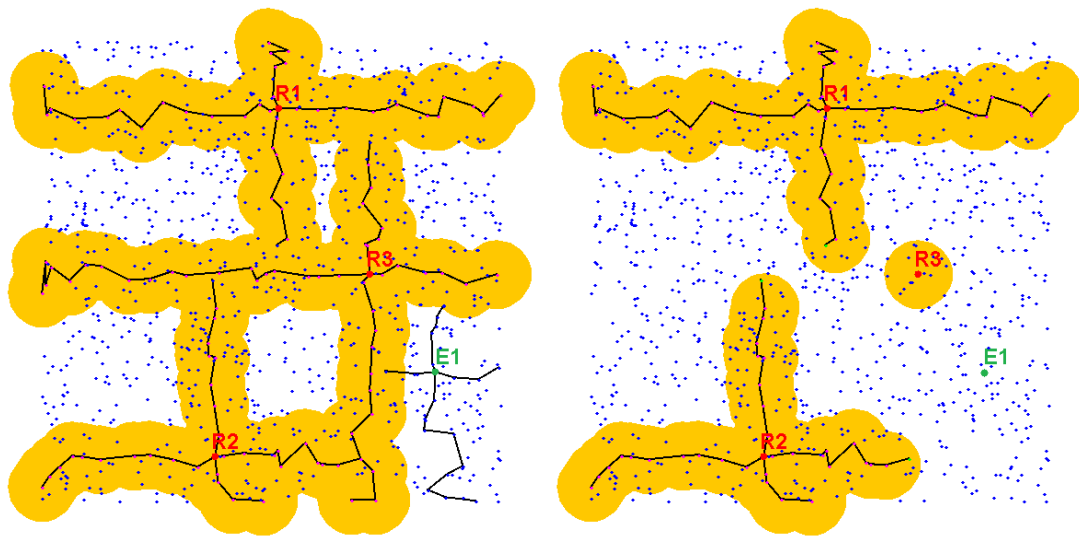
Slika 2.11: *iMeshD* ukrštena pretraga

2.2.4 Ažuriranje lokacije u okviru *iMesh* strukture

Algoritam 6 *iMesh* ažuriranje lokacije

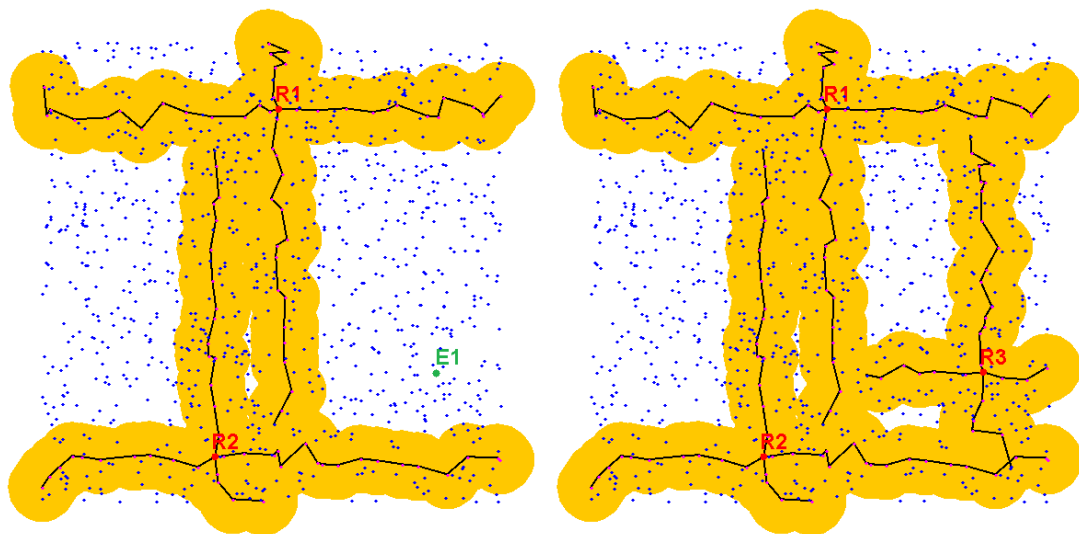
- 1: čvor E detektuje događaj
 - 2: čvor E otkriva lokaciju robota R ukrštenom pretragom
 - 3: čvor E šalje robotu R zahtev za izvršenje zadatka
 - 4: robot R šalje odjavne poruke u 4 geografska pravca
 - 5: odjavne poruke se propagiraju na isti način kao *iMesh* registracione poruke; čvorovi u okviru *iMesh* pojaseva robota R brišu informaciju o robotu R
 - 6: čvorovi kod kojih su registracione poruke koje potiču od drugih robota bile blokirane usled postojanja informacije o robotu R re-inicijalizuju proces registracije u odgovarajućim pravcima
 - 7: nastavlja se proces *iMesh* registracije čime se šire informacioni pojasi koji su prethodno bili blokirani
 - 8: robot R se premešta na poziciju događaja E
 - 9: robot R šalje registracione poruke u 4 geografska pravca sa svoje nove pozicije
 - 10: izvršava se *iMesh* registracija u cilju kreiranja *iMesh* pojaseva sa informacijom o novoj poziciji robota R
-

U ovom odeljku opisano je proširenje *iMesh* protokola koje smo predstavili u radu [LM15]. U pitanju je algoritam koji u slučaju promene lokacije jednog od robota usled odlaska na izvršenje zadatka, vrši ažuriranje njegove lokacije u smislu ponovnog uspostavljanja *iMesh* strukture na lokalizovan način, bez potrebe da cela struktura bude ponovo formirana iz početka. Ovaj proces prikazan je na slici 2.12. Za početak, kreira se *iMesh* na način opisan u odeljku 2.2.2. Po detekciji događaja, senzorski čvor E_1 sprovodi ukrštenu pretragu, čime otkriva poziciju robota R_3 (slika 2.12a). Potom E_1



(a) *iMesh* primer sa 3 robota; E_1 otkriva R_3 ukrštenom pretragom

(b) R_3 se odjvljuje



(c) *iMesh* pojasevi R_1 i R_2 se šire

(d) R_3 odlazi na lokaciju E_1 i prijavljuje se

Slika 2.12: Ažuriranje lokacije robota u okviru *iMesh* strukture

šalje robotu R_3 zahtev za obilazak mesta događaja. Pre nego što krene na izvršenje zadatka, R_3 se odjvljuje slanjem odjavnih poruka u četiri geografska pravca, na istovetan način kao tokom faze uspostavljanja *iMesh* strukture. Čvorovi u okviru informacionih pojaseva robota R_3 brišu informaciju o robotu R_3 i time više ne predstavljaju deo *iMesh* informacione strukture (slika 2.12b). Čvorovi kod kojih je primenjeno pravilo blokiranja ponovo pokreću prekinuti proces propagacije registracionih poruka, što rezultuje širenjem informacionih pojaseva robota R_1 i R_2 tokom sledeće faze (slika 2.12c). Konačno, robot R_3 dolazi na poziciju E_1 , i izvršava

zadatak. Nakon toga, ponovo se prijavljuje na mrežu slanjem registracionih poruka (slika 2.12d). Koraci koji se obavljaju prilikom ažuriranja lokacije sumirani su u algoritmu 6.

2.3 vCell protokol

Kao što je već napomenuto u odeljku 2.1, rešenje problema otkrivanja izvršilaca sa osobinom osetljivosti na razdaljinu podrazumevalo bi konstrukciju Voronoi dijagrama. U radu [LSS09] sugerise se da je za kreiranje Voronoi dijagrama neophodno globalno znanje. Međutim, kao jedan od doprinosa ove disertacije, u radu [LM15] predstavili smo novi protokol nazvan *vCell*, koji na lokalizovan način kreira informacionu strukturu čija topologija podražava strukturu Voronoi dijagrama. Ovaj protokol razvijen je vodeći računa o ostvarenju tri ključna cilja:

- Upotreba u mrežama sa neravnomernom raspodelom čvorova, koje mogu sadržati praznine u topologiji.
- Konstruisanje informacione strukture koja bi proces pretrage učinila redundantnim, obezbeđujući unapred svakom senzorskom čvoru informaciju o najbližem robotu.
- Obezbeđivanje optimalnih putanja za rutiranje zahteva za izvršenjem zadataka od čvorova koji detektuju događaje do robota.

U nastavku opisano je funkcionisanje *vCell* protokola. Osnovne pretpostavke vezane za mrežni model, kao i matematička notacija identični su kao u odeljku 2.2.1.

2.3.1 Formiranje vCell informacione strukture

Novi protokol *vCell* gradi informacionu strukturu putem lokalizovanog plavljenja. Formiranje ove strukture odvija se u vremenski sinhronizovanim iteracijama (krugovima), kao što je opisano u algoritmu 7. Za početak, svaki robot emituje registracionu poruku koja sadrži informaciju o njegovoj poziciji, koju primaju njegovi neposredni (1-hop) susedi. Po prijemu registracione poruke, svaki od senzorskih čvorova ažurira informaciju o najbližem robotu i propagira je dalje u sledećem krugu, ukoliko je primljena poruka dovela do promene informacije o najbližem robotu u datom krugu. Na ovaj način u okviru mreže kreira se struktura ćelija. Uz to, čvorovi čuvaju i informaciju o čvorovima od kojih su primili registracione poruke, odnosno o svojim roditeljima. Na ovaj način u svakoj ćeliji formira se struktura komunikacionog stabla, čiji koren je robot koji je inicirao proces formiranja ćelije. Kasnije, po

detektovanju događaja, ova struktura obezbeđuje optimalnu putanju za rutiranje zahteva za obilazak mesta događaja. Propagacija registracionih poruka okončava se po dolasku u pogranični region sa ćelijom drugog robota, ili po nailasku na prazan region koji predstavlja prazninu u mrežnoj topologiji, ili ivicu terena. Rezultujuća ćelijska struktura nalikuje Voronoi dijagramu, što je prikazano na slici 2.13b.

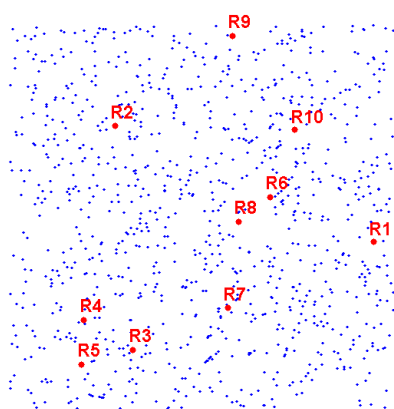
Pošto je mehanizam formiranja *vCell* strukture zasnovan na plavljenju, time je omogućeno propagacionim putanjama registracionih poruka da zaobilaze praznine u mrežnoj topologiji, čime se omogućava otkrivanje obližnjeg robota čak i u situacijama u kojima druge šeme za otkrivanje izvršilaca kao što su pravougaoni kvorum i *iMesh* ne uspevaju da pronađu izvršioca, ili dolaze do sub-optimalnog rešenja. Slika 2.14 ilustruje situaciju kada je u mreži prisutna praznina u obliku polumeseca. Čvor E_1 koji je detektovao događaj otkriva robot R_2 putem *iMesh* ukrštene pretrage. Informacioni pojasevi, kao i putanje upita prilikom ukrštene pretrage prikazani su na slici 2.14a. Robot R_3 je bliži mestu događaja (odnosno čvoru E_1) nego R_2 , ali R_3 ne može biti otkriven na ovaj način. Sa druge strane, E_1 pripada *vCell* ćeliji robota R_3 (slika 2.14b), usled prirode *vCell* protokola koji je zasnovan na plavljenju. Ovo je jedan od primera kada *iMeshD* ne uspeva da pronađe najbliži robot usled prisustva praznine u mrežnoj topologiji. Štaviše, u ovom slučaju robot R_3 ne bi mogao biti otkriven čak ni u slučaju da je primenjen *iMesh* protokol sa *GFG* rutiranjem (koje je u stanju da obiđe prepreku), budući da bi propagacione putanje registracionih poruka robota R_3 bile izblokirane od strane robota R_2 . Dakle, ovakav primer ilustruje da je *vCell* protokol u mogućnosti da bolje izlazi na kraj sa problematičnim prazninama konkavnog oblika.

Algoritam 7 Formiranje *vCell* strukture

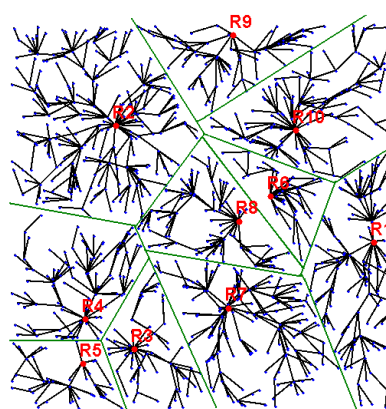
```

1: for  $i = 1 \rightarrow m$  do
2:    $S_{i.najblizi\_robot} \leftarrow NULL$ 
3:    $S_{i.roditelj} \leftarrow NULL$ 
4: end for
5: for  $i = 1 \rightarrow n$  do
6:    $R_i \in R$  šalje registracionu poruku
7: end for
8: while postoje poruke koje su primljene, a nisu obrađene do
9:   if čvor  $S_i$  je primio poruku sa informacijom o robotu  $R_k$  then
10:     $R_c \leftarrow S_{i.najblizi\_robot}$ 
11:    if  $(R_c = NULL) \vee (|S_i R_k| < |S_i R_c|)$  then
12:       $S_{i.najblizi\_robot} \leftarrow R_k$ 
13:       $S_{i.roditelj} \leftarrow$  pošiljalac poruke
14:       $S_i$  u sledećem krugu šalje poruku koja sadrži informaciju o robotu  $R_k$ 
15:    end if
16:  end if
17:  čekaj sledeći krug
18: end while

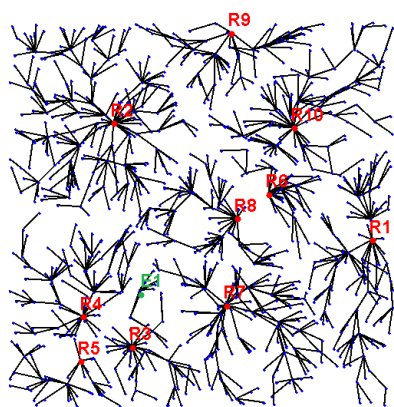
```



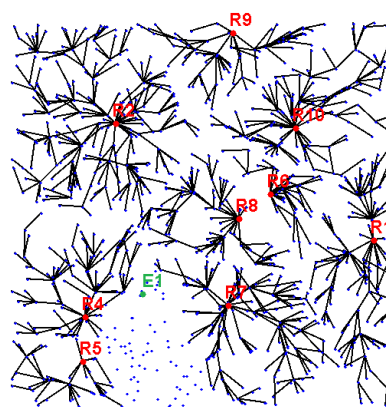
(a) Mreža sa 10 robota



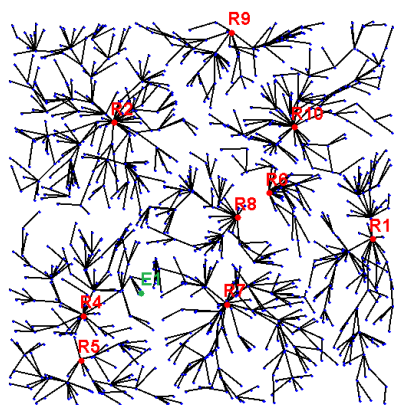
(b) vCell struktura



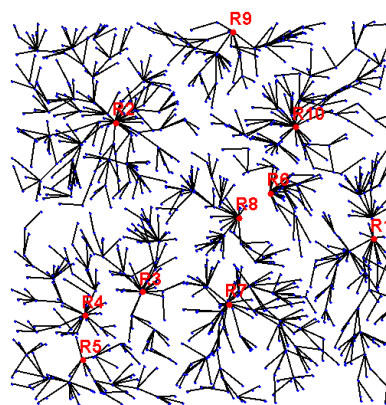
(c) Detektovan je događaj E_1



(d) Robot R_3 se odjavljuje E_1

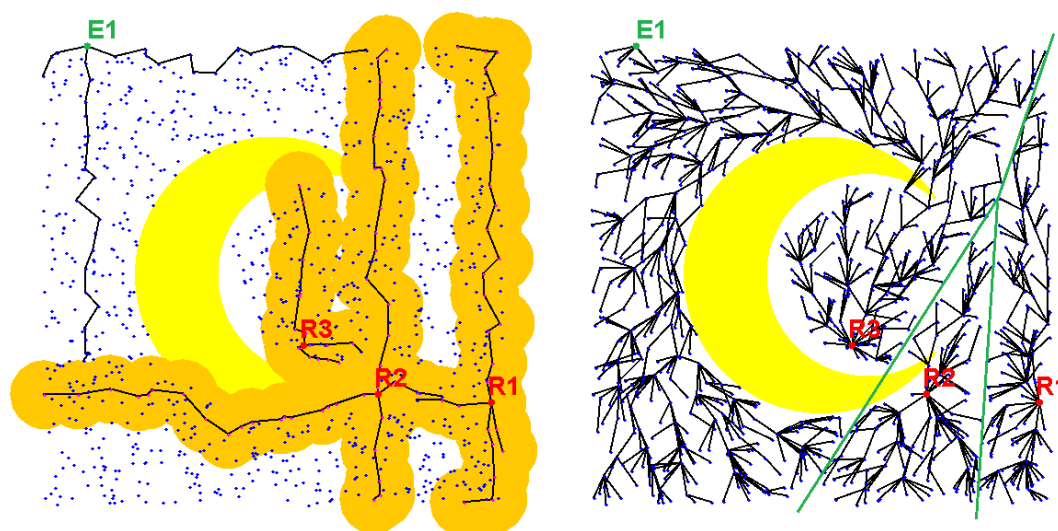


(e) vCell nakon odjave R_3



(f) vCell nakon prijave R_3

Slika 2.13: vCell struktura



(a) E_1 otkriva R_2 putem ukrštene pretrage

(b) E_1 je u ćeliji robota R_3

Slika 2.14: *iMeshD* i *vCell* u mreži sa prazninom u obliku polumeseca

Ovde je neophodno istaći da se smatra da praznine u mrežnoj topologiji predstavljaju površine koje iz nekog razloga nisu pokrivene senzorskim čvorovima, ali koje kao takve ne predstavljaju prepreku kretanju robota³. U tom slučaju je korektna pretpostavka da je najbliži robot ujedno i najbolji kandidat za izvršioca. U suprotnom, potrebno je primeniti neku od šema za planiranje robotskih putanja kao što je A* (A-star) algoritam [HNR68], što izlazi iz tematike ove disertacije.

2.3.2 Ažuriranje lokacije u okviru vCell strukture

U ovom odeljku biće prikazan način za prilagođenje mrežne strukture promenama uzrokovanim kretanjem robota. Slika 2.13c prikazuje situaciju kada je događaj detektovan od strane senzorskog čvora E_1 koji se nalazi u ćeliji robota R_3 . E_1 šalje zahtev za obilaskom mesta događaja, koristeći komunikaciono stablo kao strukturu koja obezbeđuje optimalnu putanju za rutiranje zahteva. R_3 prihvata zahtev, usled čega napušta mrežnu strukturu sprovodeći operaciju koja se naziva odjavom sa mreže. Odjavna poruka preplavljuje mrežnu ćeliju robota R_3 na isti način kao što je to bio slučaj sa registracionom porukom. Po prijemu odjavne poruke, čvorovi koji su bili unutar ćelije brišu informaciju o R_3 , čime se dobija "prazna" ćelija, što je prikazano na slici 2.13d. Čvorovi koji pripadaju susednim ćelijama i koji su primili odjavnu poruku (u pitanju su čvorovi u pograničnoj oblasti sa bivšom ćelijom R_3), ponovo pokreću registracioni proces, što rezultuje ekspanzijom susednih ćelija, čime se popunjava

³Npr. roboti mogu biti dronovi koji se odlikuju sposobnošću letenja.

praznina (slika 2.13e). Kao rezultat, ponovo se uspostavlja *vCell* struktura na nivou mreže, kao da R_3 nije ni bio prisutan u mreži. Konačno, R_3 odlazi na lokaciju E_1 , kako bi obavio zadatak, nakon čega ponovo šalje registracionu poruku u cilju kreiranja nove ćelije, kao što je prikazano na slici 2.13f. Ova poslednja operacija naziva se prijavom na mrežu.

Očigledno, ista šema može biti korišćena kada je potrebno da roboti privremeno napuste mrežu, npr. radi odlaska do stanice za dopunjavanje baterija, ili prilikom uvođenja novih robota u postojeću mrežnu strukturu. Proces ažuriranja lokacije u okviru *vCell* strukture opisan je u algoritmu 8.

Algoritam 8 *vCell* ažuriranje lokacije

```

1:  $E$  detektuje događaj i šalje zahtev robotu  $R_j$  ( $R_j = E.najbliži\_robot$ )
2: for  $i = 1 \rightarrow m$  do
3:    $S_{i.posalji\_registracionu\_poruku} \leftarrow FALSE$ 
4: end for
5:  $R_j$  šalje odjavnu poruku
6: while ima neobrađenih odjavnih poruka do
7:   if čvor  $S_i$  je primio odjavnu poruku robota  $R_j$  then
8:      $R_c \leftarrow S_{i.najbliži\_robot}$ 
9:     if  $R_c = R_j$  then
10:       $S_{i.najbliži\_robot} \leftarrow NULL$ 
11:       $S_{i.roditelj} \leftarrow NULL$ 
12:      prosledi dalje odjavnu poruku robota  $R_j$  u sledećem krugu
13:    else
14:       $S_{i.posalji\_registracionu\_poruku} \leftarrow TRUE$ 
15:    end if
16:  end if
17:  sačekaj sledeći krug
18: end while
19: for  $i = 1 \rightarrow m$  do
20:   if  $S_{i.posalji\_registracionu\_poruku} = TRUE$  then
21:      $S_i$  šalje registracionu poruku sa informacijom o robotu  $S_{i.najbliži\_robot}$ 
22:   end if
23: end for
24: izvrši korake 8 – 18 formiranja vCell strukture
25:  $R_j$  se pomera na poziciju  $E$ 
26:  $R_j$  šalje registracionu poruku
27: izvrši korake 8 – 18 formiranja vCell strukture

```

2.4 Procena komunikacionih troškova

Poređenje performansi komunikacionih protokola vrši se na osnovu komunikacionih troškova koji su nametnuti njihovom upotrebom. Pod ovim pojmom podrazumeva se broj transmisija u bežičnoj mreži. Pri tome važi princip da je količina informacije koja se prenosi slanjem svake poruke konstantna, pa se za svaku pojedinačnu transmisiju vezuje jedinična cena. Ovde će biti reči o teorijskoj estimaciji komunikacionih troškova, dok rezultati simulacija koji su prikazani u poglavlju 4, odeljak 4.1 omogućavaju realniji uvid u njihovu stvarnu veličinu.

Prilikom analize koja će biti izložena u ovom odeljku, komunikacioni troškovi biće izraženi velikom "O" notacijom koja je preuzeta iz teorije složenosti algoritama [Pap03][CLR⁺01], sa izvesnim modifikacijama. Na primer, u radovima iz oblasti *BSM* uobičajeno je da se pojavi izraz kao što je $2 \cdot O(n) + O(\sqrt{n})$, što označava da se obavljaju tri operacije, pri čemu je cena prve dve $O(n)$, a treće $O(\sqrt{n})$. Ovo odstupa od notacije korišćene za izražavanje složenosti algoritama, gde se zanemaruju multiplikativne konstante i članovi nižeg reda.

U cilju pojednostavljenja analize, uvodi se pretpostavka da je mreža kvadratnog oblika sa uniformnom raspodelom senzorskih čvorova. Životni vek svakog od analiziranih protokola podeljen je na tri faze:

- **Formiranje direktorijuma:** Tokom ove faze formira se informaciona struktura (direktorijum) koji se koristi za otkrivanje izvršilaca. Priroda direktorijuma je distribuirana, a poželjna osobina je osetljivost na razdaljinu. Na ovaj način, informacija o položaju robota biva raspodeljena među senzorskim čvorovima, tako da je za funkcionisanje protokola dovoljno da svi, ili samo neki od čvorova raspolažu informacijom o obližnjem robotu.
- **Otkrivanje i obaveštavanje izvršilaca:** Ova faza podeljena je na dve operacije. Prvo čvorovi koji su detektovali događaje dolaze do informacije o obližnjim robotima sprovodeći operaciju pretrage. Nakon toga se šalju zahtevi za obilazak mesta događaja robotima koji su otkriveni tokom operacije pretrage.
- **Održavanje direktorijuma:** Informacioni direktorijum se ažurira, u skladu sa promenama topologije izazvanim kretanjem robota, odnosno njihovim premeštanjem na mesta događaja. Robot sprovodi ažuriranje putem operacija odjave i prijave. Operacija odjave podrazumeva brisanje informacije o staroj poziciji robota iz informacionog direktorijuma, dok se operacijom prijave unosi u direktorijum informacija o novoj poziciji robota.

Analiza komunikacionih troškova tokom na faze otkrivanja i obaveštavanja izvršilaca, kao i održavanja direktorijuma odnosi se na slučaj detekcije pojedinačnog događaja i

posledične promene položaja jednog robota, kao izvršioca zadatka obilaska mesta događaja. U slučaju pojave višestrukih događaja, potrebno je ove troškove pomnožiti njihovim brojem. U tabeli 2.1 sumirani su komunikacioni troškovi tokom različitih faza kvorum, *iMesh* i *vCell* protokola.

	Troškovi formiranja	Troškovi otkrivanja	Troškovi održavanja
kvorum	$m + n \cdot O(\sqrt{m})$	$3 \cdot O(\sqrt{m})$	$2 \cdot O(\sqrt{m})$
<i>iMesh</i>	$m + O(\min\{n \cdot \sqrt{m}, m\})$	$2 \cdot O(\sqrt{m})$	$O(n \cdot \sqrt{m}) + 2 \cdot O(\sqrt{m})$
<i>vCell</i>	$O(m)$	$O(\sqrt{\lceil \frac{m}{n} \rceil})$	$3 \cdot O(\lceil \frac{m}{n} \rceil)$

Tabela 2.1: Procena komunikacionih troškova tokom različitih faza protokola za otkrivanje izvršilaca

2.4.1 Komunikacioni troškovi pravougaonog kvoruma

Inicijalni troškovi kreiranja informacione strukture pravougaonog kvoruma proizilaze iz procesa formiranja lista suseda. Za početak, svaki čvor šalje pozdravnu poruku, čime omogućava svojim neposrednim susedima da saznaju njegovu poziciju. Potom, svaki robot inicira propagaciju informacije o svojoj poziciji duž kolone u kojoj je prisutan. U radu [SLJ08] pokazano je da je u pravougaonom kvorumu cena svake od operacija (u koje spadaju formiranje i pretraga) $O(\sqrt{m})$. Stoga, ako se uračuna i inicijalizacija, ukupni troškovi formiranja kvoruma su $m + n \cdot O(\sqrt{m})$.

Tokom faze otkrivanja i obaveštavanja izvršilaca, pretraga podrazumeva slanje upita duž vrste, dok se ne dostignu ivice terena, kao i prosljeđivanje odgovora duž iste vrste. Uz to, potrebno je i poslati zahtev robotu koji je otkriven kao pogodan kandidat za izvršioca. U pitanju su tri operacije, čija ukupna cena iznosi $3 \cdot O(\sqrt{m})$.

Ažuriranje lokacije robota, koje spada u fazu održavanja direktorijuma, sastoji se iz dve operacije. Prva je odjava iz trenutne kolone, a druga je prijava koja se sprovodi nakon relokacije robota. Stoga, ukupni troškovi vezani za ovu fazu, kada je u pitanju ažuriranje lokacije pojedinačnog robota iznose $2 \cdot O(\sqrt{m})$.

2.4.2 Komunikacioni troškovi iMesh protokola

Analiza koja je prikazana u radu [LSS09] pokazuje da su troškovi kreiranja *iMesh* informacione strukture $O(\min\{n \cdot \sqrt{m}, m\})$. U troškove formiranja direktorijuma ponovo je potrebno uračunati slanje pozdravnih poruka, što ukupno čini $m + O(\min\{n \cdot \sqrt{m}, m\})$.

Faza otkrivanja i obaveštavanja izvršilaca podrazumeva ukrštenu pretragu i prosljeđivanje zahteva robotu, što su dve operacije čija ukupna cena iznosi $2 \cdot O(\sqrt{m})$.

Tokom faze održavanja, obavljaju se tri operacije: odjava robota, čija cena je $O(\sqrt{m})$, zatim proširenje *iMesh* informacionih pojaseva drugih robota, čija cena⁴ je $O(n \cdot \sqrt{m})$ i konačno, operacija prijave koja košta $O(\sqrt{m})$.

2.4.3 Komunikacioni troškovi vCell protokola

Imajući u vidu da je *vCell* protokol zasnovan na plavljenju, za njegovo funkcionisanje nije neophodno da čvorovi imaju predznanje o pozicijama svojih suseda. Stoga, proces otkrivanja susedstva može biti integrisan sa formiranjem informacionog direktorijuma, bez nametanja dodatnih komunikacionih troškova. Kompletna mreža se preplavljuje jednom tokom faze formiranja direktorijuma, pri čemu troškovi iznose $O(m)$.

Esencijalno svojstvo *vCell* informacionog direktorijuma je da svaki čvor raspolaže informacijom o robotu, kao i o svom roditelju u komunikacionom grafu. Prema tome, jedini trošak tokom druge faze proističe iz prosleđivanja zahteva robotu. U proseku, u svakoj ćeliji prisutno je $\lceil \frac{m}{n} \rceil$ čvorova, što znači da su troškovi prosleđivanja zahteva robotu $O(\sqrt{\lceil \frac{m}{n} \rceil})$.

Tokom faza održavanja obavljaju se tri operacije: odjava, proširenje susednih ćelija i prijava, odnosno kreiranje nove ćelije, u skladu sa novom pozicijom robota. Cena svake od ovih operacija je $O(\lceil \frac{m}{n} \rceil)$, što znači da je ukupna cena vezana za fazu održavanja $3 \cdot O(\lceil \frac{m}{n} \rceil)$.

⁴Ovde je u pitanju konzervativna procena koja se odnosi na najgori slučaj, kada se pretpostavlja da je robot koji se odjavljuje blokirao propagaciju registracionih poruka svih ostalih robota.

Poglavlje 3

Koordinacija među robotima u slučaju višestrukih događaja

Kao što je već pomenuto u uvodu, u ovom poglavlju predstavljena su nova centralizovana i distribuirana rešenja problema dodele zadataka izvršiocima, u slučaju pojave višestrukih događaja u *BMSR*. Nova rešenja predstavili smo na idejnom nivou u radu [LS13], a zatim smo ih detaljnije elaborirali u proširenoj i dopunjenoj verziji ovog rada [LBS15]. Rezultati simulacija koji su izloženi u poglavlju 4, odeljak 4.2 pokazuju značajna poboljšanja koja novopredložena rešenja donose u odnosu na relevantna postojeća rešenja o kojima je bilo reči u poglavlju 1, odeljak 1.6.

Sledi kratko podsećanje na postavku problema. Pretpostavlja se da su roboti vremenski sinhronizovani i da zadatke obavljaju u rundama jednakog trajanja. Ukoliko su tokom pojedinačne runde zadati skupovi mesta događaja $E = \{E_1, E_2, \dots, E_m\}$ i robota $R = \{R_1, R_2, \dots, R_n\}$, cilj je dodeliti svakom robotu R_i listu $DS_i \subset E$, koja sadrži one događaje koje R_i treba da poseti tokom dotične runde. Pri tome važe uslovi:

$$\bigcup_{i=1}^n DS_i = E \quad (3.1)$$

$$(\forall i)(\forall j)(1 \leq i, j \leq n \wedge i \neq j \Rightarrow DS_i \cap DS_j = \emptyset) \quad (3.2)$$

Drugim rečima, potrebno je svaki od događaja uvrstiti u listu za obilazak tačno jednog od robota. Uslov 3.2 važi pod pretpostavkom da je priroda događaja takva da je dovoljno da svaki događaj bude posećen od strane jednog robota. Ako je robotima dozvoljeno da tokom runde obiđu više događaja, problem se označava kao *s-dodela*(m, n), u kojem se vrši sekvencijalno upućivanje n robota kao izvršilaca na zadatke obilazaka m mesta događaja. Takođe, razmatrana je i varijanta ovog problema sa restrikcijom da se tokom runde svakom robotu dodejuje najviše jedan događaj, koja

se označava kao m -dodela(m, n). Odbacivanjem uslova 3.2, dobija se generalizovana verzija problema s -dodela(m, n), u kojoj događaji mogu zahtevati da budu posećeni od strane više različitih robota. Algoritam koji rešava ovako formulisan problem prikazan je u odeljku 3.2.2.

Od interesa je scenario u kojem se izvršenje zadataka u automatizovanoj *BMSR* obavlja u više krugova (rundi) jednakog trajanja. Na početku svake runde, prikupljaju se informacije o događajima koji su se pojavili tokom prethodne runde, nakon čega se za svaki od robota formira lista događaja za obilazak. Prilikom obilaska pojedinačnog događaja, ili njihove sekvence u zavisnosti od prirode algoritma za dodelu zadataka, roboti troše deo raspoložive energije na kretanje. Pretpostavlja se da je trajanje pojedinačne runde dovoljno da roboti obave sve zadatke, bez obzira na njihov broj, kao i na međusobne udaljenosti između lokacija događaja koji su im dodeljeni. Početna rešenja koja su prikazana u odeljcima 3.2.1 i 3.2.2 poštuju ovu vremensku pretpostavku, a potom je u odeljku 3.2.3 prikazana i varijanta centralizovanog algoritma koja uzima u obzir vremenska ograničenja, tako da je moguć slučaj da runda istekne pre nego što roboti stignu da obiđu sve događaje koji su im dodeljeni. Takvi događaji prenose se u sledeću rundu i tom prilikom im se ponovo dodeljuju izvršioци.

Po završetku svake runde, roboti se zaustavljaju na krajnjim pozicijama svojih putanja, gde čekaju na početak sledeće runde. Broj i lokacije događaja koji će se pojaviti tokom naredne runde smatraju se nepredvidljivim i ne mogu se dovesti u direktnu vezu sa brojem i lokacijama događaja koji su se pojavili tokom tekuće, ili neke od prethodnih rundi¹. Uz to, odnos između broja događaja i broja robota može biti proizvoljan. Teren po kojem su raspoređeni senzorski čvorovi i roboti smatra se homogenim, u smislu ravnomerne prostorne distribucije senzorskih čvorova, kao i odsustva prepreka koje bi onemogućavale, ili otežavale pravolinijsko kretanje robota.

Životno vreme sistema definiše se kao broj proteklih rundi pre nego što se desi situacija da neki od robota ne bude u mogućnosti da obiđe sve događaje koji su mu dodeljeni, usled nedostatka energije. U cilju produženja životnog vremena sistema, od presudne je važnosti da se liste događaja formiraju na način kojim se minimizuje ukupna dužina robotskih putanja, kao i da njihove dužine budu izbalansirane radi ravnomerne raspodele opterećenja među robotima.

Centralizovani algoritmi kao pretpostavku za svoje funkcionisanje uzimaju globalno znanje, odnosno broj i položaje svih robota i svih događaja tokom date runde. Ove informacije sakuplja odabrani mrežni entitet kome se dodeljuje koordinatora, a obično je u pitanju neki od robota, ili bežična bazna stanica. Koordinator na osnovu globalnog znanja izvršava odgovarajući algoritam za raspodelu zadataka izvršioцима, nakon čega obaveštava robote o donesenim odlukama, odnosno o zadacima koji su im dodeljeni. Podrobnije bavljenje detaljima mehanizmima koji bi omogućili centralizovano prikupljanje informacija izlazi iz okvira tematike ove disertacije. Ukoliko se u tu svrhu

¹ Slučajni proces koji poseduje navedeno svojstvo naziva se Markovljevim procesom [Gar85].

koristi preplavlivanje mreže, ono samo po sebi nameće neprihvatljivo velike komunikacione troškove, pogotovo u slučaju mreža sa velikim brojem čvorova. Ipak, primena predloženih centralizovanih rešenja može biti opravdana u manjim mrežama, gde pitanje skalabilnosti manje dolazi do izražaja. Takođe, principi primenjeni prilikom implementacije centralizovanih algoritama izloženih u odeljcima 3.1 i 3.2 primenjeni su u rešenjima koja su po svojoj prirodi distribuirana i lokalizovana. Kod takvih rešenja, koja su prikazana u odeljku 3.3, roboti primenjuju modifikovane verzije centralizovanih algoritama, koristeći pri tome lokalno znanje i lokalnu komunikaciju tokom nekoliko iteracija.

Iscrpne simulacije koje su sprovedene i čiji rezultati su prikazani u poglavlju 4 pokazuju da nova rešenja koja su predstavljena u ovoj glavi u poređenju sa relevantnim postojećim rešenjima donose značajna poboljšanja u pogledu produženja životnog vremena sistema redukujući kako ukupne dužine robotskih putanja, tako i komunikacione troškove. Štaviše, razlika u performansama postaje sve izraženija sa povećanjem veličine mreže, što je indikator dobre skalabilnosti. Napokon, pokazuje se da novi algoritmi mogu da funkcionišu i u relaksiranim uslovima u pogledu distribucije senzorskih čvorova i dometa emitovanja, u odnosu na konkurentna rešenja.

3.1 Centralizovana rešenja problema m -dodela(m,n)

Ukoliko se usvoji pretpostavka da je tokom pojedinačne runde svakom robotu dozvoljeno dodeliti najviše jedan događaj, ovako formulisan problem naziva se m -dodela(m,n) i svodi se na formiranje parova u kompletnom težinskom bipartitnom grafu. Stoga, za početak će biti izložen pregled osnovne terminologije vezane za bipartitne grafove, zajedno sa notacijom koja će biti korišćena prilikom opisa algoritama.

Graf $G = (V, D)$ sačinjavaju skup čvorova V i skup ivica (parova čvorova) $D = \{(u, v) | u \in V \wedge v \in V\}$. Za čvorove u i v kaže se da su incidentni sa ivicom $e = (u, v) \in D$. G je bipartitni graf ako se skup čvorova V može podeliti na dva disjunktna podskupa E i R ($E \cap R = \emptyset \wedge E \cup R = V$), tako da se ni za jednu ivicu u skupu D čvorovi koji su incidentni sa njom ne nalaze u istom podskupu, odnosno da važi da je $D = \{(u, v) | u \in E \wedge v \in R\}$. Uparivanje (engl. *matching*) je skup ivica $M \subset D$ takav da je svaki čvor grafa V incidentan najviše sa jednom od ivica u skupu M . Problem uparivanja maksimalne kardinalnosti podrazumeva pronalaženje uparivanja M u kom je broj ivica $|M|$ maksimalan. Ukoliko čvor v nije incidentan ni sa jednom ivicom u skupu M , za njega se kaže da je neuparen, odnosno slobodan. Uparivanje je savršeno ukoliko nijedan čvor nije slobodan, odnosno ako je $|M| = |E| = |R|$.

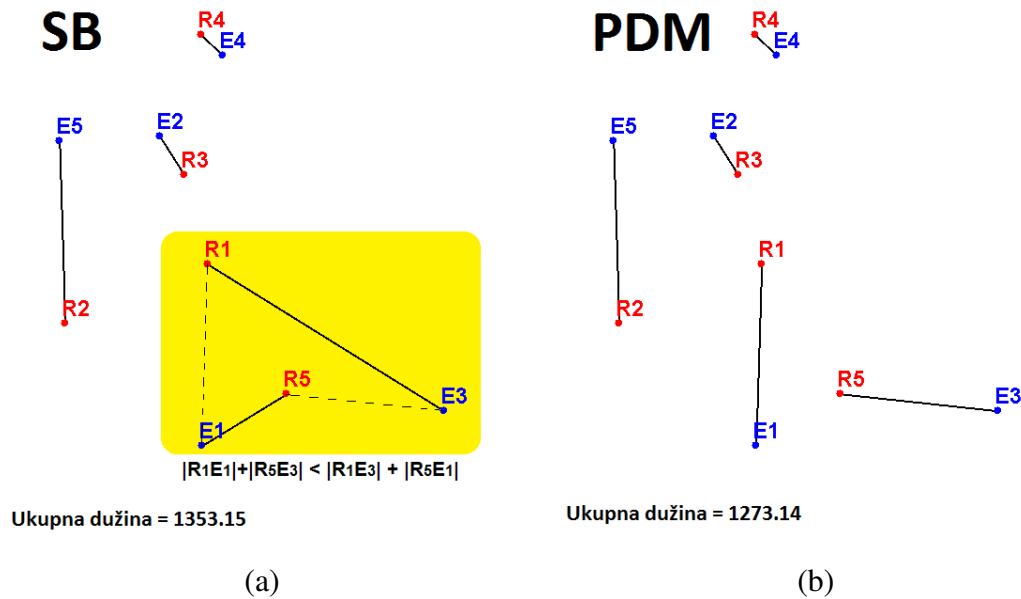
3.1.1 Centralizovano uparivanje zasnovano na udaljenostima (PDM algoritam)

Postavlja se pitanje opravdanosti razvoja novih algoritama koji rešavaju problem m -dodela(m, n), kada već postoji Mađarski metod koji garantovano daje optimalno rešenje. Problem vezan za Mađarski algoritam je njegova kompleksnost ($O(n^3)$), u kombinaciji sa značajnim memorijskim resursima potrebnim za njegovo izvršenje, budući da je neophodno sve vreme održavati matricu rastojanja koja je dimenzija $n \times n$. To je naročito problematično ako se zahteva da se algoritam izvršava na mikrokontroleru sa skromnim resursima kako po pitanju brzine, tako i po pitanju kapaciteta memorije. Stoga, kao osnova za nova rešenja korišćen je SB algoritam čija kompleksnost je $O(n^2)$ i čiji memorijski zahtevi su $O(n)$. Novo rešenje ne povećava dodatno vremensku složenost u odnosu na osnovni SB algoritam, a rezultati simulacija prikazani u poglavlju 4 pokazuju da su prosečne dužine ivica za manje od 1% duže u odnosu na one koje se dobijaju nakon izvršenja Mađarskog algoritma.

Algoritam za centralizovano uparivanje robota sa događajima zasnovano na njihovim međusobnim udaljenostima PDM (engl. *centralized Pairwise Distance-based Matching*), kreće od uparivanja dobijenog primenom algoritma stabilnih brakova SB [AGB10][GS62], opisanog u odeljku 1.6.1.2. Potom se dobijeno rešenje iterativno poboljšava razmenom partnera kod pojedinih parova robot-događaj, u cilju eliminisanja dugačkih ivica. Po izvršenju SB algoritma, svaki robot r uparen je sa događajem $E(r)$. Zatim se vrši sortiranje robota po kriterijumu dužina ivica između njih i događaja sa kojima su upareni. Drugim rečima, ivice $|rE(r)|$ se u pogledu dužina uređuju u nerastućem poretku, pošto su najduže ivice ujedno najbolji kandidati za razmenu partnera. Dakle, inicijalno rešenje dobijeno SB algoritmom poboljšava se razmatranjem svih parova ivica (A, B) i (C, D) , gde su A i B roboti, a C i D događaji. Ukoliko je ispunjen uslov da je $|AC| + |BD| > |AD| + |BC|$, parovi (A, B) i (C, D) se zamenjuju parovima (A, D) i (B, C) , budući da se na taj način ukupna dužina ivica smanjuje. Proces razmene partnera ilustrovan je slikom 3.1, gde parovi (R_1, E_3) i (R_5, E_1) razmenjuju partnere, u cilju smanjenja ukupne dužine ivica. Proces razmene partnera može biti ponavljan u više iteracija, sve dok postoje parovi koji ispunjavaju uslov za razmenu. Koraci tokom izvršenja PDM algoritma opisani su u algoritmu 9.

Teorema 1. Algoritam PDM se okončava u konačnom broju iteracija.

Dokaz. Među svim mogućim uparivanjima postoji ono sa minimalnom sumom, koje se dobija izvršenjem Mađarskog algoritma. U svakoj iteraciji, ukoliko postoje kandidati za razmenu partnera, prilikom razmene se totalna suma dužina ivica smanjuje. Pošto je skup svih ivica u polaznom bipartitnom grafu konačan, a ukupna dužina ivica se smanjuje svakom iteracijom, algoritam se okončava dostizanjem optimalnog rešenja, ili sub-optimalnog rešenja u kojem nema više parova ivica koje ispunjavaju uslov za razmenu partnera. \square

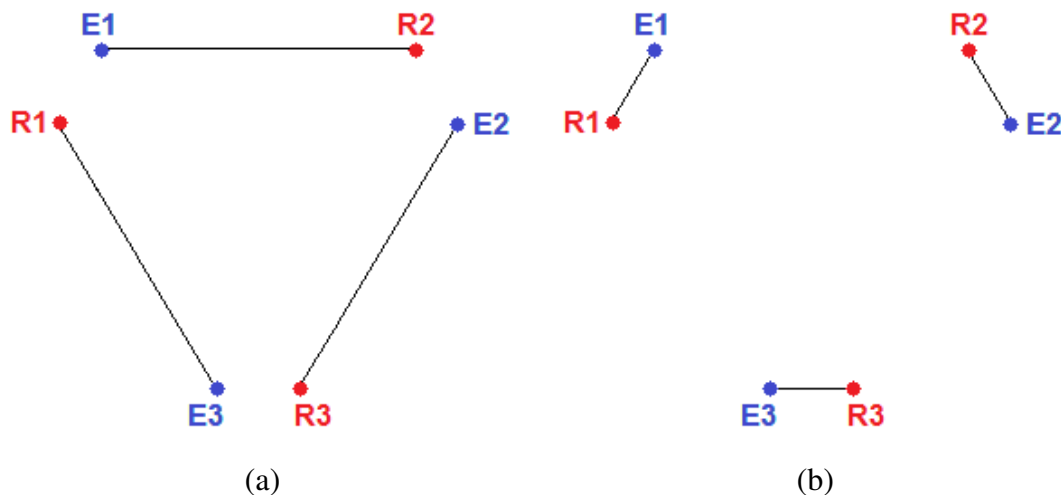


Slika 3.1: Proces razmene partnera u okviru PDM algoritma

Algoritam 9 Centralizovano uparivanje zasnovano na udaljenostima (*PDM*)**Ulaz:** skup robota R i skup događaja E **Izlaz:** skup uređenih parova M

- 1: izvršenjem SB algoritma se svaki robot r uparuje sa događajem $E(r)$
- 2: roboti se sortiraju tako da dužine ivica $|rE(r)|$ budu u nerastućem poretku
- 3: *promena* \leftarrow *true*
- 4: **while** *promena* **do**
- 5: *promena* \leftarrow *false*
- 6: **for** $r_1 = 1 \rightarrow n - 1$ **do**
- 7: **for** $r_2 = r_1 + 1 \rightarrow n$ **do**
- 8: **if** $|r_1E(r_1)| + |r_2E(r_2)| > |r_1E(r_2)| + |r_2E(r_1)|$ **then**
- 9: *temp* $\leftarrow E(r_1)$
- 10: $E(r_1) \leftarrow E(r_2)$
- 11: $E(r_2) \leftarrow temp$
- 12: *promena* \leftarrow *true*
- 13: **end if**
- 14: **end for**
- 15: **end for**
- 16: **end while**

Kao što je već napomenuto, rešenje koje se dobija izvršenjem *PDM* algoritma ne mora biti optimalno u pogledu minimizacije ukupne dužine ivica. Ovo je ilustrovano slikom 3.2, gde su čvorovi raspoređeni u nepravilni šestougao. Na slici 3.2 prikazana su uparivanja $M_1 = \{(R_1, E_3), (R_3, E_2), (R_2, E_1)\}$ (slika 3.2a) i $M_2 = \{(R_1, E_1), (R_2, E_2), (R_3, E_3)\}$ (slika 3.2b). Oba ova uparivanja predstavljaju lokalne optimume kada je *PDM* algoritam u pitanju. Uz to, u opštem slučaju dobijeno



Slika 3.2: Primeri uparivanja koja predstavljaju lokalne optimume

rešenje ne mora biti stabilno, pošto par robot-događaj koji je međusobno blizak može biti "žrtvovan" u cilju smanjenja ukupne dužine ivica.

Prilikom izvršenja *PDM* algoritma podrazumeva se da je $n = m$. Međutim, ako je $n > m$, tada je moguće dodati $n - m$ fiktivnih "besplatnih" događaja². Po izvršenju algoritma, roboti koji su upareni sa ovim fiktivnim događajima neće se pomerati tokom date runde.

3.1.2 Centralizovano uparivanje zasnovano na otpornostima (PRM algoritam)

U cilju balansirane raspodele opterećenja među robotima, ideja za poboljšanje *PDM* algoritma predstavljenog u prethodnom odeljku je implementacija mehanizma kojim se roboti kojima je preostalo manje energije uparuju sa bližim događajima (i obratno). Stoga, uveli smo novu metriku za dodelu težina ivicama između robota i događaja.

Neka je EN inicijalna energija robota. Ako se usvoji pretpostavka da je energija koju robot troši na kretanje linearno proporcionalna dužini putanje, inicijalna energija može biti interpretirana u smislu ukupne dužine koju robot može da pređe. Ako je u datom trenutku energija robota CR , ona se može normalizovati na vrednost $rem = CR/EN$ koja pripada intervalu $[0, 1]$ i koja predstavlja udeo preostale energije. Tada se vrednost $res = 1/rem$ može interpretirati kao otpornost. Ako robot treba da pređe rastojanje d , troškovi kretanja biće izraženi proizvodom rastojanja i otpornosti ($d \cdot res$). Ovo se koristi kao težina ivica u u modifikovanom algoritmu nazvanom *PRM* (engl. *centralized Pairwise Resistance-based Matching*). Njegov pseudo-kod istovetan je onome koji je prikazan u algoritmu 9, sa sledećim razlikama. U liniji br.1, prilikom izvršenja *SB*

²Dužine ivica između ovih događaja i svih robota iznose 0.

algoritma, poređenje $|re| < |r'e|$ treba zameniti sa $|re| \cdot res(r) < |r'e| \cdot res(r')$. Dalje, u liniji broj 8, uslov

if $|r_1E(r_1)| + |r_2E(r_2)| > |r_1E(r_2)| + |r_2E(r_1)|$ **then...**

treba zameniti uslovom

if $|r_1E(r_1)| \cdot res(r_1) + |r_2E(r_2)| \cdot res(r_2) > |r_1E(r_2)| \cdot res(r_1) + |r_2E(r_1)| \cdot res(r_2)$ **then...**

Nova metrika koja je nazvana otpornošću, po svojoj prirodi daje opravdanje takvom nazivu: roboti sa manje preostale energije više se "opiru" daljem kretanju. U scenarijima automatizovanog izvršavanja zadataka tokom više krugova, ovakva metrika je korišćena i za implementaciju drugih centralizovanih i distribuiranih rešenja sa balansiranjem opterećenja među robotima na osnovu preostale energije.

3.2 Centralizovana rešenja problema s -dodela(m, n)

Striktno uparivanje ne mora nužno predstavljati najbolji način za dodelu zadataka robotima kao izvršiocima. Naime, moguća je situacija da je za neki od robota suma rastojanja koja je potrebno prevaliti radi obilaska više mesta događaja manja nego što su direktna rastojanja tih istih događaja od robota sa kojima su upareni. Ovakva situacija već je elaborirana u odeljku 1.4.1 i ilustrovana slikom 1.6.

Sa druge strane, ukoliko je broj događaja veći od broja robota ($m > n$), tada je neophodno usvojiti princip da tokom pojedinačne runde roboti mogu da obiđu više događaja. Kao što je izloženo u odeljku 1.4.2, u ovom slučaju problem koji je nazvan s -dodela(m, n) postaje NP-težak. Budući da za ovakve probleme nije poznato da li postoje algoritmi koji daju optimalno rešenje u polinomijalnom vremenu, ovde će biti prikazane polinomijalne heuristike koje rešavaju različite varijante problema s -dodela(m, n). Rezultati simulacija koji su prikazani u poglavlju 4 potvrđuju bolju efikasnost predloženih rešenja u odnosu na od ranije poznata rešenja koja su opisana u odeljku 1.6.3.

3.2.1 Centralizovana dodela sekvenci događaja izvršiocima (SQD i SRD algoritam)

Algoritam *WPT* [WPT10] rešava slučaj kada je broj robota manji od broja događaja ($n < m$) tako što skup događaja deli na tačno n nepraznih grupa (klastera), a zatim uparuje robote sa klasterima. Novi centralizovani algoritam za dodelu sekvenci događaja izvršiocima označen skraćenicom *SQD* (engl. *Sequence Dispatch*) formiranje klastera vrši implicitno, pošto sam po sebi dozvoljava da roboti obiđu više događaja tokom iste runde. Uz to, ovakav pristup dozvoljava da pojedinim robotima uopšte ne budu dodeljeni događaji za obilazak ukoliko su previše udaljeni od njih, čak i u slučaju kad je broj robota manji od broja događaja. Ovakva situacija prikazana je na slici 3.3b.

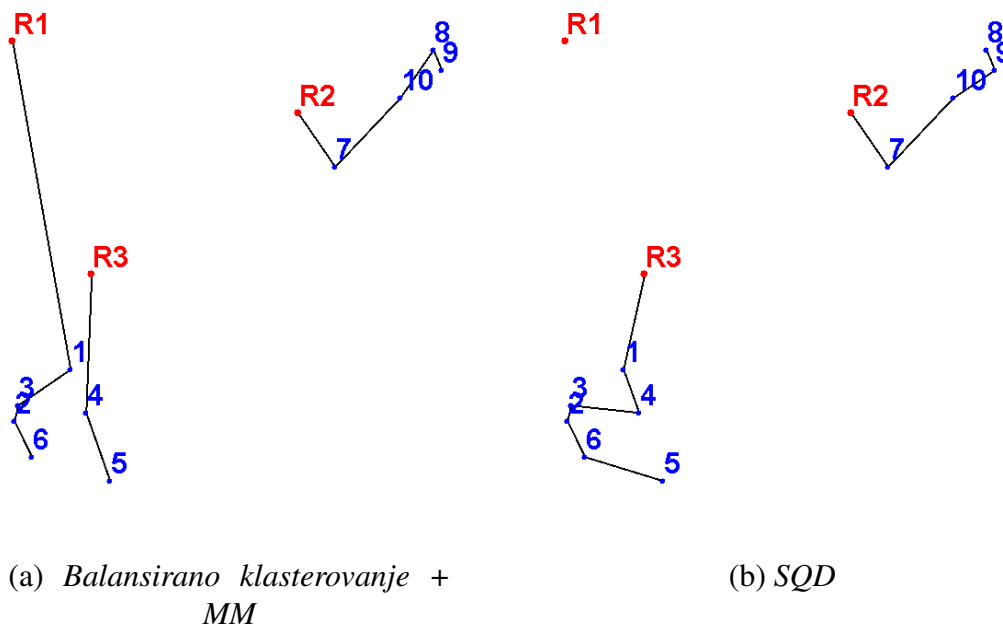
Isti pohlepni algoritam može da radi za proizvoljne vrednosti m i n , bez obzira na njihov odnos. Uz to, čak i kad je $n < m$, pojedini roboti mogu da obiđu više događaja. Algoritam funkcioniše na sledeći način. Na početku, svi događaji označavaju se kao slobodni, što znači da čekaju da im se dodeli izvršilac. U svakoj iteraciji pronalazi se par robot-slobodan događaj, za koji je međusobno rastojanje minimalno. Robot ažurira svoju poziciju na poziciju događaja³, a događaj se označava kao posećen. Iteracije se ponavljaju, sve dok svi događaji ne budu označeni kao posećeni. Opis *SQD* dat je u algoritmu 10.

³U ovom slučaju nije reč o stvarnoj fizičkoj poziciji na kojoj se robot trenutno nalazi, nego o poziciji na kojoj će se robot naći u budućnosti, prilikom obilaska selektovanog događaja.

Algoritam 10 Centralizovana dodela sekvenci događaja izvršiocima (*SQD*)**Ulaz:** skup robota R i skup događaja E **Izlaz:** skup sekvenci događaja DS

- 1: označi sve događaje kao slobodne
- 2: za svaki slobodan događaj e odredi najbližeg robota $r(e)$
- 3: **for** $i = 1 \rightarrow m$ **do**
- 4: pronajdi slobodan događaj e za koji je rastojanje $|er(e)|$ minimalno
- 5: uključiti događaj e u sekvencu događaja robota $r(e)$
- 6: ažuriraj poziciju robota $r(e)$ na poziciju e
- 7: označi događaj e kao posećen
- 8: ažuriraj udaljenosti svih slobodnih događaja e' u odnosu na novu poziciju $r(e)$
- 9: **if** $|e'r(e)| < |e'r(e')|$ **then**
- 10: $r(e') \leftarrow r(e)$
- 11: **end if**
- 12: **end for**

Nakon izvršenja *SQD* algoritma, dobijene sekvence događaja za obilazak mogu biti unapređene primenom *TSP* heuristike, ukoliko se ispostavi da se time dodatno skraćuju dužine robotskih putanja.



Slika 3.3: Primer dodele sekvenci događaja izvršiocima

Slika 3.3 prikazuje primer sa tri robota i deset događaja. Balansirano klasterovanje [WPT10] particioniše skup događaja na tri klastera, nakon čega se roboti uparuju sa klasterima pomoću *MM* algoritma, opisanog u odeljku 1.6.1.3 (slika 3.3a). Sa druge strane, *SQD* algoritam kreira sekvence događaja na takav način da nekima od robota ne mora biti dodeljen nijedan događaj za obilazak (robot R_1 na slici 3.3b). Time se

izbegava dugačka putanja koju bi robot R_1 morao da pređe da bi stigao do klastera koji mu je dodeljen u slučaju prikazanom na slici 3.3a.

Kao i u prethodnom odeljku, ponovo je moguće izvršiti modifikaciju algoritma tako da otpornosti budu uzete u obzir, sa ciljem balansiranja opterećenja. Promena se postiže zamenom $|er(e)|$ sa $|er(e)| \cdot res(r(e))$ u svim izrazima. Modifikovani algoritam za dodelu sekvenci događaja izvršiocima baziranu na otpornostima označava se skraćenicom *SRD* (engl. *Sequence Resistance-based Dispatch*).

Po svojoj prirodi, *SQD* i *SRD* su pohlepne heuristike koje kao takve u opštem slučaju proizvode sub-optimalna rešenja. Sa druge strane, jedna od prednosti im je jednostavnost realizacije budući da se mogu implementirati tako da se izvršavaju u vremenu $O(m(m+n))$.

3.2.2 Centralizovana dodela višestrukih izvršilaca događajima (GSQD algoritam)

U do sada razmatranim scenarijima, važila je pretpostavka da svaki događaj treba da bude posećen od strane jednog od robota. Međutim, priroda događaja može biti takva da se zahteva prisustvo više robota tokom iste runde. Stoga, algoritmi za uparivanje robota sa događajima se modifikuju na sledeći način. Ako događaj E_i zahteva da bude posećen od strane j različitih robota, kreira se j fiktivnih događaja $E_{i1}, E_{i2}, \dots, E_{ij}$ na istoj lokaciji, a zatim se izvršava algoritam za uparivanje kojim se svakom od ovih događaja dodeljuje različit robot.

Kada je reč o generalizovanom slučaju dodele sekvenci događaja izvršiocima, kreće se od pretpostavke da roboti mogu da obilaze više događaja tokom pojedinačne runde, a sa druge strane, neki od događaja mogu zahtevati da budu posećeni od strane više robota tokom iste runde. U tu svrhu uvodi se niz $C = (c_1, c_2, \dots, c_m)$ koji sadrži brojače poseta. Inicijalno, vrednost c_i je broj različitih robota koji trebaju da posete događaj E_i ($1 \leq c_i \leq n$). Matrica rastojanja $D[m \times n]$ definiše se vrednostima $d_{i,j} = |E_i R_j|$. Algoritam *SQD* modifikuje se tako da prilikom određivanja robota R_j kao izvršioca kojem se dodeljuje zadatak obilaska događaja E_i , brojač poseta c_i se dekrementira. Istovremeno, vrednost $d_{i,j}$ u matrici rastojanja postavlja se na ∞ , kako bi se osiguralo da isti robot ne može da bude dodeljen istom događaju tokom tekuće runde. Pozicija R_j menja se na poziciju E_i , a sve konačne vrednosti u istoj koloni matrice rastojanja ažuriraju se u skladu sa tim. Rezultujući modifikovani algoritam nazvan je generalizovanom dodelom sekvenci događaja izvršiocima, skraćeno (*GSQD*) (engl. *Generalized Sequence Dispatch*). Detaljniji opis sledi u algoritmu 11.

Slika 3.4 ilustruje *GSQD*, za $n = m = 8$. Zahtevani broj robota (c_i) koji treba da poseti događaj E_i prikazan je u zagradi. Na primer, E_5 koji zahteva da bude posećen od strane 3 robota, biva dodeljen robotima R_3, R_5 i R_7 .

Algoritam 11 Generalizovana dodela sekvenci događaja izvršiocima (*GSQD*)

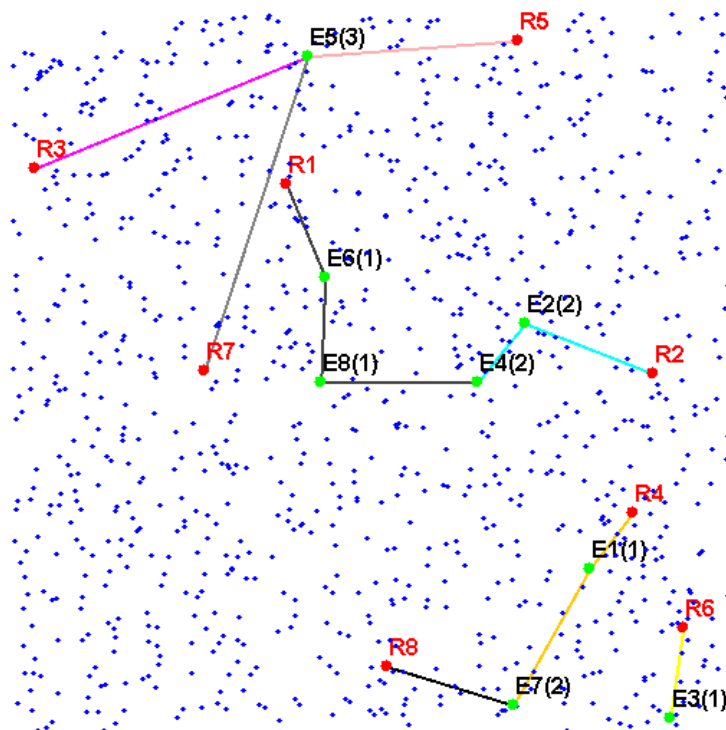
Ulaz: skup robota R , skup događaja E , niz brojača događaja C

Izlaz: skup sekvenci događaja DS

```

1: for  $i = 1 \rightarrow m$  do
2:   for  $j = 1 \rightarrow n$  do
3:      $d_{i,j} \leftarrow |E_i R_j|$ 
4:   end for
5: end for
6: while postoji bar jedna vrednost  $c_i \neq 0$  do
7:   pronađi  $(i, j)$  tako da je  $c_i \neq 0$  i  $d_{i,j}$  je minimalno
8:   uvrsti  $E_i$  u sekvencu događaja dodeljenih robotu  $R_j$ 
9:    $c_i \leftarrow c_i - 1$ 
10:   $d_{i,j} \leftarrow \infty$ 
11:  ažuriraj poziciju  $R_j$  na  $E_i$ 
12:  for  $i = 1 \rightarrow m$  do
13:    if  $d_{i,j} < \infty$  then
14:       $d_{i,j} \leftarrow |E_i R_j|$ 
15:    end if
16:  end for
17: end while

```



Slika 3.4: Primer GSQD

3.2.3 Centralizovana dodela sekvenci događaja izvršiocima sa vremenskim ograničenjima (SQDT algoritam)

Dosadašnja razmatranja bila su usredsređenja na minimizaciju kretanja, odnosno dužine robotskih putanja, bez vođenja računa o vremenskim ograničenjima. Međutim, kretanje robota, kao i obavljanje zadataka na mestima događaja predstavljaju akcije koje zahtevaju određeno vreme. Stoga uvodimo sledeće pretpostavke u ciju modelovanja vremenskih aspekata problema:

- Kretanje robota R_j obavlja se ravnomernom brzinom v_j . U skladu sa tim, vreme potrebno da robot prevali rastojanje d iznosi $\Delta t = d/v_j$.
- Vreme potrebno da robot R_j obavi zahtevanu akciju na mestu događaja E_i označava se sa $\tau_{i,j}$. Usvajanjem ove pretpostavke modelovana je heterogenost među robotima, pošto je na taj način robotima sa različitim resursima dozvoljeno da troše različita vremena za izvršenje istih zadataka.
- Ukupno trajanje pojedinačne runde je T .

Za svaki robot R_i , vrednost t_i predstavlja vreme potrebno da robot obiđe sve događaje koji su mu dodeljeni u datoj rundi i da na lokacijama događaja obavi potrebne zadatke. Na početku svake runde, ove vrednosti se inicijalizuju na 0. Modifikovana varijanta *GSQD* prikazana je u algoritmu 12 i naziva se centralizovanom dodelom sekvenci događaja izvršiocima sa vremenskim ograničenjima, što se označava skraćenicom *SQDT* (engl. *Sequence Dispatch with Timing constraints*).

Neophodno je napomenuti da je moguća situacija da vreme koje je potrebno da se obrade svi događaji prevazilazi trajanje runde T . U tom slučaju, svi neobrađeni događaji prebacuju se u narednu rundu koja počinje neposredno pošto i poslednji robot odradi zadatke koji su mu dodeljeni tokom tekuće runde.

Algoritam 12 Centralizovana dodela sekvenci događaja izvršiocima sa vremenskim ograničenjima (*SQDT*)

Ulaz: skup robota R , skup događaja E , inicijalne vrednosti c_i , v_j i $\tau_{i,j}$

Izlaz: skup sekvenci događaja DS

```

1: for  $i = 1 \rightarrow m$  do
2:   for  $j = 1 \rightarrow n$  do
3:      $d_{i,j} \leftarrow |E_i R_j|$ 
4:      $t_i \leftarrow 0$ 
5:   end for
6: end for
7: while postoji bar jedna vrednost  $c_i \neq 0$  do
8:   pronađi  $(i, j)$  takve da  $c_i \neq 0$  i  $t_j + d_{i,j}/v_j + \tau_{i,j}$  je minimalno
9:    $tmp \leftarrow t_j$ 
10:   $t_j \leftarrow t_j + d_{i,j}/v_j + \tau_{i,j}$ 
11:  if  $t_j > T$  then
12:    prebaci sve događaje  $E_k$  takve da  $c_k > 0$  u sledeću rundu
13:     $t_j \leftarrow tmp$ 
14:    naredi robotima da odrade zadatke koji su im dodeljeni
15:    okončaj tekuću rundu
16:  end if
17:  uvrsti  $E_i$  u sekvencu događaja dodeljenih robotu  $R_j$ 
18:   $c_i \leftarrow c_i - 1$ 
19:   $d_{i,j} \leftarrow \infty$ 
20:  ažuriraj poziciju  $R_j$  na  $E_i$ 
21:  for  $i = 1 \rightarrow m$  do
22:    if  $d_{i,j} < \infty$  then
23:       $d_{i,j} \leftarrow |E_i R_j|$ 
24:    end if
25:  end for
26: end while
27: naredi robotima da odrade zadatke koji su im dodeljeni

```

Moguće je zapaziti da je *SQDT* najopštiji od svih algoritama za dodelu sekvenci događaja robotima koji su predstavljeni u ovom poglavlju. *SQDT* se svodi na *GSQD*, ako su ulazni parametri takvi da je za vaki događaj E_i i svaki robot R_j , $\tau_{i,j} = 0$ i $v_i = V$, tako da $V \rightarrow \infty$. Pored toga, ovaj algoritam može biti dalje redukovana na *SQD*, ako za svaki događaj E_i važi da je $c_i = 1$.

3.3 Distribuirana rešenja problema s -dodela(m,n)

Za razliku od distribuiranog *WPT* algoritma [WPT10], koji se zasniva na podeli mreže na polja u okviru pravougaone rešetke, rešenje koje je ovde predstavljeno se ne oslanja na rešetkastu strukturu, čime se eliminiše uticaj dimenzija rešetke na proces dodele zadataka robotima. Kao i kod centralizovanih rešenja, ponovo se polazi od pretpostavke da su roboti vremenski sinhronizovani i da obavljaju zadatke u rundama jednakog trajanja. Tokom rundi, roboti posećuju mesta događaja pri čemu obavljaju zadatke, prikupljaju informacije o događajima koji su se u međuvremenu pojavili i iniciraju promene u informacionoj infrastrukturi koje su neophodne usled promene njihovih pozicija. Nova runda započinje procesom donošenja odluka, gde je svaki robot odgovoran za događaje koji su mu prijavljeni. Dodela zadataka obavlja se putem pregovora između robota, korišćenjem komunikacije u robotskoj mreži. Umesto pravougaonog kvoruma, koji je korišćen u distribuiranom *WPT* algoritmu, novi distribuirani algoritam koristi *iMesh* strukturu [LM12][LM15] kao mehanizam putem kojeg se informacije o događajima prosleđuju robotima. Zatim se sprovode aukcije u robotskoj mreži na sličan način kao u okviru k -SAAP protokola [MLMS13], u cilju daljeg unapređenja dobijenog rešenja.

Kada su u pitanju komunikacioni aspekti distribuiranih rešenja problema koordinacije robota koja će su detaljnije opisana u nastavku, koriste se sve tri komunikacione paradigme pomenute u odeljku 1.2.1. Komunikacija senzor-senzor i senzor-robot obavlja se na istovetan način, korišćenjem male snage i posledično malog dometa emitovanja. Prvenstveni razlog za ovakav pristup komunikaciji leži u težnji da se smanji potrošnja energije pri slanju poruka kod baterijski napajanih senzorskih čvorova. Takođe je važno napomenuti da se komunikacija male snage koristi za otkrivanja robota kao izvršilaca od strane senzorskih čvorova, kao i za dojavljivanje informacija o događajima robotima. Pri tome, roboti se tretiraju kao mrežni entiteti sa istim komunikacionim svojstvima kao i statični senzorski čvorovi. Sa druge strane, komunikacija robot-robot koristi veću snagu emitovanja, čime se postiže veći domet. Ovo je opravdano većom količinom energije kojom roboti raspolažu u odnosu na senzorske čvorove, te je moguće uspostaviti direktnu komunikaciju između robota, kako bi se izbeglo dodatno opterećenje statičke senzorske mreže.

3.3.1 Distribuirana dodela sekvenci događaja izvršiocima (MAD i MRD algoritam)

Novi distribuirani algoritam za dodelu sekvenci događaja izvršiocima označava se skraćenicom *MAD* (engl. *iMesh and Auction-based Dispatch*) i tokom svake runde odvija se na sledeći način.

1. Na početku runde kreira se *iMesh* struktura u statičkoj mreži. Čvorovi koji su detektovali događaje otkrivaju pozicije robota putem ukrštene pretrage, nakon čega im šalju zahteve za obilazak lokacija događaja. Tokom ove faze roboti formiraju liste događaja za koje su im stigli zahtevi za obilazak. Sive linije na slici 3.6a povezuju događaje sa robotima kojima su prijavljeni.
2. Roboti komuniciraju između sebe u cilju saznavanja informacija o svojim susedima u robotskoj mreži. To se postiže tako što svaki robot šalje pozdravnu poruku kojim vrši preplavlivanje robotske mreže. Ovo preplavlivanje je ograničeno na k hopova, te je stoga lokalnog karaktera. Pošto je struktura pozdravne poruke takva da sadrži poziciju i identifikaciju robota od kojeg je potekla, identifikaciju neposrednog pošiljaoca, kao i informaciju o broju hopova koji je do tada prešla, na osnovu ovih informacija svaki od robota može u potpunosti da rekonstruiše topologiju robotske mreže, zaključno sa robotima koji su k hopova udaljeni od njega. Ovo je od važnosti za rutiranje poruka koje se razmenjuju u robotskoj mreži tokom naredne, aukcione faze. Na posletku, treba naglasiti da u zavisnosti od međusobnih položaja robota, njihova mreža može, ali i ne mora biti povezana. Na slici 3.6a je prikazan primer sa deset robota, gde roboti $R_0, R_1, R_2, R_4, R_5, R_7, R_8$ i R_9 čine jednu, a roboti R_3 i R_6 drugu povezanu podmrežu. Broj i topologija povezanih mrežnih komponenti nemaju uticaj na to da li će algoritam biti uspešno izvršen, nego se eventualno odražavaju na kvalitet rešenja.
3. Tokom naredne faze, roboti obavljaju aukcije u vremenski ograničenim iteracijama. U svakoj iteraciji, robot R_i donosi odluku zasnovanu na lokalnom znanju, u koje spadaju lista događaja koji su prijavljeni robotu R_i i pozicije suseda R_i u robotskoj mreži, udaljenih do k hopova od njega. Opis koraka koje robot R_i obavlja tokom jedne iteracije prikazan je u algoritmu 13. Implementacija algoritma odrađena je u obliku konačnog automata, čiji dijagram stanja je prikazan na slici 3.5.
4. Aukcijska faza okončava se nakon što su svi roboti ispraznili liste zahteva i formirali sekvence događaja za obilazak. Finalni poredak kojim će događaji biti posećivani može da se razlikuje od redosleda kojim su događaji uvrštavani u liste, pošto potencijalno mogu biti poboljšani primenom *TSP* heuristike.
5. Po okončanju aukcijske faze, roboti se kreću duž utvrđenih putanja, posećujući događaje koji su im dodeljeni. Nakon što odrade zadatke, roboti ostaju na krajnjim pozicijama, tj. na pozicijama poslednjih događaja koje su obišli tokom tekuće runde i na tim pozicijama dočekuju početak naredne runde. Na slici 3.6b prikazane su putanje robota nakon izvršenja distribuiranog algoritma za dodelu zadataka izvršiocima *MAD*.

Algoritam 13 Pojedinačna iteracija koju izvršava robot R_i u okviru algoritma *MAD*

Ulaz: lista događaja prijavljenih robotu R_i , lista i pozicije svih k -hop suseda

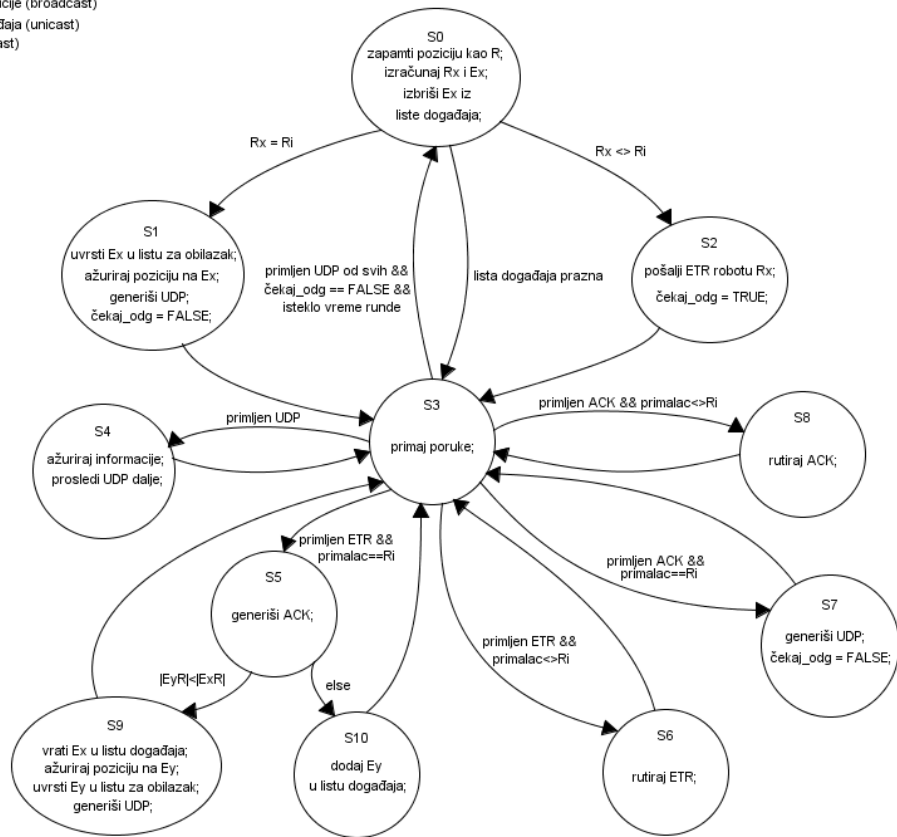
- 1: zapamti trenutnu poziciju: $R \leftarrow (x_i, y_i)$
- 2: **if** lista događaja je neprazna **then**
- 3: pronajdi robot R_x među k -hop susedima (uključujući sam robot R_i) i događaj E_x iz liste događaja, takve da je rastojanje $|E_x R_x|$ minimalno
- 4: izbriši E_x iz liste prijavljenih događaja
- 5: **if** $R_x = R_i$ **then**
- 6: ažuriraj poziciju R_i na poziciju E_x
- 7: uvrsti E_x u listu za obilazak
- 8: generiši *UDP* poruku sa informacijom o novoj poziciji
- 9: **else**
- 10: generiši *ETR* poruku sa informacijom o događaju E_x i pošalji je robotu R_x
- 11: **end if**
- 12: **end if**
- 13: **while** nisu primljene *UDP* poruke od svih k -hop suseda **do**
- 14: **if** primljena *UDP* poruka **then**
- 15: ažuriraj poziciju robota koji je generisao *UDP* poruku
- 16: prosledi *UDP* poruku dalje, ako je brojač hopova $< k$
- 17: **end if**
- 18: **if** primljena *ETR* poruka od robota R_y **then**
- 19: generiši *ACK* poruku i pošalji je robotu R_y
- 20: **if** $R_x = R_i$ i $|E_y R| < |E_x R|$ **then**
- 21: vrati E_x iz liste za obilazak u listu prijavljenih događaja
- 22: uvrsti E_y u listu za obilazak
- 23: ažuriraj poziciju R_i na poziciju E_y
- 24: generiši *UDP* poruku sa informacijom o novoj poziciji
- 25: **else**
- 26: dodaj E_y u listu prijavljenih događaja
- 27: **end if**
- 28: **end if**
- 29: **if** primljena *ACK* poruka **then**
- 30: generiši *UDP* poruku sa informacijom o staroj poziciji (poziciji R)
- 31: **end if**
- 32: **end while**
- 33: sačekaj da istekne vreme predviđeno za tekuću iteraciju

Tokom aukcijske faze, roboti komuniciraju razmenjujući tri različita tipa poruka:

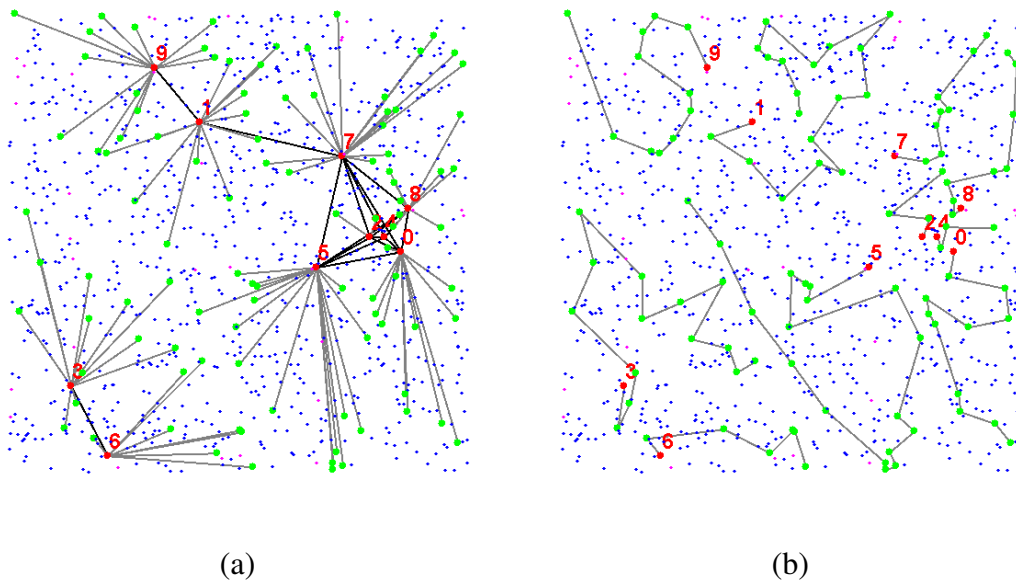
- *UDP* (Ažuriranje pozicije, engl. *update position*) prenosi informaciju o novoj poziciji robota. Prenosi se plavljenjem ograničenim na k hopova.
- *ETR* (Transfer događaja, engl. *event transfer*) generiše se kada robot donese odluku da prosledi događaj drugom robotu.
- *ACK* (Potvrda, engl. *acknowledgment*) generiše se po prijemu *ETR* poruke.

Tipovi poruka:

- UDP - ažuriranje pozicije (broadcast)
- ETR - transfer događaja (unicast)
- ACK - potvrda (unicast)



Slika 3.5: Implementacija aukcione faze MAD algoritma kao distribuiranog konačnog automata



Slika 3.6: Distribuirana dodela sekvenci događaja izvršiocima (MAD)

Razmatranje vezano za korišćenje otpornosti kao metrike u centralizovanim rešenjima važi i u distribuiranom slučaju. Distribuirani algoritam može se modifikovati zamenom izraza tipa $|R_x E_y|$ izrazima tipa $|R_x E_y| \cdot res(R_x)$, kako bi preko otpornosti bila uzeta u obzir i preostala energija robota. Modifikovani distribuirani algoritam za distribuiranu dodelu sekvenci događaja izvršiocima zasnovanu na otpornostima označava se skraćenicom *MRD* (engl. *iMesh and auction Resistance based Distributed dispatch*).

3.3.2 Distribuirana dodela višestrukih izvršilaca događajima (GAD algoritam)

U nastavku predstavljeno je proširenje distribuiranog algoritma u cilju dodele izvršilaca događajima koji zahtevaju prisustvo više robota tokom iste runde. Pretpostavke su identične onima koje su iznesene u odeljku 3.2.2. Robot R_i formira dve zasebne liste događaja. Prva sadrži događaje M_1, M_2, \dots, M_k koji zahtevaju da budu posećeni od strane više robota ($c_j > 1$ je zahtevani broj poseta za događaj M_j), dok druga lista sadrži događaje S_1, S_2, \dots, S_l , za koje je dovoljno da budu posećeni od strane samo jednog robota.

Neka je $h(R_i)$ ukupan broj k -hop suseda robota R_i . Tokom prve faze, roboti izvršavaju algoritam kojim se obrađuju događaji koji zahtevaju višestruke posete. Za svaki događaj M_j , ukoliko je $c_j \leq h(R_i) + 1$, robot R_i sortira svoje k -hop susede (uključujući sebe) po kriterijumu rastojanja od događaja M_j , a zatim upućuje zahteve za obilazak događaja M_j robotima koji zauzimaju prvih c_j mesta u sortiranoj listi. U suprotnom, zadatak ne može biti izvršen tokom tekuće runde, usled ograničenog broja robota u povezanoj pod mreži, ili usled nedovoljnog broja k -hop suseda⁴. Zahtevi se u tom slučaju šalju svim k -hop susedima, dok se preostalih $c_j - (h(R_i) + 1)$ poseta ostavljaju za sledeću rundu. Po prijemu zahteva, roboti uvrštavaju događaj u svoje liste za obilazak.

Tokom druge faze, događaji koji zahtevaju samo jednu posetu bivaju raspodeljeni pomoću *MAD* algoritma, kao što je opisano u prethodnom odeljku. Finalna lista događaja koje će robot R obići tokom trenutne runde sastoji se od događaja koji su u nju dodati tokom obe faze izvršenja algoritma. Putanja se dobija izvršenjem *TSP* heuristike sa listom događaja kao ulaznom instancom. Rezultujući modifikovani algoritam za distribuiranu dodelu višestrukih izvršilaca događajima označava se skraćenicom *c*

⁴Ukoliko je u pitanju nedovoljan broj k -hop suseda, može se pokušati sa adaptivnim testiranjem da li je broj $2k, 4k, 8k$, itd. suseda dovoljan za izvršenje zadatka.

Poglavlje 4

Poređenje performansi algoritama sa postojećim rešenjima

Prilikom istraživanja u oblasti bežičnih senzorskih i robotskih mreža, pogotovo kada su u pitanju mreže sa velikim brojem čvorova, uobičajena je praksa da se funkcionalnost i efikasnost predloženih rešenja proverava serijom simulacija u odgovarajućem softverskom simulacionom okruženju. Ovakav pristup nosi sa sobom višestruke prednosti:

- Nema potrebe za manipulisanjem velikim brojem hardverskih modula, što bi bilo skupo, komplikovano za rukovanje i zahtevalo bi ogromnu količinu vremena za pripremu pojedinačnog eksperimenta. Poređenja radi, tokom istraživanja koje je elaborirano u radu [LPMS09] koristili smo 42 bežična čvora tipa *WSN430*¹. Priprema eksperimenta koja uključuje programiranje čvorova, punjenje baterija i raspoređivanje čvorova, zajedno sa vremenom potrebnim za odvijanje i ponavljanje samog eksperimenta dovoljan broj puta zahteva značajnu količinu vremena koja se meri danima, pa čak i nedeljama, u zavisnosti od složenosti eksperimenta.
- Postavka simulacije je fleksibilna i rekonfigurabilna, što znači da je moguće drastično promeniti uslove izmenom svega nekoliko linija koda.
- Proces evaluacije je automatizovan i omogućava u kratkom vremenskom roku obavljanje velikog broja simulacija, kako bi dobijeni rezultati imali željenu statističku pouzdanost. Uz to, statistička obrada je takođe automatizovana.
- Relativno je jednostavno uporediti nova rešenja sa postojećim rešenjima drugih istraživača, budući da se u većini radova iz ove oblasti koristi ista metodologija evaluacije.

¹Ovo je skroman broj čvorova u odnosu na pretpostavljene simulacione scenarije o kojima će biti reči u ovom poglavlju.

Simulacije koje su sprovedene tokom rada na ovoj disertaciji i čiji najvažniji rezultati su prikazani u ovom poglavlju, sprovedene su korišćenjem simulacionog okruženja koje je kreirano i realizovano specijalno za ovu namenu. Prilikom simulacije algoritama usvojeni su simulacioni scenariji koji se odvijaju u idealizovanim uslovima. Idealizacija se odnosi na *cross-layer* modelovanje bežične mreže, uz ignorisanje nesavršenosti vezanih za bežičnu komunikaciju, koje bi mogle dovesti do gubitka i/ili grešaka u prijemu poruka. Realizacija takvih scenarija odrađena je u obliku simulacionih programa u programskom jeziku *C*, korišćenjem razvojnog okruženja *Eclipse*². Uz to, u istom razvojnom okruženju razvijena je i grafička aplikacija u programskom jeziku *Java*, kojom je omogućena vizualizacija dobijenih rešenja i čijim je korišćenjem generisana većina ilustracija koje su prikazane u ovoj disertaciji.

4.1 Simulacije protokola za otkrivanje izvršilaca

4.1.1 Postavka simulacionog okruženja

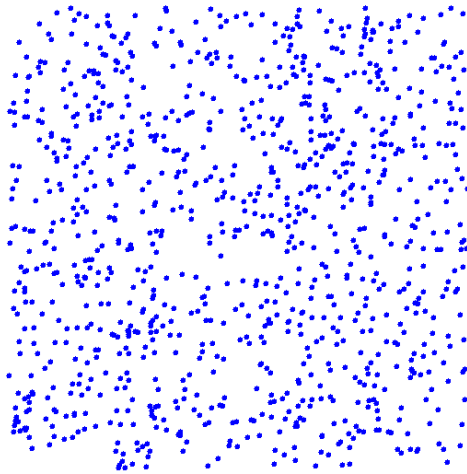
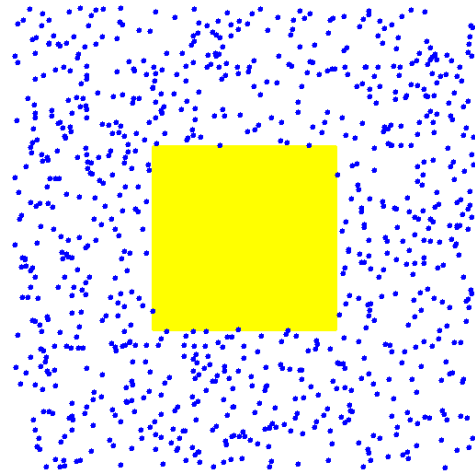
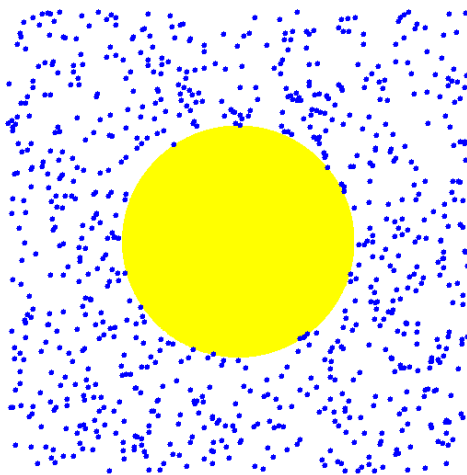
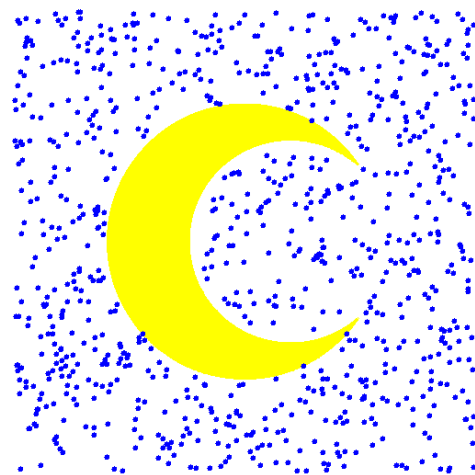
Ključni cilj simulacija čiji rezultati su prikazani u ovom odeljku je provera efikasnosti predloženih metoda i ujedno demonstracija funkcionalnosti koncepta. Za evaluaciju performansi komunikacionih protokola korišćeni su simulacioni programi, sa idealizovanim pretpostavkama kada je komunikacija u pitanju. Ovakva strategija primenjena je i u radovima gde su predstavljeni pravougaoni kvorum [SLJ08] i *iMesh* [LSS09]. Kroz simulacije poređene su performanse protokola za otkrivanje izvršilaca baziranih na strukturama pravougaonog kvoruma, *iMesh* i konačno novoj *vCell* strukturi predstavljenoj u poglavlju 2. Korišćenjem dva različita kriterijuma za rutiranje poruka (*MFR* rutiranje sa najvećim napretkom i *LDR* rutiranje sa minimalnim odstupanjem od prave propagacione putanje), dobijeno je ukupno pet protokola za poređenje: kvorum sa *MFR* rutiranjem, koji će u nastavku biti označen skraćenicom *Q_MFR*, kvorum sa *LDR* rutiranjem (skraćeno *Q_LDR*), *iMeshC* (*iMesh* sa *MFR* rutiranjem), *iMeshD* (*iMesh* sa *LDR* rutiranjem) i *vCell*.

Pretpostavljeno je da je mreža postavljena u okviru kvadratnog terena dimenzija³ 1000×1000 . Komunikacioni model je *UDG* graf, pri čemu je domet bežičnog emitovanja $R = 75$. Za početak, 1000 senzorskih čvorova raspoređeno je na slučajan način po celoj površini terena (slika 4.1a). U tom slučaju, prosečni stepen čvora⁴ iznosi oko 16, što se smatra gustom mrežom. Zatim su u mrežnu topologiju uvedene praznine različite geometrije, kao što je prikazano na slikama 4.1b, 4.1c and 4.1d). U ovim slučajevima korišćen je manji broj čvorova, srazmerno površini praznih oblasti, u cilju očuvanja približno podjednake gustine mreže.

²<https://eclipse.org/>

³U nastavku će radi opštosti dužina biti izražavana u neimenovanim jedinicama.

⁴Stepen čvora definiše se kao broj neposrednih suseda u komunikacionom grafu i kao takav predstavlja uobičajenu meru gustine mreže.

(a) *Uniformna raspodela čvorova*(b) *Praznina u obliku kvadrata*(c) *Praznina u obliku kruga*(d) *Praznina u obliku polumeseca*Slika 4.1: *Mrežne topologije*

Tokom simulacija, broj robota povećavan je od 1 to 10, pri čemu je na slučajan način generisano $100 \times 4 = 400$ različitih rasporeda čvorova, za svaki broj robota. Po generisanju mrežne topologije, u okviru pojedinačne simulacije vršena je evaluacija komunikacionih troškova tokom faza formiranja informacionog direktorijuma, otkrivanja izvršilaca i održavanja direktorijuma, o čemu je bilo reči u odeljku 2.4. Takođe, posmatran je procenat uspeha u pronalaženju najbližeg izvršioca. Kako bi simulacije bile sveobuhvatne, simulaciona procedura obavlja se tako što se redom svakom od statičkih senzorskih čvorova dodeljuje uloga čvora koji obavlja pretragu. Simulaciona strategija opisana je u algoritmu 14.

Algoritam 14 Simulacije protokola za otkrivanje izvršilaca**Ulaz:** skup senzorskih čvorova S **Izlaz:** prosečan broj poruka prilikom formiranja direktorijuma i pronalaženja izvršilaca, procenat uspeha u pronalaženju najbližeg izvršioca

```

1: for  $BR\_ROBOTA = 1 \rightarrow 10$  do
2:   for  $SIMULACIJA = 1 \rightarrow 100$  do
3:     generiši topologiju mreže  $T$  na slučajan način i memoriši je
4:     for  $PROTOKOL = kvorum\_MFR \rightarrow vCell$  do
5:       nad polaznom topologijom  $T$  konstruiši informacioni direktorijum  $D$  i memoriši ga
6:       for  $i = 1 \rightarrow BR\_CVOROVA$  do
7:         kreni od topologije  $T$  sa informacionim direktorijumom  $D$ 
8:          $R_j \leftarrow pretraga(S_i)$ 
9:         if  $R_j$  koji je otkriven u procesu pretrage je najbliži događaju  $S_i$  then
10:          inkrementiraj brojač pronalazaka najbližeg izvršioca
11:        end if
12:        izvrši odjavu robota  $R_j$ 
13:        promeni poziciju robota  $R_j$  na poziciju događaja  $S_i$ 
14:        izvrši prijavu robota  $R_j$ 
15:      end for
16:    end for
17:  end for
18: end for

```

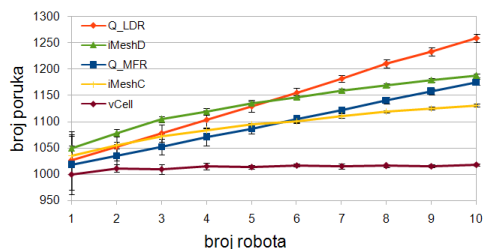
Napomene:

- Prilikom konstrukcije direktorijuma u liniji 5, pretrage u liniji 8, odjave u liniji 12 i prijave u liniji 14 izvršavaju se odgovarajući algoritmi u sklopu komunikacionog protokola čija evaluacija se vrši u datoj iteraciji.
- Za svaki od gore pomenutih algoritama beleži se ukupan broj generisanih poruka, koji se na kraju deli sa brojem čvorova i brojem simulacija, čime se dobijaju prosečne vrednosti komunikacionih troškova.
- Simulacioni program ponavlja se 4 puta, za različite mrežne topologije prikazane na slici 4.1.
- Tabele sa kompletnim rezultatima simulacija koji su prikazani na graficima u odeljku 4.1 date su u dodatku A.

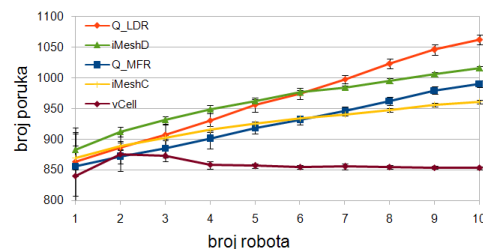
4.1.2 Komunikacioni troškovi

Grafici na slikama 4.2, 4.3 i 4.4 prikazuju komunikacione troškove tokom različitih faza protokola za pronalaženje izvršilaca, u okviru 95%–tnih statističkih intervala poverenja.

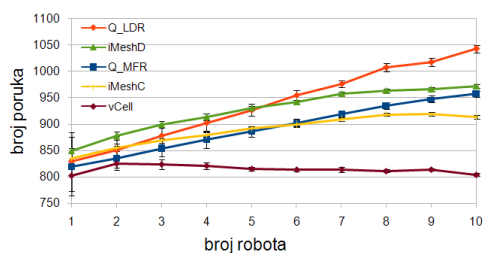
Tokom faze formiranja informacionog direktorijuma, protokoli bazirani na kvorumu i *iMesh* prvo obavljaju proces formiranja tabela susednosti putem slanja i prijema pozdravnih poruka i stoga imaju iste inicijalne troškove koji iznose m . Troškovi koji slede nakon toga vezani su za sam proces formiranja informacionog direktorijuma. Prilikom formiranja kvoruma, troškovi rastu linearno sa povećanjem broja robota (n), pošto svaki novi robot troši približno jednak broj poruka za razglašavanje svoje pozicije duž pravca istok-zapad⁵. Protokoli bazirani na *iMesh* strukturi imaju nešto veće inicijalne troškove u odnosu na kvorum, pošto se registracione poruke umesto u 2 prosleđuju u 4 pravca, ali imaju manju stopu rasta usled lokalizovane prirode protokola. Sa druge strane, u okviru *vCell* protokola, otkrivanje suseda integrisano je sa formiranjem direktorijuma, što rezultuje time da su ukupni troškovi tokom ove faze neznatno viši od m i za razliku od ostalih protokola, nemaju osetan rast sa povećanjem broja robota. Simulacioni rezultati za komunikacione troškove tokom formiranja direktorijuma prikazani su u tabelama A.1 do A.4 i ilustrovani su graficima na slici 4.2.



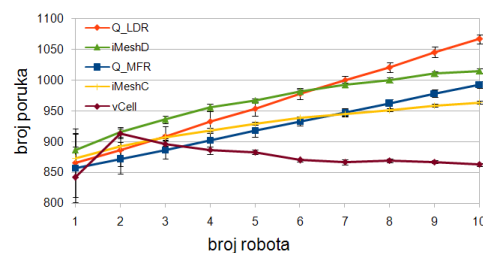
(a) Uniformna raspodela čvorova



(b) Praznina u obliku kvadrata



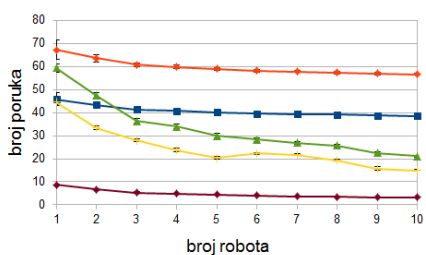
(c) Praznina u obliku kruga



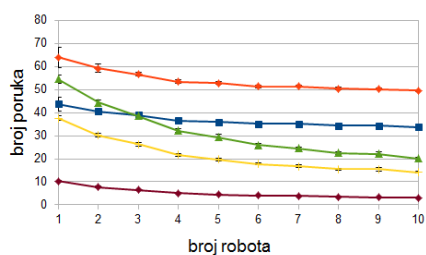
(d) Praznina u obliku polumeseca

Slika 4.2: Komunikacioni troškovi tokom faze konstrukcije direktorijuma

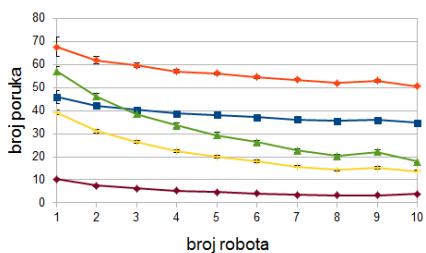
⁵Ovde dolazi do izvesnih odstupanja u slučaju prisustva praznina u mrežnoj topologiji, kao i u zavisnosti od toga da li je praznina konveksna ili konkavna.



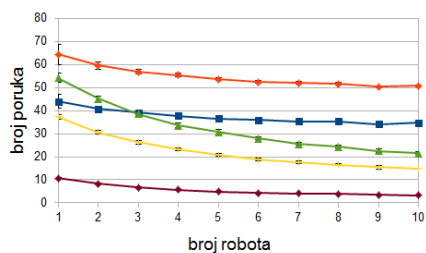
(a) Uniformna raspodela čvorova



(b) Praznina u obliku kvadrata

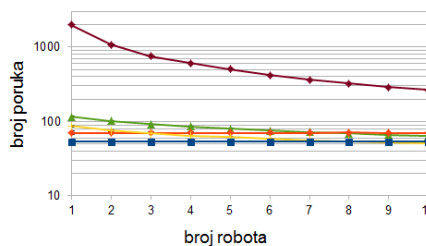


(c) Praznina u obliku kruga

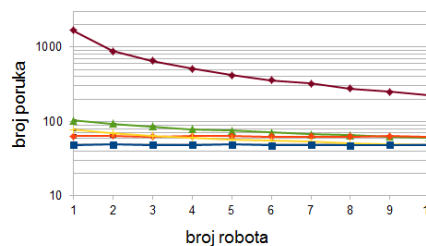


(d) Praznina u obliku polumeseca

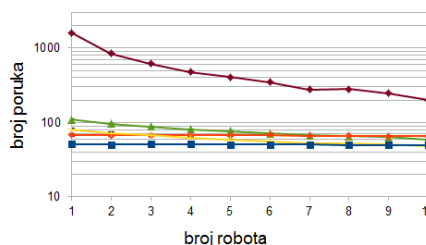
Slika 4.3: Komunikacioni troškovi tokom faze otkrivanja i obaveštavanja izvršilaca



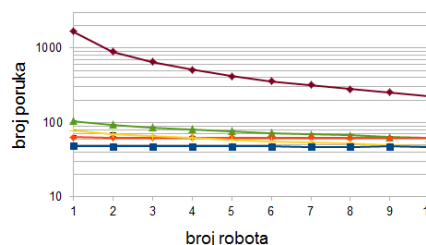
(a) Uniformna raspodela čvorova



(b) Praznina u obliku kvadrata



(c) Praznina u obliku kruga



(d) Praznina u obliku polumeseca

Slika 4.4: Komunikacioni troškovi tokom faze održavanja direktorijuma

Tokom faze otkrivanja i obaveštavanja izvršilaca, *vCell* proizvodi značajnije niže komunikacione troškove u poređenju sa ostalim protokolima i to oko 4 puta niže u odnosu na bolji protokol baziran na *iMesh* strukturi (*iMeshC*) i 4 – 10 puta niži u odnosu na bolji protokol baziran na kvorumu (*Q_MFR*). Ovakva efikasnost proističe iz

dva razloga: odsustva operacije pretrage koja je u *vCell* protokolu redundantna, uz rutiranje zahteva optimalnom putanjom, što podrazumeva minimalan mogući broj hopova. Komunikacioni troškovi tokom faze otkrivanja i obaveštavanja izvršilaca prikazani su u tabelama A.5 do A.8 i ilustrovani su graficima na slici 4.3.

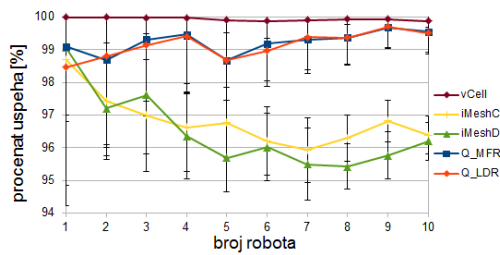
Sa druge strane, *vCell* nameće najveće troškove kada je u pitanju održavanje. Tokom faze održavanja, obavljaju se tri operacije zasnovane sa plavljenju. Raskorak u veličini komunikacionih troškova između *vCell* i ostalih protokola značajno se smanjuje sa povećanjem broja robota. Ovo može biti problem onda kada je potrebno često obavljati operacije ažuriranja informacione strukture i tada neki od protokola sa nižim troškovima održavanja može predstavljati bolji izbor. Komunikacioni troškovi tokom faze održavanja direktorijuma prikazani su u tabelama A.9 do A.12 i ilustrovani su graficima na slici 4.4.

Grafici na slikama 4.2-4.4 pokazuju da prisustvo i geometrija praznina nemaju značajniji uticaj na odnos između komunikacionih troškova kod posmatranih protokola. Ovo opažanje odnosi se na sve tri faze životnog veka protokola za otkrivanje izvršilaca.

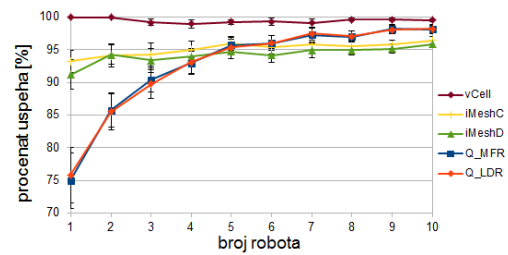
4.1.3 Procenat uspeha u pronalaženju najbližeg izvršioca

Dalje, analiziran je procenat uspeha u pronalaženju najboljeg, to jest najbližeg izvršioca, za svaki od pet posmatranih protokola. Kada je u pitanju energetska efikasnost, ovo je veoma bitno pitanje, ako se ima u vidu težnja za minimizacijom kretanja robota. Kretanje samo po sebi troši nekoliko redova veličine više energije u odnosu na komunikaciju koja se obavlja na istom rastojanju.

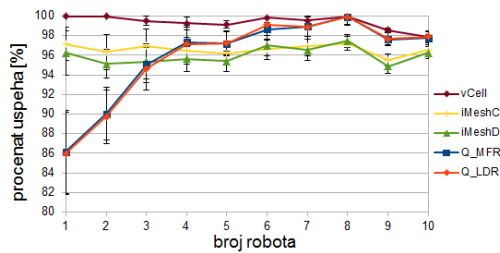
Rezultati prikazani na slici 4.5 i u tabelama A.13 do A.16 pokazuju da *vCell* postiže najbolji procenat uspeha u pronalaženju najbližeg robota od svih posmatranih protokola. Kada je raspodela čvorova uniformna, ili kada je u topologiji prisutna praznina konvexnog oblika, ovaj procenat iznosi blizu 100%, dok je nešto niži u prisustvu konkavne praznine u obliku polumesaca, ali je i u tom slučaju iznad 95%. Generalno, praznine u topologiji značajno smanjuju procenat uspeha za ostala četiri protokola, pogotovo u slučaju malog broja robota ($n < 3$) i to je upravo situacija kada geometrija praznine dolazi do izražaja. Konkavan oblik pokazuje se kao najproblematičniji, pošto u pojedinim slučajevima može da dovede do zaustavljanja propagacije registracionih poruka. Sa druge strane, "glatka" konvexna praznina u obliku kruga ima tendenciju da deformiše propagacione putanje, umesto da ih blokira. Sve u svemu, *vCell* ima najveći procenat uspeha, što je pogotovo primetno u slučajevima kada je broj robota mali. Konkretno, za $n = 1$, *vCell* pokazuje do 30% veći procenat u odnosu na protokole bazirane na kvorumu, odnosno do 15% u odnosu na protokole zasnovane na *iMesh* strukturi, u zavisnosti od strukture mreže. Razlika u performansama smanjuje se sa povećanjem broja robota.



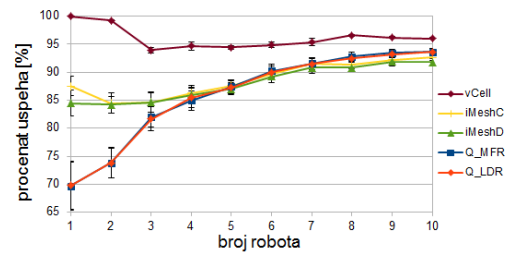
(a) Uniformna raspodela čvorova



(b) Praznina u obliku kvadrata



(c) Praznina u obliku kruga

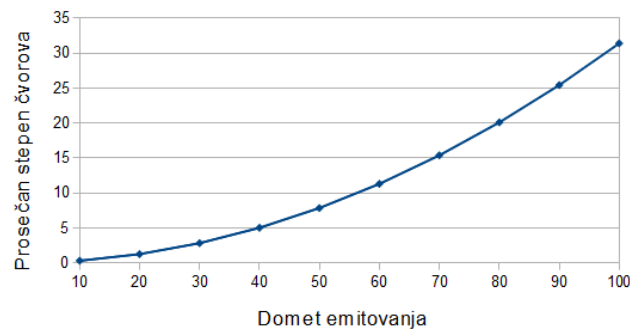


(d) Praznina u obliku polumeseca

Slika 4.5: Procenat uspeha u pronalaženju najbližeg izvršioca za konstantan dometa emitovanja $R = 75$

4.1.4 Uticaj gustine mreže na performanse protokola za otkrivanje izvršilaca

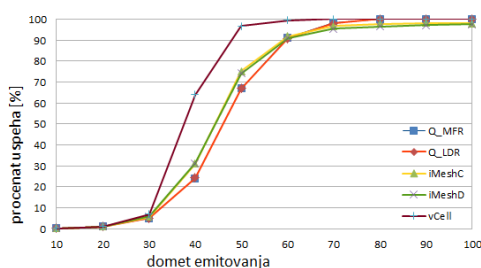
U ovom odeljku biće reči o uticaju gustine mreže na performanse protokola za otkrivanje izvršilaca. Najpogodniji način za izražavanje gustine mreže je prosečan stepen čvorova koji će ovde biti označen sa δ . U mrežama koje su modelovane kao *JDG* sa ravnomernom raspodelom čvorova, gustina se može približno izraziti u funkciji dometa emitovanja R kao $\delta = \frac{R^2 \cdot \pi}{a^2} \cdot m$, gde je a dužina spoljašnje ivice terena. U pitanju je kvadratna funkcija čiji grafik je prikazan na slici 4.6.



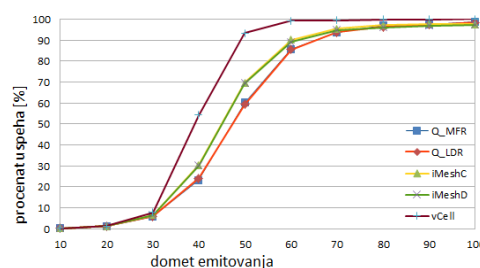
Slika 4.6: Stepen čvorova u zavisnosti od dometa emitovanja

Grafici prikazani na slici 4.7 na osnovu podataka iz tabela A.17 do A.20 ilustruju procenat uspeha u nalaženju najbližeg izvršioca. Rezultati su dobijeni dodatnom serijom simulacija, gde je broj robota konstantan ($n = 5$), a domet emitovanja se menja od 10 do 100 u koracima po 10. Na graficima uočavaju se tri karakteristične zone, na osnovu kojih je izvršena klasifikacija mreža u tri kategorije:

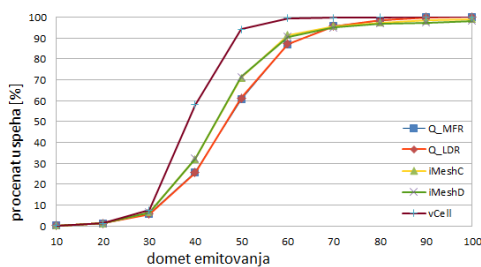
- **Retke mreže** ($\delta < 5$). Mrežni graf sastoji se od većeg broja podgrafova koji nisu međusobno povezani. Svi protokoli za otkrivanje izvršilaca postižu nizak procenat uspeha (ispod 10%), budući da im je delovanje ograničeno na podmreže u kojima se nalaze roboti.
- **Mreže srednje gustine** ($5 < \delta < 10$). U ovom slučaju postoji nekoliko većih podmreža koje nisu međusobno povezane. Dolazi do značajnijeg poboljšanja performansi protokola, koje su međutim i dalje zavisne od veličine podmreža. Slika 4.7 pokazuje da *vCell* ostvaruje značajno bolji uspeh u ovoj oblasti u odnosu na druge protokole (i do 40%).
- **Guste mreže** ($\delta > 10$). Najveći broj čvorova ulazi u sastav povezanog mrežnog grafa. Procenat uspeha je visok (iznad 90%) za sve posmatrane protokole.



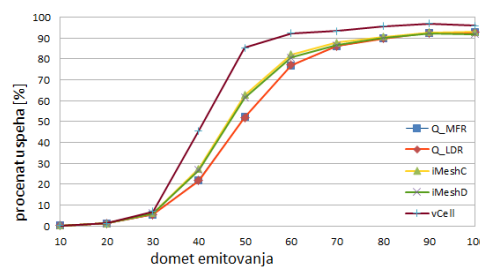
(a) Uniformna raspodela čvorova



(b) Praznina u obliku kvadrata



(c) Praznina u obliku kruga

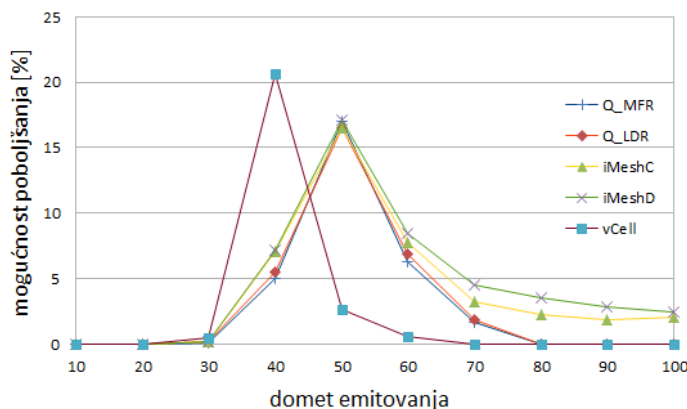


(d) Praznina u obliku polumeseca

Slika 4.7: Procenat uspeha u pronalaženju najbližeg izvršioca za konstantan broj robota $n = 5$

U dosadašnjim razmatranjima, akcenat je bio na procentu uspeha u pronalaženju najboljeg (najbližeg) robota. Međutim, interesantno je razmotriti i procenat uspeha u

pronalaženju bilo kog (ne nužno najbližeg) robota. Stoga smo definisali mogućnost poboljšanja kao razliku između ove dve vrednosti, dakle između procenta uspeha u pronalaženju bilo kog robota i procenta uspeha u pronalaženju najbližeg robota. Grafik prikazan na slici 4.8 prikazuje na koji način ova vrednost zavisi od dometa emitovanja u mreži sa uniformnom raspodelom čvorova. Moguće je uočiti da je mogućnost poboljšanja najveća u mrežama srednje gustine. Mehanizam pomoću kojeg bi poboljšanje moglo biti postignuto podrazumeva uključivanje nove komunikacijske paradigme, a to bi bila komunikacija robot-robot sa većim dometom emitovanja i primenu odgovarajućih principa koordinacije u robotskoj mreži, kao što su lokalne aukcije [MLMS13].



Slika 4.8: *Mogućnost poboljšanja nalaženja najbližeg izvršioca u mreži sa uniformnom raspodelom čvorova*

4.1.5 Diskusija o životnom vremenu sistema

Kao što je već rečeno u uvodnom poglavlju, u ovom istraživanju pojam energetske efikasnosti posmatran je iz dve različite perspektive. Prvo pitanje je kako produžiti životno vreme statičkih senzorskih čvorova, a drugo je kako prolongirati trenutak kada će roboti iscrpeti energiju usled kretanja. Kada se posmatra životno vreme čitavog sistema, ova dva pitanja međusobno se prepliću i kao takva trebalo bi da budu zajedno razmatrana. Kako pokazuju rezultati naših simulacija, procenat uspeha u pronalaženju najbližeg, ili u lošijoj varijanti bilo kog robota u velikoj meri zavisi od gustine mreže, koja je diktirana dometom emitovanja. Ako se usvoji generalni model $P(R) = R^4 + 2 \cdot 10^8$ za snagu potrebnu za bežično emitovanje sa dometom R , koji je izložen u radu [RM98], može se proceniti broj poruka koji pojedinačni čvor može da pošalje tokom svog životnog vremena, pod uslovom da je poznat kapacitet izvora napajanja (baterije) i da je domet emitovanja podešen na vrednost koja obezbeđuje željenu gustinu mreže. Povećanje dometa emitovanja smanjuje broj poruka koje su statički čvorovi u stanju da pošalju, ali sa druge strane pomaže produženju životnog vremena robota, time što dovodi do skraćivanja putanja koje roboti prelaze. Stoga, ako

su na raspolaganju odgovarajući modeli za izražavanje potrošnje energije kojima se opisuje zavisnost potrebne snage emitovanja od željenog dometa za statične čvorove i zavisnost utrošene energije od pređenog rastojanja za robote, moguće je podesiti parametre sistema u cilju maksimizacije životnog vremena celog sistema. Ovo je pitanje koje nije detaljnije izučavano tokom rada na ovoj disertaciji i predstavlja temu za buduća istraživanja.

4.1.6 Zaključci u vezi sa protokolima za otkrivanje izvršilaca

Simulacije čiji rezultati su prikazani u ovom odeljku prikazuju ponašanje protokola za otkrivanje izvršilaca u bežičnim mrežama senzora i robota sa različitim svojstvima u pogledu geometrije i distribucije čvorova. Performanse novog protokola nazvanog *vCell* koji na lokalizovan način kreira informacionu strukturu nalik na Voronoi dijagram poređene su sa postojećim protokolima (pravougaoni kvorum i *iMesh*). Posmatran je slučaj kada se pojavljuje pojedinačni događaj koji biva detektovan od strane statičkog senzorskog čvora i koji zahteva reakciju od strane jednog od robota. Proces pretrage omogućava da robot bude informisan o postojanju i lokaciji događaja. Pošto kretanje robota izaziva promene u mrežnoj strukturi, uvedeni su novi algoritmi koji ponovo uspostavljaju informacionu strukturu nakon što jedan od robota promeni lokaciju, a zatim je putem simulacija vršena evaluacija njihovih komunikacionih troškova.

Rezultati simulacija pokazuju da u pogledu komunikacionih troškova potrebnih za uspostavljanje informacione strukture, *vCell* ima niže inicijalne troškove u odnosu na druge protokole, budući da je kod njega faza otkrivanja suseda uklopljena u sam proces formiranja direktorijuma. Takođe, troškovi vezani za fazu otkrivanja izvršilaca najniži su kod *vCell* protokola, stoga što je u ovom slučaju proces pretrage redundantan, a uz to struktura komunikacionog stabla obezbeđuje optimalne putanje za prosleđivanje zahteva robotima. Međutim, njegova mana je to što proizvodi značajno veće troškove tokom faze održavanja, u odnosu na ostale protokole.

Kada je u pitanju postotak uspeha u pronalaženju najboljeg (najbližeg) robota, *vCell* protokol u gustim mrežama pokazuje superiorne performanse koje se ogledaju u procentu od skoro 100%. Prisustvo i geometrija praznina u mrežnoj topologiji utiče na procenat uspeha protokola baziranih na pravougaonom kvorumu i *iMesh* strukturi značajno više nego kada je u pitanju *vCell*, koji postojano postiže procenat veći od 95%. Gustina međutim ima presudan uticaj u ovom pogledu, usled problema sa povezanošću komunikacionog grafa u retkim mrežama i mrežama srednje gustine.

Simulacije takođe pokazuju da protokoli koji koriste rutiranje zasnovane na minimalnom odstupanju od prave propagacione putanje (*Q_LDR* i *iMeshD*) imaju veće komunikacione troškove i postižu niže procenat uspeha u odnosu na protokole koji koriste rutiranje sa maksimalnim napretkom (*Q_MFR* i *iMeshC*), bez obzira na

topologiju mreže. Time je pokazano da nepravilan oblik propagacionih putanja nema negativan uticaj na performanse protokola, čime je opovrgnuta pretpostavka koju smo izneli u prethodnom radu [LM12].

Formiranje *vCell* informacione strukture zasnovano je na lokalizovanom plavljenju, za razliku od pravougaonog kvoruma i *iMesh* strukture koji koriste geografske informacije pri donošenju odluka vezanih za rutiranje. Stoga, korišćenje skupog lokalizacionog hardvera kao što je *GPS* nije neophodno za formiranje same *vCell* strukture, nego u cilju informisanja robota o tačnoj lokaciji događaja. Imajući ovo u vidu, implementacija protokola baziranih na jeftinim lokalizacionim metodama ostaje izazov koji će biti predmet budućih istraživanja.

Na početku poglavlja bilo je reči o tome da simulaciono okruženje koje je korišćeno tokom ovog istraživanja obavlja simulacije uz pretpostavku da su uslovi u kojima se komunikacija idealizovani. Dalji koraci evaluacije ponašanja protokola mogli bi da idu u pravcu korišćenja nekog od simulatora kao što je *NS-2*⁶ koji uzima u obzir realističnije pretpostavke, modelujući *PHY* i *MAC* mrežne slojeve. Na ovaj način mogu biti uzete u obzir situacije koje se javljaju u realnosti kao što su preslušavanja i kolizije pri emitovanju. Konačan i najkredibilniji metod evaluacije opisanih protokola bi bio putem sprovođenja eksperimenata i emulacija korišćenjem pravih senzorskih čvorova. Ovo može biti postignuto korišćenjem velikih eksperimentalnih okruženja kao što je *SensLab/IoT-lab*⁷, ili usvajanjem neke od emulacionih strategija koje omogućavaju obavljanje eksperimenata sa manjim brojem čvorova, kao što je opisano u radovima [LPMS09] i [PRM⁺12].

Konačno, imajući u vidu da otkrivanje izvršilaca treba da bude stavljeno u širi kontekst koordinacije među robotima u slučaju pojave više konkurentnih događaja, koja se obavlja u više krugova, postavlja se pitanje izbora protokola koji će najbolje poslužiti svrsi. U ovom odeljku pokazano je da *vCell* ima superiorne performanse u odnosu na ostale protokole, ali i visoke troškove kada je u pitanju održavanje. Budući da tokom svake runde dolazi do relokacije svih robota, to praktično znači da je početkom svake runde potrebno formirati informacioni direktorijum iz početka. Budući da se radi o scenariju koordinacije koja se obavlja u više krugova, pri čemu se položaji statičkih senzorskih čvorova ne menjaju između dva kruga, visoki inicijalni troškovi procesa otkrivanja suseda kod *iMesh* protokola pojavljuju se samo jednom i to u prvom krugu. Stoga, u svakom narednom krugu formiranje *iMesh* strukture kreće od poznate topologije mreže, ako ne računamo robote, koji ipak moraju da pošalju pozdravne poruke pošto su promenili pozicije tokom prethodnog kruga. Samim tim, troškovi formiranja direktorijuma drastično opadaju (manji su za m u odnosu na one prikazane u odeljku 4.1.2), te stoga *iMesh* predstavlja bolji izbor od *vCell* protokola kada se koordinacija obavlja u više krugova, zahvaljujući značajno nižim troškovima formiranja informacione strukture.

⁶<http://www.isi.edu/nsnam/ns/>

⁷<https://www.iot-lab.info/>

4.2 Simulacije centralizovanih i distribuiranih algoritama za dodelu zadataka robotima

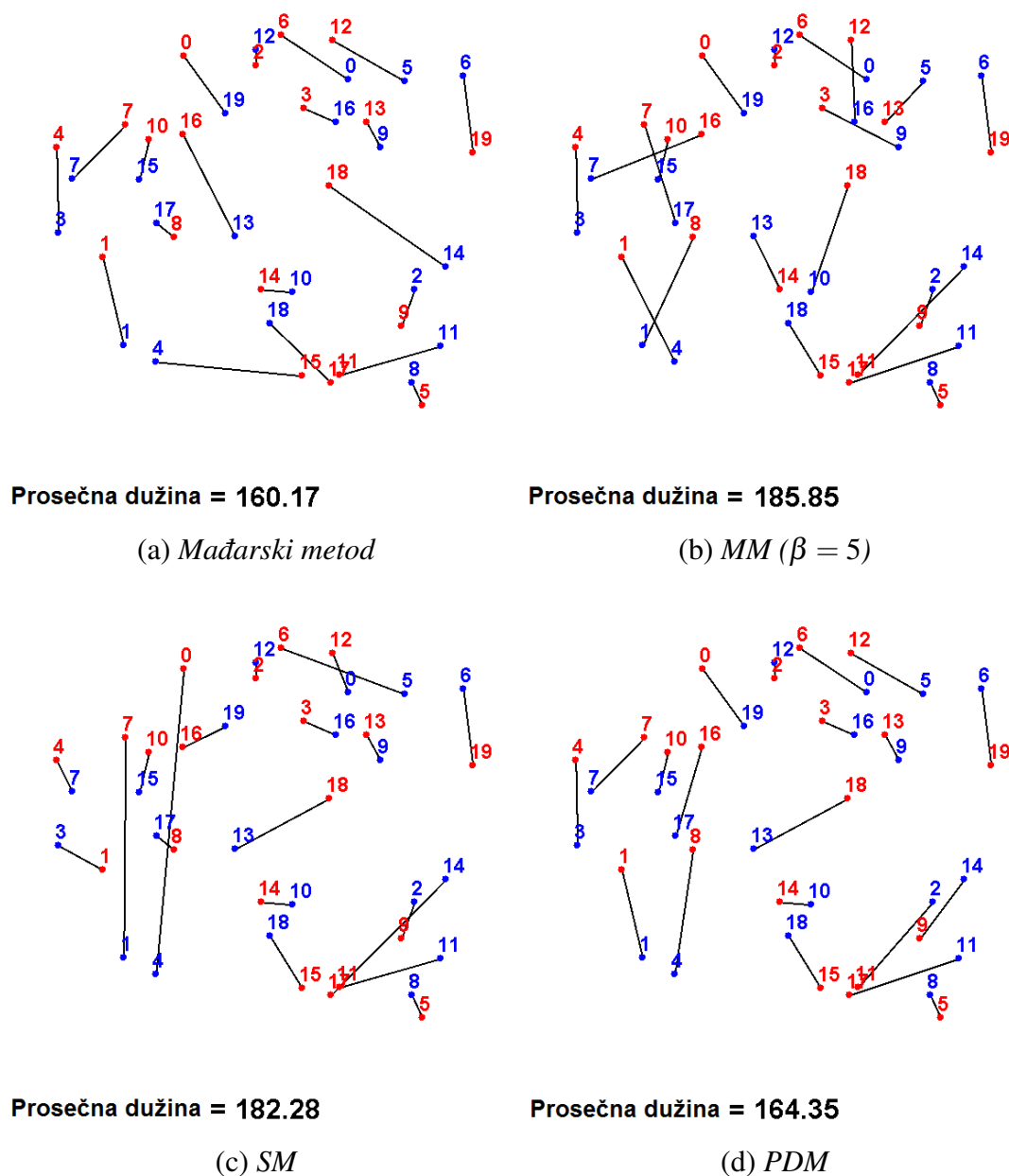
U ovom odeljku prikazani su i komentarisani rezultati simulacija kojima su poređene performanse novih rešenja opisanih u poglavlju 3 sa relevantnim postojećim rešenjima čiji pregled je dat u poglavlju 1, odeljci 1.6.1 i 1.6.3. Simulacioni algoritam predstavlja modifikaciju algoritma 14. Kompletni rezultati prikazani su u tabelama u dodatku B.

4.2.1 Centralizovano uparivanje robota s događajima

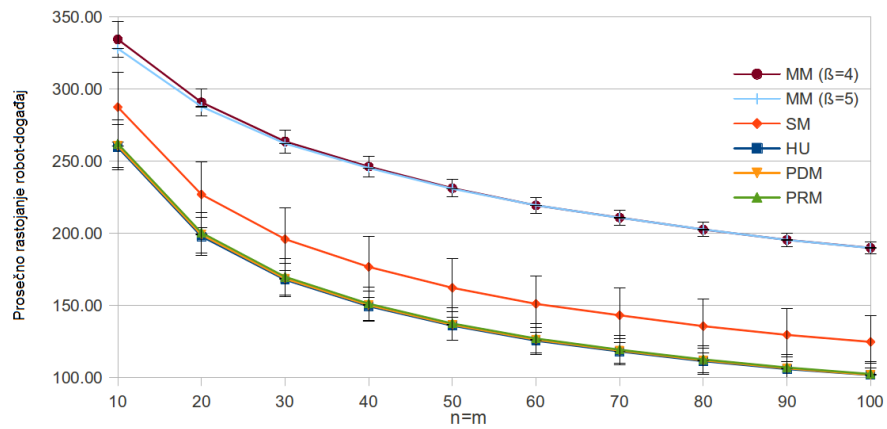
Kada su u pitanju centralizovani algoritmi za dodelu zadataka robotima, za početak poredili smo *SB* [AGB10][GS62], *MM* [WPT10] i Mađarski metod [Kuh55] sa našim algoritmima *PDM* and *PRM* [LS13][LBS15] opisanim u odeljku 3.1. Zajedničko za sve navedene algoritme je da rešavaju problem m -dodela(m, n). Na slici 4.9 prikazana su uparivanja koja se dobijaju primenom različitih algoritama na primeru u kom je $m = n = 20$.

Inicijalno, roboti se postavljaju unutar kvadratnog terena veličine 1000×1000 , a nakon toga se na početku svake runde nalaze na krajnjim lokacijama, tj. lokacijama događaja koji su im dodeljeni tokom prethodne runde. Događaji se pojavljuju na slučajnim pozicijama i u svakoj rundi je njihov broj jednak broju robota. Dve najvažnije mere performansi algoritama su prosečne međusobne udaljenosti robot-događaj tokom svih rundi i životno vreme sistema, koje je definisano kao broj rundi pre nego što se desi da neki od robota ne može da stigne do događaja koji mu je dodeljen, usled nedostatka energije. Smatra se da je energija koja se troši na kretanje linearno proporcionalna pređenoj udaljenosti, kao i da je inicijalna energija jednaka za sve robote i omogućava im da pređu ukupno po 50000 jedinica rastojanja.

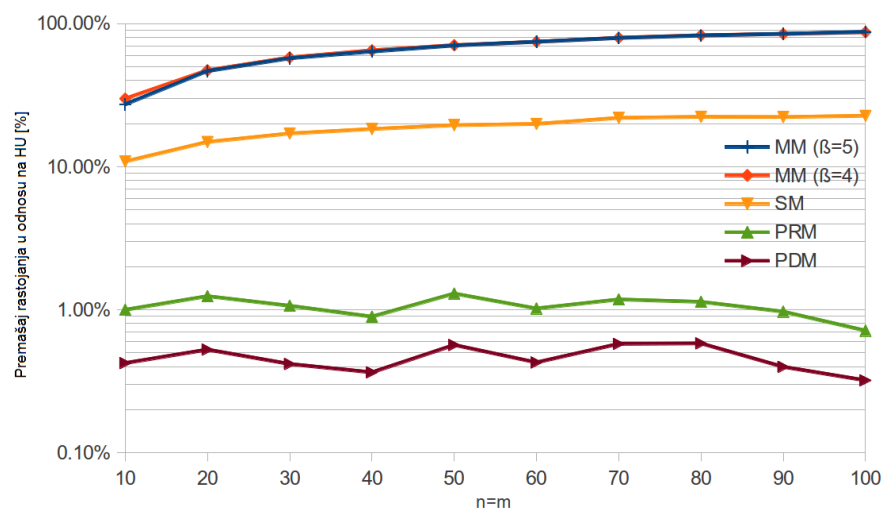
Slika 4.9 ilustruje parove robot-događaj koji se formiraju izvršenjem različitih algoritama za uparivanje. Naš *PDM* algoritam eliminiše udaljene parove i stoga daje bolju raspodelu opterećenja među robotima u poređenju sa osnovnim *SB* algoritmom. Rezultati simulacija prikazani na slikama 4.10a i 4.11 dati su u okviru 95%-tnih statističkih intervala poverenja (primetiti da su linije koje predstavljaju granice intervala poverenja jedva vidljive na slici 4.11). Podaci za prosečne dužine koje roboti prelaze tokom runde, dobijeni su izvršavanjem serija od po 100 simulacija sa parametrom $n = m$ koji se menja od 10 do 100 i prikazani su na slici 4.10a i u tabeli B.1. Uz to, poređeni su premašaji prosečnih dužina koje daju algoritmi za uparivanje, u odnosu na algoritam koji daje optimalno rešenje (Mađarski metod, označen skraćenicom *HU*), što je prikazano na grafiku na slici 4.10b i u tabeli B.2. Standardne devijacije rastojanja između robota i događaja koji su im dodeljeni prikazane su na slici 4.10c i u tabeli B.3.

Slika 4.9: Uparivanje robota sa događajima ($n = m = 20$)

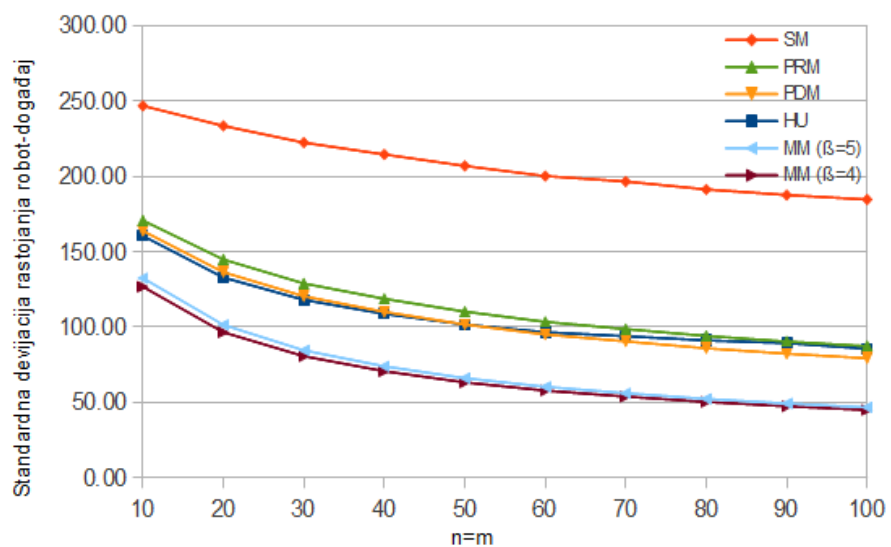
Rezultati simulacija pokazuju da su prosečna rastojanja robot-događaj za *PDM* oko 10 – 20% kraća u odnosu na *SM*. Dalje, ova rastojanja neznatno su (ne više od 1.3%) duža nego u slučaju kada se vrši optimalno uparivanje Mađarskim algoritmom. *PRM* produkuje slične, odnosno < 1% duže prosečne dužine putanja u odnosu na *PDM* algoritam. *MM* ($\beta = 4$) i *MM* ($\beta = 5$) produkuju duža prosečna rastojanja između robota i događaja koji su im dodeljeni, nego *SB* (15% duže za $n = 10$, pa i do 50% duže za $n = 100$, što daje indicaciju trenda povećanja razlike u performansama sa povećanjem broja parova).



(a)



(b)

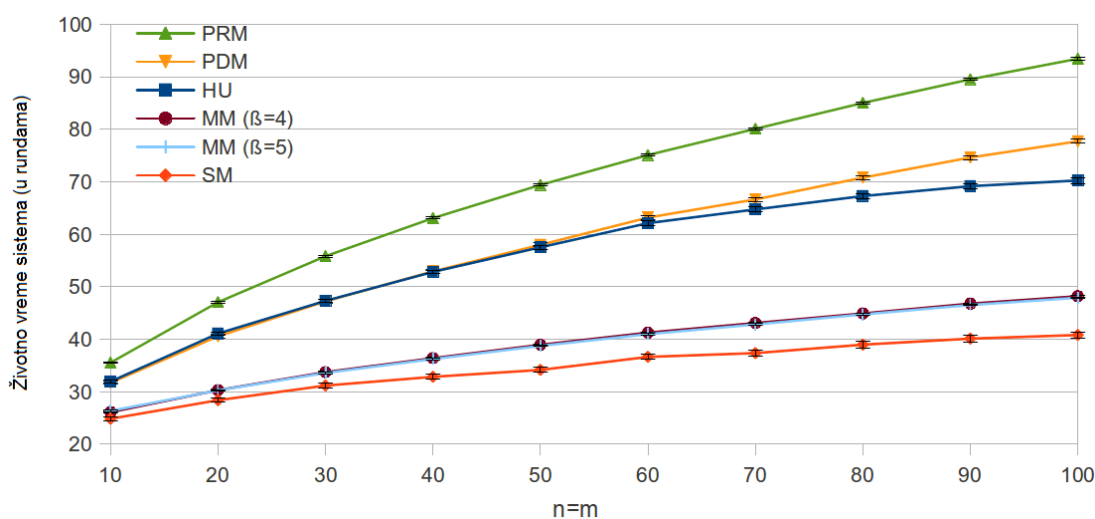


(c)

Slika 4.10: Rezultati simulacija za centralizovane algoritme uparivanja

Autori rada [WPT10] tvrde da je *MM* algoritam osmišljen sa idejom da se redukuje standardna devijacija među dužinama putanja. Naše simulacije potvrdile su ovu tvrdnju: *MM* ima 2 – 4 puta manju standardnu devijaciju nego *SB* i 20% – 45% manju u odnosu na *PDM* ili *PRM*. Ipak, ovakav pristup nije suviše efikasan kada se posmatra u svetlu ostvarenja primarnog cilja, budući da rezultuje značajno dužim putanjama robota.

Ključna prednost *PDM* u odnosu na *SB* ogleda se u značajnom unapređenju raspodele opterećenja među robotima, što se očituje povećanjem životnog vremena sistema u scenarijima sa više rundi (slika 4.11 i tabela B.4). U pogledu životnog vremena, najbolje performanse pokazuje *PRM*, sa 10% – 20% dužim životnim vremenom u odnosu na *PDM* i 10% – 35% dužim životnim vremenom nego u slučaju primene Mađarskog algoritma. Ovakav rezultat ne treba da deluje čudno, uprkos činjenici da Mađarski algoritam vrši optimalno uparivanje. Razlog za ovo leži u boljoj raspodeli opterećenja među robotima, pošto *PDM* postiže manju standardnu devijaciju dužina, a time i njihovu ravnomerniju distribuciju u odnosu na Mađarski algoritam, a *PRM* povrh toga uzima u obzir i preostalu energiju robota. Dalje, *PDM* pokazuje 30% – 90% duže životno vreme u odnosu na *SB*, dok *MM* pokazuje svega 5% – 35% duže životno vreme nego *SB*. *PRM* produžuje životno vreme sistema za značajnih 35% – 95% u odnosu na najbolji algoritam za uparivanje koji je do sada bio razmatran u radovima iz oblasti *BMSR* (*MM*[WPT10]).

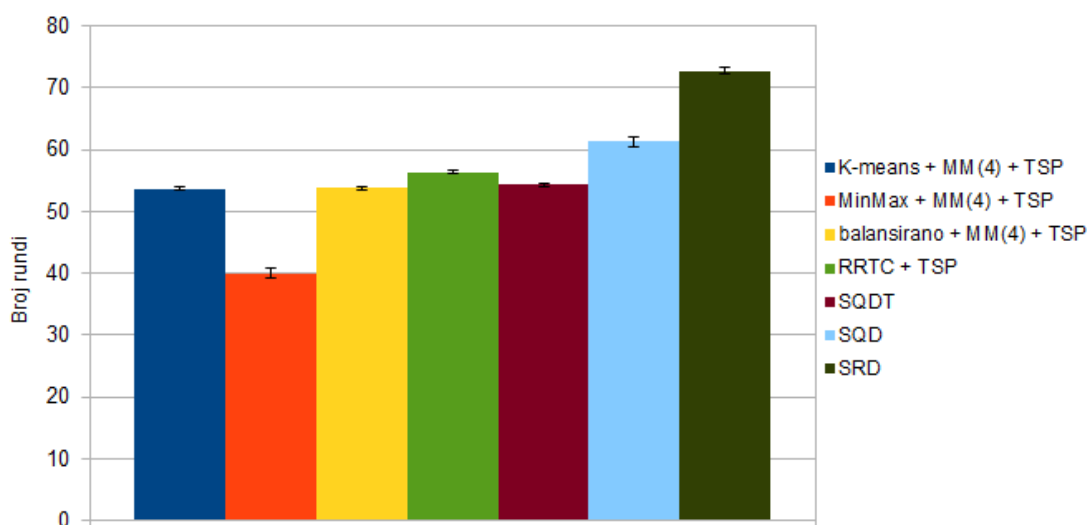


Slika 4.11: Životno vreme sistema prilikom primene različitih algoritama za uparivanje robota s događajima

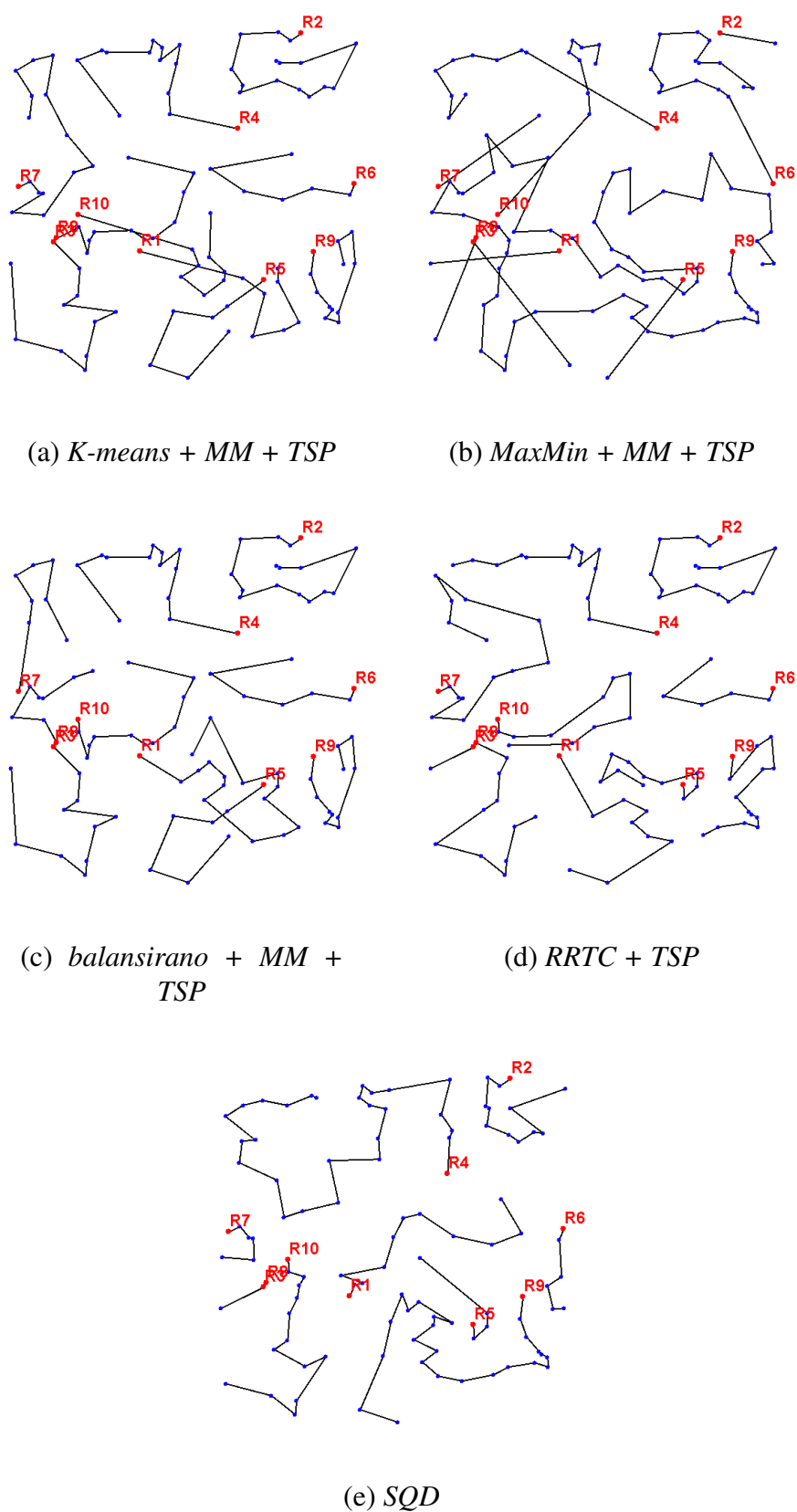
4.2.2 Centralizovana dodela sekvenci događaja izvršiocima

Nove algoritme *SQD*, *SRD* i *SQDT* poredili smo sa tri algoritma iz rada [WPT10], bazirana na *K-means*, *MaxMin* i *balansiranom* klasterovanju, kao i sa *RRTC* algoritmom iz rada [EGK⁺04], koji obavlja pokriće grafa stablima sa korenjem u skupu R . U svakoj rundi na slučajan način se među 1000 senzorskih čvorova bira $m = 100$ čvorova za koje se smatra da su detektovali događaje. Ove događaje opslužuje skup od $n = 10$ robota. Dimenzije terena, kao i inicijalna energija robota identični su kao u prethodnoj seriji simulacija. Slika 4.13 ilustruje putanje robota nakon primene različitih algoritama za dodelu događaja izvršiocima.

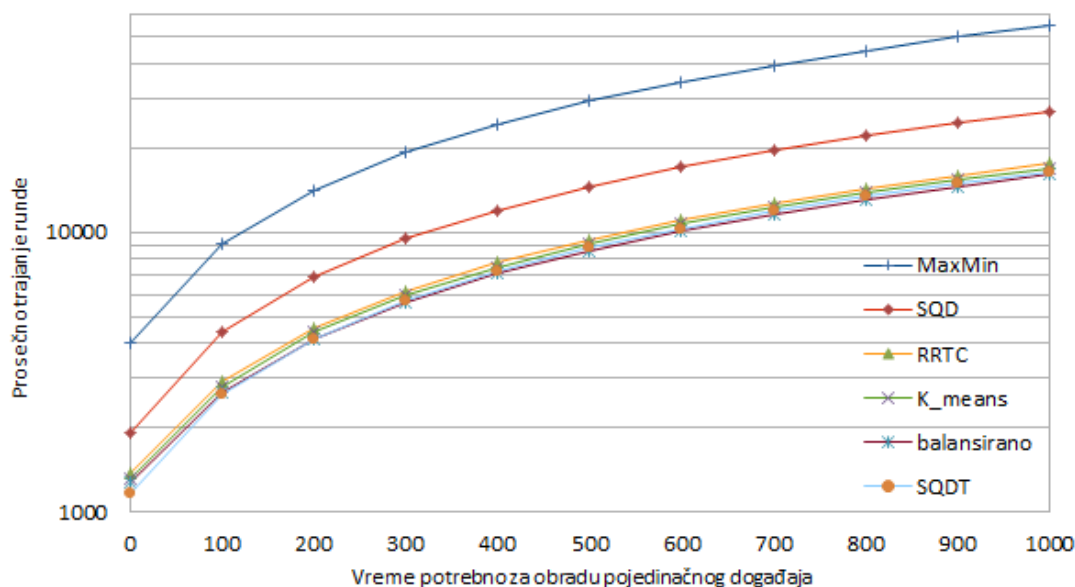
Rezultati za seriju od 100 simulacija prikazani su na slici 4.12 i u tabeli B.5, sa 95%–tnim intervalima poverenja. Kada je u pitanju životno vreme sistema, *K-means* klasterovanjem postiže se približno isti rezultat kao i sofisticiranijom šemom za balansirano klasterovanje. Ovakav rezultat ne iznenađuje, budući da se balansirano klasterovanje dobija iz *K-means* šeme nakon ne više od dve do tri iteracije sa podelom i spajanjem klastera. Naši jednostavni pohlepni algoritmi postižu duže životno vreme u odnosu na sve tri metode bazirane na klasterovanju i prikazane u radu [WPT10], kao i u odnosu na šemu baziranu na *RRTC* algoritmu [EGK⁺04]. *SRD* postiže oko 16% duže životno vreme nego *SQD* i oko 25% duže u odnosu na od ranije poznate algoritme (*K-means klasterovanje + MM + TSP / balansirano klasterovanje + MM + TSP / RRTC + TSP*, kao i u odnosu na naš *SQDT* algoritam (pri čemu je $(\tau_i = 0)$). Svi maločas pomenuti konkurentski algoritmi postižu sličan rezultat, osim *MinMax klasterovanje + MM + TSP*, gde je rezultat znatno lošiji usled nedovoljno dobro izbalansirane veličine klastera.



Slika 4.12: Životno vreme sistema prilikom primene različitih algoritama za sekvencijalnu dodelu događaja robotima

Slika 4.13: Primer dodele sekvenci događaja izvršiocima ($n = 10, m = 100$)

Dalje, simulirali smo centralizovane algoritme za dodelu sekvenci događaja izvršiocima, sa vremenskim ograničenjima. Trajanje runde definišemo kao maksimalno vreme potrebno da robot poseti i obradi sve događaje koji su mu dodeljeni. Uzimajući u obzir notaciju uvedenu u odeljku 3.2.3, usvojena je pretpostavka da svi roboti imaju istu brzinu ($\forall i \leq n)(v_i = 1)$. U pitanju je jedinična brzina koja podrazumeva da robot tokom kretanja prelazi jednu jedinicu rastojanja u jedinici vremena (npr. $1m/s$). Grafik na slici 4.14 dobijen na osnovu podataka iz tabele B.6 pokazuje zavisnost trajanja runde od vremena τ_i potrebnog da robot obradi pojedinačan događaj. Za ovu vrednost takođe se uvodi pretpostavka da je ista za sve robote. Vreme potrebno da robot R_i obavi zadatak koji mu je dodeljen tokom tekuće runde je $T_i = L_i \cdot v_i + k_i \cdot \tau_i$, gde L_i označava dužinu putanje, a k_i označava ukupan broj događaja dodeljenih robotu R_i . Prema tome, trajanje runde iznosi $T = \max_{i=1}^n T_i$.



Slika 4.14: Prosečno trajanje runde

Rezultati simulacija pokazuju da *SQDT* postiže približne prosečne dužine runde kao *K-means* klasterovanje + *MM* + *TSP*, *balansirano* klasterovanje + *MM* + *TSP* i *RRTC* + *TSP* algoritmi, unutar margine od 10%. Trajanje runde značajno je duže za *SQD* (oko 65%), a naročito za *MaxMin* klasterovanje + *MM* + *TSP* (oko 3.6 puta). U poslednja dva slučaja, do degradacije performansi dolazi usled neravnomerne raspodele opterećenja među robotima.

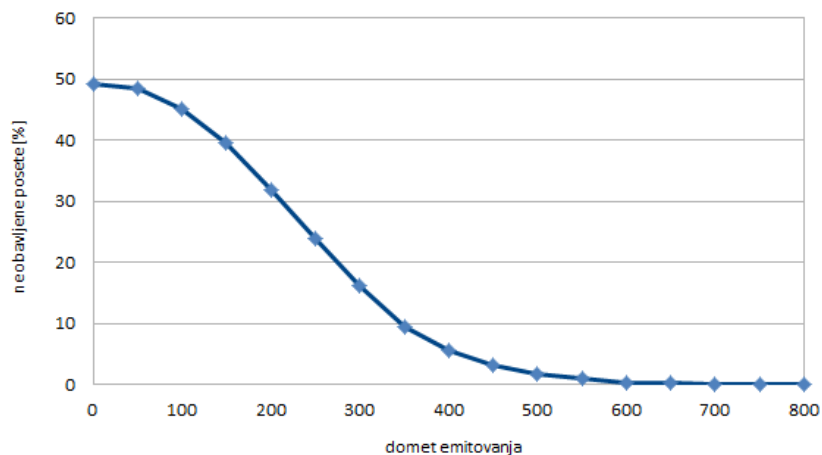
Potrebno je napomenuti da je *SQDT* algoritam osmišljen za korišćenje u heterogenim mrežama, sa različitim vrednostima v_i , odnosno τ_i . U tom smislu, pretpostavke o jednakim brzinama robota, kao i o jednakim vremenima potrebnim za obradu događaja pogoduju algoritmima koji nemaju ugrađen mehanizam kojim se uzimaju u obzir vremenski parametri sistema. Ponašanje algoritama u pogledu trajanja runde u heterogenim mrežama ostaje kao jedna od tema koja će biti obrađivana tokom budućih istraživanja.

4.2.3 Centralizovana i distribuirana dodela višestrukih izvršilaca događajima

Naredna serija simulacija sprovedena je za *GSQD* i distribuirani *GAD* algoritam, u slučaju kada je dozvoljeno da događaji zahtevaju da budu posećeni od strane više robota. Tokom svake runde, prisutno je $n = 10$ robota i na slučajan način bira se $m = 50$ statičkih čvorova za koje se smatra da su detektovali događaje. Pri tome podrazumeva se sledeća raspodela verovatnoća za vrednosti c_i :

$$Pr\{c_i = j\} = \begin{cases} \frac{1}{2^j} & , \text{ za } 1 \leq j < n \\ \frac{1}{2^{n-1}} & , \text{ za } j = n \\ 0 & , \text{ za } j > n \end{cases}$$

Prema tome, matematičko očekivanje $E(c_i) = 2 - \frac{n+2}{2^n}$, što je vrednost koja konvergira ka 2. Ukupan broj poseta koji se zahteva tokom tekuće runde je $VR = \sum_{i=1}^m c_i$. Centralizovani algoritam *GSQD* uvek je u stanju da pokrije sve događaje zahtevanim brojem poseta, budući da je raspodelom verovatnoća obezbeđeno da $(\forall 1 \leq i \leq m)(c_i \leq n)$. Ovo ne mora biti slučaj i za distribuirani algoritam *GAD*, usled ograničenog broja robota u podmreži, što je objašnjeno u odeljku 3.3.2. Topologija robotske mreže u velikoj meri je zavisna od dometa emitovanja. Slika 4.15 i tabela B.7 prikazuju procenat neobavljenih poseta koje se prenose u sledeći krug, u zavisnosti od dometa emitovanja. Kada je domet 0, roboti nisu u mogućnosti da komuniciraju međusobno, stoga svaki robot posećuje samo one događaje koji su mu direktno prijavljeni. Kako je $E(c_i) \approx 2$, a svaki od događaja biva posećen tačno jednom, oko 50% zahtevanog broja poseta se prenosi u sledeću rundu. Ovaj procenat opada ispod 1% za domete emitovanja $R > 500$.



Slika 4.15: Procenat broja poseta koje se prenose u narednu rundu

Takođe, poredili smo performanse *GSQD* i *GAD* algoritma u pogledu prosečne dužine putanja koje roboti prelaze. Simuliran je slučaj kada je domet emitovanja fiksiran ($R = 500$) i kako bi poređenje bilo pravedno, u obzir su uzimane samo one simulacije u kojima nije preostala nijedna poseta za sledeću rundu. U proseku, *GAD* produkuje 6.2% duže putanje u odnosu na *GSQD*, što znači da ne pokazuje značajno lošije performanse u ovom pogledu.

4.2.4 Distribuirana dodela sekvenci događaja izvršiocima

U nastavku poređene su performanse novih distribuiranih rešenja sa *WPT* algoritmom [WPT10]. Ispostavlja se da simulacioni parametri moraju biti izabrani veoma pažljivo, kako bi *WPT* uopšte mogao da funkcioniše. Za početak, posmatra se teren dimenzija 1000×1000 koji je pokriven rešetkom sa 10×10 ćelija, što znači da dužina stranica svake kvadratne ćelije iznosi $a = 100$. Kako bi bilo osigurano da se u svakoj ćeliji nalazi bar po jedan senzorski čvor koji će preuzeti ulogu koordinatora, koristi se pseudo-slučajna raspodela čvorova. Dalje, kako bi se obezbedilo da podmreža koju formiraju koordinatori bude povezana, odnosno da je svaki koordinatorski čvor u stanju da komunicira sa koordinatorskim čvorovima u svim ćelijama koje imaju zajedničku ivicu sa njegovom ćelijom, morali smo da podesimo domet komunikacije na vrednost $a \cdot \sqrt{5} \approx 224$, što je prilično velika vrednost imajući u vidu dimenzije ćelija i dimenzije terena. Pošto se nova rešenja ne oslanjaju na strukturu rešetke, mogu da funkcionišu i u znatno relaksiranijim uslovima u pogledu raspodele čvorova i dometa emitovanja, čak i za domete 3 – 4 puta kraće nego što je to neophodno u slučaju *WPT* algoritma.

Korišćeno je simulaciono okruženje sa istim parametrima kao u odeljku 4.2.2: u svakoj rundi je $n = 10$, $m = 100$, a inicijalna energija robota je 50000. Domet komunikacije postavljen je na vrednost $r = 250$ za komunikaciju sensor-senzor i sensor-robot, odnosno $R = 400$ za komunikaciju robot-robot⁸. Rezultati serije od 100 simulacija prikazani su u tabeli B.8, u okviru 95%–tnih intervala poverenja. Komunikacioni troškovi nisu uzimani u obzir kao kriterijum za određivanje životnog vremena sistema.

Rezultati pokazuju da novi distribuirani algoritmi *MAD* i *MRD* značajno produžavaju životno vreme sistema, smanjujući pri tome komunikacione troškove, gledano u odnosu na najbolji do sada poznat distribuirani algoritam *WPT*. *MAD* bez aukcije ($k = 0$), pronalazi najbliži robot za svaki događaj u 95% slučajeva i postiže oko 20% duže životno vreme u odnosu na *WPT*.

WPT koristi komunikacionu šemu zasnovanu na pravougaonom kvorumu. Tokom faze takmičenja, intenzivno se koristi komunikacija u statičkoj mreži, što rezultuje više od 4 puta većim komunikacionim troškovima. Uz to, komunikacija se pretežno obavlja između koordinatorskih čvorova, čiji broj iznosi 10% od ukupnog broja senzorskih

⁸Kao što je ranije napomenuto, ova vrsta komunikacije koristi se samo u novim distribuiranim algoritmima.

čvorova. To dovodi do neravnomerne raspodele opterećenja čija posledica je da koordinatori znatno ranije iscrpe raspoloživu energiju u odnosu na ostale čvorove.

Uvođenjem aukcija u robotsku mrežu, životno vreme sistema se produžava za oko 14% u odnosu na slučaj kada nema komunikacije među robotima (odnosno za $k = 0$) i za čak oko 37% u odnosu na *WPT*. Iznenadjuće, pokazuje se da povećanje broja hopova u aukcijskoj fazi ne donosi vidljivo poboljšanje u pogledu performansi sistema, budući da se najveća korist od aukcije dešava u prvom hopu (za $k = 1$). Prosečne dužine putanja neznatno se smanjuju povećanjem broja hopova i to za oko 2% kraće za $k = 4$ nego za $k = 1$, uz približno jednak broj rundi). Takođe, nema vidljivog poboljšanja životnog vremena kada se metrika promeni sa razdaljine na otpornost, budući da *MAD* i *MRD* pokazuju praktično iste performanse.

Isti simulacioni parametri korišćeni su prilikom evaluacije centralizovanih i distribuiranih algoritama za dodelu sekvenci događaja izvršiocima, čime je omogućeno poređenje njihovog učinka. Naš distribuirani algoritam *MAD* postiže bolji rezultat od svih centralizovanih rešenja predloženih u radu [WPT10]. Prosečno životno vreme sistema u slučaju primene *MAD* algoritma za ($k > 0$) za 4.5% je duže u odnosu na algoritam *balansirano klasterovanje + MM + TSP* koji postiže najduže životno vreme od svih algoritama iz rada [WPT10].

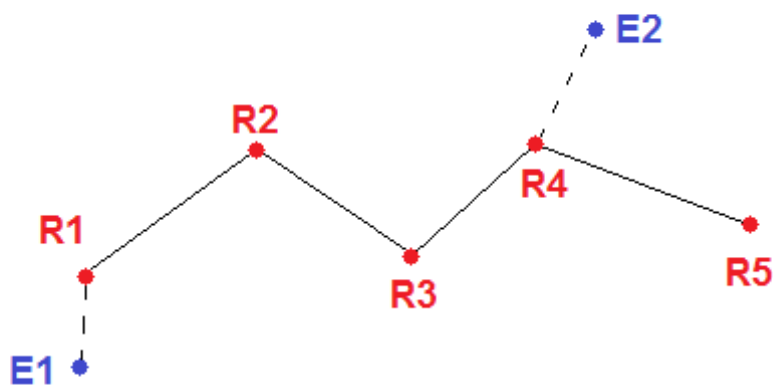
4.2.5 Zaključci u vezi sa algoritmima za dodelu zadataka izvršiocima

Problem koordinacije robota proučavan je polazeći od pretpostavke da su roboti vremenski sinhronizovani i da obavljaju zadatke u rundama jednakog trajanja. Na početku svake runde, roboti su statični pri čemu sakupljaju informacije o događajima i kreiraju liste događaja za obilazak. Nakon obavljanja zadataka, tj. obilaska događaja koji su im dodeljeni tokom date runde, roboti čekaju početak sledeće runde, ostajući na krajnjim pozicijama svojih putanja. U poglavlju 3 opisana su nova centralizovana i distribuirana rešenja problema dodele zadataka u bežičnim mrežama senzora i robota sa proizvoljnim odnosom između broja robota i broja događaja. Dalje, dozvoljeno je robotima da tokom iste runde budu zaduženi za obilazak više događaja i obratno, da događaji zahtevaju da budu posećeni od strane više robota. Prilikom pregleda literature nije uočeno da postoje radovi koji uzimaju u obzir ovu poslednju generalizaciju. U svakom slučaju, cilj je minimizovati dužine putanja koje roboti prelaze, kako bi bio maksimizovan broj rundi pre nego što roboti istroše energiju. U tu svrhu primenjena su dva različita pristupa i to uparivanje događaja sa izvršiocima i dodela sekvenci događaja izvršiocima. Prvo su razmatrane centralizovane, a zatim i distribuirane verzije rešenja.

Koncept je potvrđen simulacijama, koje pokazuju da su algoritmi koji su razvijeni tokom rada na ovoj disertaciji skalabilni u odnosu na postojeća rešenja, budući da se sa

povećanjem broja robota i događaja razlika u performansama povećava u korist novih rešenja. Kada je u pitanju uparivanje robota s događajima, novi *PDM* algoritam ima manju kompleksnost u odnosu na Mađarski metod koji daje optimalno rešenje, pri čemu je dobijeno rešenje u proseku svega za oko 1% lošije u pogledu prosečnih rastojanja robot-događaj. Ovo je međutim kompenzovano dužim životnim vremenom sistema, budući da novi algoritmi dovode do ujednačenije raspodele opterećenja među robotima.

Povoljna raspodela robota koji formiraju povezanu mrežu u pogledu lokacija gde je moguća, ili gde je unapred prognozirana pojava događaja, takođe može biti uzeta u razmatranje. Način na koji se roboti organizuju kako bi bila postignuta povoljna topologija mreže zasebno je pitanje koje je proučavano u radovima [DLNS09], [CSA14], [NS10]. Takođe, moguće je da se roboti kreću tragajući za događajima, ali u ovom slučaju održavanje strukture za otkrivanje izvršilaca postaje problematično. Jedna od mogućih strategija za postavljanje robota može biti da roboti kreću sa početnih pozicija i razilaze se po terenu, korišćenjem virtualnih sila. Pri tome je neophodno da ivice i topologija terena bude unapred poznata. Po završetku raspoređivanja, roboti staju i čekaju događaje, ne pomerajući se osim ako su pozvani da obiđu mesto događaja. Komunikacija između robota može se obavljati direktno, ili posredstvom senzorskih čvorova, pri čemu se koristi odgovarajući protokol za rutiranje kao što je recimo prikazano u radu [SN06]. Koordinacija može biti obavljena u stilu društvene mreže, [ZS14], umesto na osnovu susednosti. Pomenuti problemi i scenariji ostavljeni su za budući rad.



Slika 4.16: Problematičan slučaj distribuirane koordinacije

U ovoj disertaciji problem je formulisan tako da u slučaju kada neki od događaja zahteva prisustvo više robota, ne zahteva se da roboti budu fizički prisutni u isto vreme

na mestu događaja. Na primer, prilikom gašenja požara, svaki robot može da obavlja zadatak nezavisno od drugih robota, dok se u zavisnosti od intenziteta požara donosi odluka koliko robota je potrebno da bi se obavilo gašenje. Ukoliko se zahteva da više robota simultano bude prisutno na istom mestu, problem postaje značajno složeniji. U primeru prikazanom na slici 4.16, prikazana je robotska pod mreža koja se sastoji od pet robota, pri čemu je direktna komunikacija moguća između svaka dva uzastopna, a aukcije su lokalizovane na $k = 2$ hopa. Pretpostavka je da je događaj E_1 koji zahteva simultano prisustvo 3 robota prijavljen robotu R_1 , a događaj E_2 koji zahteva simultano prisustvo 4 robota prijavljen je robotu R_4 . R_1 je u stanju da odgovori na zahtev pozivanjem R_2 i R_3 , a R_4 pozivanjem R_2 , R_3 i R_5 . Postavlja se pitanje kako koordinirati robote u pogledu redosleda kojim će obilaziti događaje, budući da roboti R_1 i R_4 nemaju informacija jedan o drugom. Ako roboti samostalno donose odluku o redosledu obilaska, može se recimo desiti situacija da R_2 i R_3 krenu ka različitim događajima i nikad se međusobno ne dočekaju (engl. *deadlock*). To znači da je neophodan dogovor između njih, ali takav dogovor nije moguć i npr. u slučaju robota R_2 i R_5 koji treba da simultano posete događaj E_2 , budući da ova dva robota ne znaju jedan za drugog. Zaključak koji se nameće je da su u ovako postavljenom distribuiranom slučaju potrebni sofisticiraniji mehanizmi koordinacije, što takođe ostaje otvoreno pitanje i predstavlja jedan od budućih istraživačkih izazova.

Prilog A

Rezultati simulacija protokola za otkrivanje izvršilaca

A.1 Komunikacioni troškovi faze formiranja informacionog direktorijuma

Parametri simulacija:

Domest emitovanja $R = 75$

$$\text{Broj senzorskih čvorova} = \begin{cases} 1000, & \text{za topologiju sa uniformnom raspodelom čvorova} \\ 840, & \text{za topologiju sa prazninom u obliku kvadrata} \\ 803, & \text{za topologiju sa prazninom u obliku kruga} \\ 843, & \text{za topologiju sa prazninom u obliku polumeseca} \end{cases}$$

broj robota	Q_MFR	Q_LDR	iMeshC	iMeshD	vCell
1	1017.56	1026.14	1034.99	1049.27	999.98
2	1034.94	1051.96	1055.10	1078.50	1011.63
3	1052.32	1077.86	1072.27	1104.33	1010.07
4	1070.33	1104.09	1083.16	1118.80	1015.20
5	1087.12	1129.41	1094.63	1134.49	1014.26
6	1104.32	1154.85	1100.88	1145.91	1016.83
7	1122.3	1182.31	1110.14	1159.30	1015.33
8	1140.26	1210.60	1119.32	1169.19	1016.18
9	1157.16	1233.01	1125.56	1179.11	1015.85
10	1174.78	1258.97	1130.94	1187.06	1018.21

Tabela A.1: Troškovi formiranja direktorijuma - uniformna raspodela čvorova

broj robota	Q_MFR	Q_LDR	iMeshC	iMeshD	vCell
1	855.82	863.62	869.94	882.55	839.97
2	871.22	886.18	889.48	912.15	876.04
3	885.16	907.08	902.77	931.44	872.53
4	900.52	930.45	916.15	948.51	858.11
5	918.59	956.21	925.78	961.75	856.75
6	931.09	974.41	934.06	976.23	855.09
7	946.62	997.63	940.51	983.79	855.47
8	962.48	1022.85	947.48	994.77	854.57
9	979.43	1045.72	955.68	1005.67	853.89
10	989.81	1062.35	961.04	1015.46	853.3

Tabela A.2: Troškovi formiranja direktorijuma - praznina u obliku kvadrata

broj robota	Q_MFR	Q_LDR	iMeshC	iMeshD	vCell
1	820.15	828.80	834.72	849.57	802.96
2	835.82	851.19	855.77	877.34	824.94
3	853.13	877.64	869.21	899.02	823.87
4	870.24	902.67	879.89	913.22	821.00
5	886.00	927.00	892.16	930.91	815.30
6	902.36	954.51	898.74	942.16	813.54
7	919.64	976.47	908.54	957.60	813.59
8	935.02	1007.42	918.45	963.11	811.62
9	947.64	1017.49	919.18	966.12	813.94
10	957.56	1042.37	913.51	972.34	804.30

Tabela A.3: Troškovi formiranja direktorijuma - praznina u obliku kruga

broj robota	Q_MFR	Q_LDR	iMeshC	iMeshD	vCell
1	857.62	865.70	872.60	886.30	841.96
2	871.33	886.15	892.21	916.04	913.21
3	886.76	908.78	906.75	936.03	895.86
4	902.01	933.04	918.29	955.59	886.14
5	918.04	953.61	929.63	966.92	883.38
6	933.37	978.32	938.87	981.13	870.03
7	947.44	1000.27	944.96	992.12	866.90
8	961.54	1020.49	951.55	999.61	869.31
9	978.47	1045.40	959.03	1010.48	867.02
10	992.36	1066.77	963.83	1015.13	862.99

Tabela A.4: Troškovi formiranja direktorijuma - praznina u obliku polumeseca

A.2 Komunikacioni troškovi faze otkrivanja izvršilaca

broj robota	Q_MFR	Q_LDR	iMeshC	iMeshD	vCell
1	45.81	67.11	44.47	59.27	8.74
2	43.30	63.45	33.39	47.29	6.72
3	41.35	60.64	27.98	36.31	5.39
4	40.75	59.83	23.76	33.97	4.93
5	40.09	58.88	20.36	29.99	4.47
6	39.56	58.16	22.28	28.36	4.10
7	39.40	57.80	21.80	26.82	3.76
8	39.04	57.25	19.40	25.66	3.55
9	38.70	56.84	15.79	22.41	3.35
10	38.50	56.59	14.99	21.07	3.24

Tabela A.5: Troškovi pretrage i obaveštavanja izvršilaca - uniformna raspodela čvorova

broj robota	Q_MFR	Q_LDR	iMeshC	iMeshD	vCell
1	43.70	63.86	37.47	54.58	10.25
2	40.55	59.20	30.16	44.33	7.72
3	38.84	56.55	26.22	38.45	6.44
4	36.45	53.39	21.70	32.18	5.17
5	35.86	52.66	19.71	29.38	4.59
6	35.15	51.31	17.73	25.92	4.05
7	35.12	51.31	16.78	24.46	3.84
8	34.26	50.40	15.56	22.52	3.55
9	34.23	50.15	15.42	22.16	3.35
10	33.81	49.56	14.17	20.07	3.15

Tabela A.6: Troškovi pretrage i obaveštavanja izvršilaca - praznina u obliku kvadrata

broj robota	Q_MFR	Q_LDR	iMeshC	iMeshD	vCell
1	45.83	67.59	39.20	57.22	10.36
2	42.05	61.72	31.16	46.16	7.42
3	40.29	59.58	26.47	38.47	6.22
4	38.81	56.92	22.41	33.53	5.20
5	38.11	56.12	19.97	29.38	4.62
6	37.11	54.53	18.08	26.35	4.08
7	36.17	53.32	15.80	22.71	3.52
8	35.57	51.91	14.24	20.22	3.30
9	35.88	52.83	15.08	22.10	3.33
10	34.63	50.58	13.66	17.91	3.84

Tabela A.7: Troškovi pretrage i obaveštavanja izvršilaca - praznina u obliku kruga

broj robota	Q_MFR	Q_LDR	iMeshC	iMeshD	vCell
1	43.99	64.27	37.29	54.20	10.71
2	40.62	59.55	30.59	45.03	8.28
3	39.02	56.77	26.24	38.46	6.62
4	37.70	55.42	23.21	33.57	5.68
5	36.58	53.53	20.66	30.74	4.98
6	35.83	52.35	18.77	27.98	4.38
7	35.36	52.00	17.58	25.57	4.04
8	35.30	51.60	16.52	24.27	3.84
9	34.07	50.26	15.41	22.55	3.55
10	34.72	50.73	14.81	21.54	3.35

Tabela A.8: Troškovi pretrage i obaveštavanja izvršilaca - praznina u obliku polumeseca

A.3 Komunikacioni troškovi faze održavanja informacionog direktorijuma

broj robota	Q_MFR	Q_LDR	iMeshC	iMeshD	vCell
1	53.58	70.60	86.94	115.82	2000.96
2	53.54	70.45	75.91	101.59	1082.45
3	53.47	70.50	69.29	91.68	757.01
4	53.49	70.55	64.49	85.22	604.26
5	53.40	70.42	61.85	80.30	506.87
6	53.33	70.31	58.38	74.93	418.07
7	53.37	70.50	56.26	71.97	364.53
8	53.50	70.81	54.15	68.93	324.76
9	53.34	70.33	52.67	66.44	289.73
10	53.32	70.19	51.53	63.75	265.63

Tabela A.9: Troškovi održavanja direktorijuma - uniformna raspodela čvorova

broj robota	Q_MFR	Q_LDR	iMeshC	iMeshD	vCell
1	43.70	63.86	37.47	54.58	10.25
2	40.55	59.20	30.16	44.33	7.72
3	38.84	56.55	26.22	38.45	6.44
4	36.45	53.39	21.70	32.18	5.17
5	35.86	52.66	19.71	29.38	4.59
6	35.15	51.31	17.73	25.92	4.05
7	35.12	51.31	16.78	24.46	3.84
8	34.26	50.40	15.56	22.52	3.55
9	34.23	50.15	15.42	22.16	3.35
10	33.81	49.56	14.17	20.07	3.15

Tabela A.10: Troškovi održavanja direktorijuma - praznina u obliku kvadrata

broj robota	Q_MFR	Q_LDR	iMeshC	iMeshD	vCell
1	51.19	67.81	81.22	109.71	1606.92
2	50.84	66.82	72.76	95.62	839.52
3	51.24	67.67	67.17	88.55	614.90
4	51.09	67.19	61.94	80.57	479.35
5	50.67	67.00	58.73	76.19	405.73
6	50.65	67.78	56.01	71.00	347.63
7	50.42	66.42	52.27	67.06	278.15
8	49.89	66.13	52.86	65.17	279.22
9	50.04	65.62	50.35	63.17	248.31
10	49.64	65.60	47.44	58.91	198.80

Tabela A.11: Troškovi održavanja direktorijuma - praznina u obliku kruga

broj robota	Q_MFR	Q_LDR	iMeshC	iMeshD	vCell
1	51.19	67.81	81.22	109.71	1606.92
2	50.84	66.82	72.76	95.62	839.52
3	51.24	67.67	67.17	88.55	614.90
4	51.09	67.19	61.94	80.57	479.35
5	50.67	67.00	58.73	76.19	405.73
6	50.65	67.78	56.01	71.00	347.63
7	50.42	66.42	52.27	67.06	278.15
8	49.89	66.13	52.86	65.17	279.22
9	50.04	65.62	50.35	63.17	248.31
10	49.64	65.60	47.44	58.91	198.80

Tabela A.12: Troškovi održavanja direktorijuma - praznina u obliku polumeseca

A.4 Procenat uspeha u pronalaženju najbližeg izvršioca

broj robota	Q_MFR	Q_LDR	iMeshC	iMeshD	vCell
1	99.10	98.47	98.70	99.07	100.00
2	98.69	98.81	97.43	97.21	99.99
3	99.31	99.15	97.00	97.61	99.98
4	99.47	99.41	96.62	96.36	99.98
5	98.68	98.68	96.77	95.70	99.91
6	99.19	98.97	96.20	96.02	99.88
7	99.31	99.38	95.93	95.50	99.91
8	99.37	99.37	96.30	95.43	99.94
9	99.68	99.71	96.82	95.77	99.93
10	99.55	99.51	96.38	96.20	99.88

Tabela A.13: Procenat uspeha u zavisnosti od broja robota - uniformna raspodela čvorova

broj robota	Q_MFR	Q_LDR	iMeshC	iMeshD	vCell
1	74.97	75.83	93.27	91.21	100.00
2	90.68	90.12	98.97	98.74	100.00
3	94.29	94.27	99.58	99.43	99.99
4	95.99	96.28	99.71	99.61	100.00
5	98.06	97.64	100.00	100.00	100.00
6	99.13	98.88	100.00	99.99	100.00
7	99.27	99.43	99.99	99.99	99.99
8	99.46	99.49	99.99	99.94	99.99
9	99.59	99.42	100.00	100.00	100.00
10	99.71	99.66	100.00	100.00	100.00

Tabela A.14: Procenat uspeha u zavisnosti od broja robota - praznina u obliku kvadrata

broj robota	Q_MFR	Q_LDR	iMeshC	iMeshD	vCell
1	86.17	85.99	97.16	96.24	100.00
2	90.04	89.74	96.36	95.13	99.99
3	95.05	94.63	96.95	95.36	99.51
4	97.33	97.15	96.47	95.62	99.27
5	97.21	97.25	96.16	95.41	99.14
6	98.66	99.12	96.63	97.01	99.84
7	98.94	98.91	96.93	96.54	99.56
8	99.92	99.92	97.19	97.46	99.92
9	97.62	97.67	95.51	94.89	98.56
10	97.75	97.88	96.52	96.27	97.88

Tabela A.15: Procenat uspeha u zavisnosti od broja robota - praznina u obliku kruga

broj robota	Q_MFR	Q_LDR	iMeshC	iMeshD	vCell
1	69.68	69.80	87.51	84.46	100.00
2	85.16	84.82	95.90	96.05	100.00
3	92.70	91.94	98.21	97.95	99.99
4	94.04	94.47	98.42	98.81	100.00
5	96.56	96.40	99.46	99.63	100.00
6	97.20	96.97	99.93	99.83	100.00
7	98.35	98.16	99.77	99.96	100.00
8	98.48	98.52	99.85	99.85	99.99
9	98.93	98.96	99.99	99.99	100.00
10	99.16	99.16	99.99	99.97	100.00

Tabela A.16: Procenat uspeha u zavisnosti od broja robota - praznina u obliku polumeseca

U narednoj seriji simulacija se postavka menja fiksiranjem broja robota ($n = 5$), a domet emitovanja se menja od 10 do 100 u koracima po 10.

domet emitovanja	Q_MFR	Q_LDR	iMeshC	iMeshD	vCell
10	0.17	0.17	0.17	0.17	0.17
20	1.13	1.12	1.16	1.16	1.16
30	5.01	5.03	5.80	5.90	6.99
40	24.05	24.21	31.00	31.23	63.93
50	66.93	67.04	75.22	74.21	96.70
60	91.05	90.78	91.92	91.07	99.41
70	98.17	97.98	96.74	95.52	100.00
80	100.00	100.00	97.73	96.44	100.00
90	100.00	100.00	98.11	97.18	100.00
100	100.00	100.00	97.94	97.57	100.00

Tabela A.17: Procenat uspeha u zavisnosti od dometa - Uniformna raspodela čvorova

domet emitovanja	Q_MFR	Q_LDR	iMeshC	iMeshD	vCell
10	0.20	0.20	0.20	0.20	0.20
20	1.25	1.25	1.28	1.28	1.29
30	5.87	5.86	6.51	6.65	7.82
40	23.24	24.08	30.44	30.15	54.22
50	60.20	59.29	69.67	69.37	93.27
60	85.70	85.56	90.16	89.16	99.14
70	93.82	93.75	95.47	94.80	99.54
80	96.85	96.33	97.07	96.06	99.92
90	97.36	97.39	97.66	96.94	99.82
100	98.54	98.45	97.65	97.30	100.00

Tabela A.18: Procenat uspeha u zavisnosti od dometa - praznina u obliku kvadrata

domet emitovanja	Q_MFR	Q_LDR	iMeshC	iMeshD	vCell
10	0.16	0.16	0.16	0.16	0.16
20	1.33	1.33	1.36	1.37	1.37
30	5.81	5.82	6.53	6.62	7.65
40	25.59	25.60	32.17	32.31	58.09
50	60.84	61.14	70.91	71.28	94.12
60	87.00	87.13	91.33	90.53	99.42
70	95.85	95.69	95.74	94.98	99.61
80	98.31	98.30	97.26	96.89	99.85
90	100.00	99.94	98.40	97.43	100.00
100	100.00	100.00	98.97	98.26	100.00

Tabela A.19: Procenat uspeha u zavisnosti od dometa - praznina u obliku kruga

domet emitovanja	Q_MFR	Q_LDR	iMeshC	iMeshD	vCell
10	0.22	0.22	0.22	0.22	0.22
20	1.28	1.27	1.32	1.32	1.33
30	5.42	5.42	6.12	6.13	7.05
40	21.97	21.85	27.47	26.93	45.53
50	52.17	51.92	62.71	61.35	85.31
60	76.92	77.09	82.05	80.67	92.34
70	86.15	86.13	87.92	86.50	93.39
80	89.95	89.80	90.63	90.00	95.59
90	92.42	92.42	92.63	92.24	96.87
100	92.98	92.66	92.93	91.84	95.92

Tabela A.20: Procenat uspeha u zavisnosti od dometa - praznina u obliku polumeseca

U tabeli A.21 prikazan je procenat uspeha u pronalaženju bilo kog (ne nužno najbližeg) robota za različite komunikacione protokole, pri čemu je $n = 5$, a domet emitovanja se menja od 10 do 100 u koracima po 10.

domet emitovanja	Q_MFR	Q_LDR	iMeshC	iMeshD	vCell
10	0.17	0.17	0.17	0.17	0.17
20	1.13	1.13	1.17	1.17	1.17
30	5.12	5.18	5.99	6.11	7.45
40	29.10	29.74	38.08	38.44	84.62
50	83.95	83.56	91.72	91.37	99.34
60	97.36	97.68	99.66	99.54	100.00
70	99.86	99.84	100.00	100.00	100.00
80	100.00	100.00	100.00	100.00	100.00
90	100.00	100.00	100.00	100.00	100.00
100	100.00	100.00	100.00	100.00	100.00

Tabela A.21: Procenat uspeha u pronalaženju bilo kog izvršioca u zavisnosti od dometa - uniformna raspodela čvorova

Prilog B

Rezultati simulacija algoritama za dodelu zadataka izvršiocima

B.1 Centralizovano uparivanje robota s događajima

$n = m$	HU	SM	PDM	PRM	MM ($\beta = 4$)	MM ($\beta = 5$)
10	259.39	287.43	260.48	262.06	334.41	328.05
20	197.70	226.78	198.85	200.26	290.75	287.80
30	167.60	195.75	168.39	169.63	263.60	262.16
40	149.24	176.52	149.98	150.99	246.12	245.16
50	135.73	162.01	136.35	137.26	231.15	230.75
60	125.26	150.86	125.98	126.82	219.21	219.27
70	117.80	142.96	118.44	119.06	210.71	210.61
80	111.07	135.40	111.67	112.31	202.44	202.52
90	105.60	129.34	106.11	106.71	195.28	195.23
100	101.46	124.45	101.56	102.18	189.74	189.65

Tabela B.1: Prosečno rastojanje robot-događaj

$n = m$	MM ($\beta = 5$)	MM ($\beta = 4$)	SM	PRM	PDM
10	26.47%	28.92%	10.81%	1.03%	0.42%
20	45.58%	47.07%	14.71%	1.29%	0.58%
30	56.42%	57.28%	16.80%	1.21%	0.47%
40	64.27%	64.92%	18.28%	1.17%	0.50%
50	70.01%	70.30%	19.36%	1.13%	0.46%
60	75.05%	75.00%	20.43%	1.24%	0.57%
70	78.78%	78.87%	21.36%	1.07%	0.54%
80	82.33%	82.25%	21.90%	1.12%	0.54%
90	84.87%	84.92%	22.48%	1.05%	0.48%
100	86.93%	87.02%	22.66%	0.72%	0.35%

Tabela B.2: Premašaj rastojanja u odnosu na HU algoritam

$n = m$	HU	SM	PDM	PRM	MM ($\beta = 4$)	MM ($\beta = 5$)
10	160.39	246.79	163.69	170.62	126.65	132.34
20	132.78	233.46	136.32	144.72	96.41	101.32
30	117.90	222.35	120.35	128.83	80.53	84.41
40	108.62	214.50	109.98	118.63	70.54	73.80
50	101.54	206.88	101.59	110.12	63.11	65.99
60	96.45	200.13	94.94	103.31	57.72	60.19
70	93.78	196.52	90.37	98.51	53.69	55.92
80	91.02	191.26	85.76	93.98	50.20	52.18
90	89.18	187.54	82.20	90.26	47.28	49.14
100	85.46	184.58	79.17	87.21	44.83	46.56

Tabela B.3: Standardna devijacija rastojanja robot-događaj

$n = m$	HU	SM	PDM	PRM	MM ($\beta = 4$)	MM ($\beta = 5$)
10	31.92	24.83	31.69	35.54	26.08	26.34
20	41.08	28.37	40.56	47.04	30.27	30.25
30	47.29	31.16	47.20	55.80	33.71	33.56
40	52.81	32.83	52.91	63.06	36.39	36.22
50	57.50	34.14	57.96	69.38	38.93	38.72
60	62.10	36.63	63.15	75.06	41.23	40.95
70	64.71	37.35	66.62	80.07	43.07	42.80
80	67.25	38.96	70.77	85.01	44.89	44.71
90	69.14	40.11	74.59	89.50	46.77	46.51
100	70.23	40.79	77.69	93.42	48.18	47.89

Tabela B.4: Životno vreme sistema (u rundama)

B.2 Centralizovana dodela sekvenci događaja izvršiocima

Algoritam	Životno vreme (u rundama)
K-means klasterovanje + MM($\beta = 4$) + TSP	53.71
MaxMin klasterovanje + MM($\beta = 4$) + TSP	39.98
Balansirano klasterovanje + MM($\beta = 4$) + TSP	53.78
RRTC + TSP	56.35
SQDT ($\tau_i = 0$)	54.29
SQD	61.27
SRD	72.76

Tabela B.5: Životno vreme sistema prilikom centralizovane dodele sekvenci događaja izvršiocima

τ_i	MaxMin	SQD	RRTC	K_means	balansirano	SQDT
0	4001.9	1930.9	1377.1	1323.6	1278.4	1159.8
100	9073.6	4424.9	2922.7	2819.6	2662.4	2626.2
200	14150.6	6942.5	4535.2	4371.8	4133.9	4153.4
300	19227.6	9462.2	6163.2	5935.4	5618.6	5688.5
400	24304.6	11983.2	7793.9	7505.1	7106.6	7226.0
500	29381.6	14504.2	9425.1	9077.1	8596.2	8765.0
600	34458.6	17025.2	11057.4	10649.1	10086.2	10304.0
700	39535.6	19546.2	12690.4	12221.9	11576.2	11843.0
800	44612.6	22067.2	14323.4	13794.9	13066.2	13382.0
900	49689.6	24588.2	15956.4	15367.9	14556.2	14921.0
1000	54766.6	27109.2	17589.4	16940.9	16046.2	16460.0

Tabela B.6: Prosečno trajanje runde

B.3 Centralizovana i distribuirana dodela višestrukih izvršilaca događajima

domet emitovanja	procenat neobavljenih poseta
0	49.29%
50	48.56%
100	45.08%
150	39.61%
200	31.81%
250	23.83%
300	16.07%
350	9.42%
400	5.48%
450	3.13%
500	1.62%
550	0.92%
600	0.31%
650	0.13%
700	0.07%
750	0.04%
800	0.04%

Tabela B.7: Procenat broja poseta koje se prenose u narednu rundu

B.4 Distribuirana dodela sekvenci događaja izvršiocima

algoritam	broj rundi	broj poruka (senzorska mreža)	br. poruka (robotska mreža)	dužina robotskih putanja po rundi
WPT	41.43 ± 0.36	7679.26 ± 20.71	0.00 ± 0.00	10380.67 ± 31.55
MAD(k = 0)	50.19 ± 0.66	1798.22 ± 3.73	0.00 ± 0.00	8272.54 ± 14.56
MAD(k = 1)	56.07 ± 0.44	1789.18 ± 2.50	306.22 ± 1.72	7877.33 ± 12.38
MAD(k = 2)	56.71 ± 0.49	1788.95 ± 2.40	869.18 ± 7.15	7793.96 ± 11.86
MAD(k = 3)	56.52 ± 0.44	1789.17 ± 2.54	1365.49 ± 15.14	7765.82 ± 13.55
MAD(k = 4)	57.02 ± 0.45	1789.43 ± 2.44	1671.30 ± 19.47	7750.49 ± 12.94
MRD(k = 1)	56.07 ± 0.44	1789.18 ± 2.50	306.22 ± 1.72	7877.33 ± 12.38
MRD(k = 2)	56.71 ± 0.48	1789.03 ± 2.39	869.58 ± 7.22	7792.53 ± 11.97
MRD(k = 3)	56.57 ± 0.45	1789.19 ± 2.52	1363.00 ± 14.85	7765.21 ± 13.18
MRD(k = 4)	57.01 ± 0.48	1789.50 ± 2.54	1672.00 ± 20.27	7749.60 ± 13.29

Tabela B.8: Distribuirani algoritmi za dodelu zadataka izvršiocima

Literatura

- [AAI09] Nor Azlina Ab Aziz, Kamarulzaman Ab Aziz, and Wan Zakiah Wan Ismail. Coverage strategies for wireless sensor networks. *World academy of science, Engineering and technology*, 50:145–150, 2009.
- [AE⁺09] L Alazzawi, A Elkateeb, et al. Performance evaluation of the wsn routing protocols scalability. *Journal of Computer Systems, Networks, and Communications*, 2008, 2009.
- [AGB10] Kemal Akkaya, Ismail Guneydas, and Ali Bicak. Autonomous actor positioning in wireless sensor and actor networks using stable-matching. *International Journal of Parallel, Emergent and Distributed Systems*, 25:439–464, 2010.
- [AIRX08] Waleed Alsalih, Kamrul Islam, Yurai Núñez Rodríguez, and Henry Xiao. Distributed voronoi diagram computation in wireless sensor networks. In *SPAA*, page 364, 2008.
- [AK04] Ian F Akyildiz and Ismail H Kasimoglu. Wireless sensor and actor networks: research challenges. *Ad hoc networks*, 2(4):351–367, 2004.
- [APL99] Konstantinos N Amouris, Symeon Papavassiliou, and Miao Li. A position-based multi-zone routing protocol for wide area mobile ad-hoc networks. In *Vehicular Technology Conference, 1999 IEEE 49th*, volume 2, pages 1365–1369. IEEE, 1999.
- [Aur91] Franz Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405, 1991.
- [AV10] Ian F Akyildiz and Mehmet Can Vuran. *Wireless sensor networks*, volume 4. John Wiley & Sons, 2010.
- [AY05] Kemal Akkaya and Mohamed Younis. A survey on routing protocols for wireless sensor networks. *Ad hoc networks*, 3(3):325–349, 2005.
- [BA98] Tucker Balch and Ronald C Arkin. Behavior-based formation control for multirobot teams. *Robotics and Automation, IEEE Transactions on*, 14(6):926–939, 1998.

- [BCDV09] Chiara Buratti, Andrea Conti, Davide Dardari, and Roberto Verdone. An overview on wireless sensor networks technology and evolution. *Sensors*, 9(9):6869–6896, 2009.
- [Ber88] D.P. Bertsekas. The auction algorithm: A distributed relaxation method for the assignment problem. *Annals of Operations Research*, 14(1):105–123, 1988.
- [BG91] DP Barnes and JO Gray. Behaviour synthesis for co-operant mobile robot control. In *Control 1991. Control'91., International Conference on*, pages 1135–1140. IET, 1991.
- [BGT03] Curt Bererton, Geoffrey J Gordon, and Sebastian Thrun. Auction mechanism design for multi-robot coordination. In *Advances in Neural Information Processing Systems*, page None, 2003.
- [BHK⁺03] Marc Berhault, He Huang, Pinar Keskinocak, Sven Koenig, Wedad Elmaghraby, Paul Griffin, and Anton Kleywegt. Robot exploration with combinatorial auctions. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 2, pages 1957–1962. IEEE, 2003.
- [BM76] John Adrian Bondy and Uppaluri Siva Ramachandra Murty. *Graph theory with applications*, volume 290. Macmillan London, 1976.
- [BMJ⁺98] Josh Broch, David A Maltz, David B Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pages 85–97. ACM, 1998.
- [BMSS05] Wolfram Burgard, Mark Moors, Cyrill Stachniss, and Frank E Schneider. Coordinated multi-robot exploration. *Robotics, IEEE Transactions on*, 21(3):376–386, 2005.
- [BMSU01] Prosenjit Bose, Pat Morin, Ivan Stojmenović, and Jorge Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless networks*, 7(6):609–616, 2001.
- [BPC⁺07] Paolo Baronti, Prashant Pillai, Vince WC Chook, Stefano Chessa, Alberto Gotta, and Y Fun Hu. Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and zigbee standards. *Computer communications*, 30(7):1655–1695, 2007.
- [CASH08] Qing Cao, Tarek Abdelzaher, John Stankovic, and Tian He. The liteos operating system: Towards unix-like abstractions for wireless

- sensor networks. In *Information Processing in Sensor Networks, 2008. IPSN'08. International Conference on*, pages 233–244. IEEE, 2008.
- [CCJ90] Brent N Clark, Charles J Colbourn, and David S Johnson. Unit disk graphs. *Discrete mathematics*, 86(1):165–177, 1990.
- [CD13] Zuhail Can and Murat Demirbas. A survey on in-network querying and tracking services for wireless sensor networks. *Ad Hoc Networks*, 11(1):596–610, 2013.
- [CFK97] Y Uny Cao, Alex S Fukunaga, and Andrew Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous robots*, 4(1):7–27, 1997.
- [CH02] Min Cao and Christoforos Hadjicostis. Distributed algorithms for voronoi diagrams and application in ad-hoc networks. *Preprint, Oct*, pages 1–12, 2002.
- [Che89] L Paul Chew. Constrained delaunay triangulations. *Algorithmica*, 4(1-4):97–108, 1989.
- [Chr76] Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, DTIC Document, 1976.
- [CIQC02] Krishnendu Chakrabarty, S Sitharama Iyengar, Hairong Qi, and Eungchun Cho. Grid coverage for surveillance and target location in distributed sensor networks. *Computers, IEEE Transactions on*, 51(12):1448–1453, 2002.
- [CLR⁺01] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, Clifford Stein, et al. *Introduction to algorithms*, volume 2. MIT press Cambridge, 2001.
- [CMM98] Joseph Czyzyk, Michael P Mesnier, and Jorge J Moré. The neos server. *Computing in Science and Engineering*, 5(3):68–75, 1998.
- [CPG⁺14] Shaobin Cai, Hongqi Pan, Zhenguo Gao, Nianmin Yao, and Zhiqiang Sun. Research of localization algorithm based on weighted voronoi diagrams for wireless sensor network. *EURASIP Journal on Wireless Communications and Networking*, 2014(1):50, 2014.
- [CSA14] Salimur Choudhury, Kai Salomaa, and Selim G. Akl. Cellular automaton-based algorithms for the dispersion of mobile wireless sensor networks. *International Journal of Parallel, Emergent and Distributed Systems*, 29(2):147–177, 2014.

- [CSKC01] Luiz Chaimowicz, Thomas Sugar, Vijay Kumar, and Mario Fernando Montenegro Campos. An architecture for tightly coupled multi-robot cooperation. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 3, pages 2992–2997. IEEE, 2001.
- [CSZ⁺09] Yebin Chen, Jian Shu, Sheng Zhang, Linlan Liu, and Limin Sun. Data fusion in wireless sensor networks. In *Electronic Commerce and Security, 2009. ISECS'09. Second International Symposium on*, volume 2, pages 504–509. IEEE, 2009.
- [CV10] B. Coltin and M. Veloso. Mobile robot task allocation in hybrid wireless sensor networks. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2932–2937, 2010.
- [DGF⁺91] Jean-Louis Deneubourg, Simon Goss, Nigel Franks, Ana Sendova-Franks, Claire Detrain, and Laeticia Chrétien. The dynamics of collective sorting robot-like ants and ant-like robots. In *Proceedings of the first international conference on simulation of adaptive behavior on From animals to animats*, pages 356–363, 1991.
- [DGV04] Adam Dunkels, Bjorn Gronvall, and Thiemo Voigt. Contiki—a lightweight and flexible operating system for tiny networked sensors. In *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pages 455–462. IEEE, 2004.
- [DJMW96] Gregory Dudek, Michael RM Jenkin, Evangelos Milios, and David Wilkes. A taxonomy for multi-agent robotics. *Autonomous Robots*, 3(4):375–397, 1996.
- [DLNS09] Shantanu Das, Hai Liu, Amiya Nayak, and Ivan Stojmenovic. A localized algorithm for bi-connectivity of connected mobile robots. *Telecommunication Systems*, 40(3-4):129–140, 2009.
- [DP10] Walteneus Dargie and Christian Poellabauer. *Fundamentals of wireless sensor networks: theory and practice*. John Wiley & Sons, 2010.
- [DPH07] Saumitra M Das, Himabindu Pucha, and Y Charlie Hu. On the scalability of rendezvous-based location services for geographic wireless ad hoc routing. *Computer Networks*, 51(13):3693–3714, 2007.
- [DPZ04] Richard Draves, Jitendra Padhye, and Brian Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 114–128. ACM, 2004.

- [DS83] Randall Davis and Reid G Smith. Negotiation as a metaphor for distributed problem solving. *Artificial intelligence*, 20(1):63–109, 1983.
- [DS00] M Bernardine Dias and Anthony Stentz. A free market architecture for distributed control of a multirobot system. In *6th International Conference on Intelligent Autonomous Systems (IAS-6)*, pages 115–122, 2000.
- [DZKS06] M Bernardine Dias, Robert Zlot, Nidhi Kalra, and Anthony Stentz. Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE*, 94(7):1257–1270, 2006.
- [EGHK99] Deborah Estrin, Ramesh Govindan, John Heidemann, and Satish Kumar. Next century challenges: Scalable coordination in sensor networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 263–270. ACM, 1999.
- [EGK⁺04] Guy Even, Naveen Garg, Jochen Könemann, R Ravi, and Amitabh Sinha. Min–max tree covers of graphs. *Operations Research Letters*, 32(4):309–315, 2004.
- [FDS⁺02] Rafael Fierro, Aveek Das, John Spletzer, Joel Esposito, Vijay Kumar, James P Ostrowski, George Pappas, Camillo J Taylor, Yerang Hur, Rajeev Alur, et al. A framework and architecture for multi-robot coordination. *The International Journal of Robotics Research*, 21(10-11):977–995, 2002.
- [FGG06] Qing Fang, Jie Gao, and Leonidas J. Guibas. Landmark-based information storage and retrieval in sensor networks. In *In The 25th Conference of the IEEE Communication Society (INFOCOM'06)*, pages 1–12, 2006.
- [FGNW06] Stefan Funke, Leonidas J. Guibas, An Nguyen, and Yusu Wang. Distance-sensitive routing and information brokerage in sensor networks. In *in IEEE International Conference on Distributed Computing in Sensor System (DCOSS'06)*, pages 234–251, 2006.
- [FK05] Roy Friedman and Guy Korland. Timed grid routing (tigr) bites off energy. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 438–448. ACM, 2005.
- [FS06] Hannes Frey and Ivan Stojmenovic. On delivery guarantees of face and combined greedy-face routing in ad hoc and sensor networks. In *Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 390–401. ACM, 2006.

- [Gar85] CW Gardiner. *Stochastic methods*. Springer-Verlag, Berlin–Heidelberg–New York–Tokyo, 1985.
- [GCD⁺03] Dani Goldberg, Vincent Cicirello, M Bernadine Dias, Reid Simmons, Stephen Smith, and Anthony Stentz. Market-based multi-robot planning in a distributed layered architecture. In *Multi-robot systems: From swarms to intelligent automata: Proceedings from the 2003 international workshop on multi-robot systems*, volume 2, pages 27–38, 2003.
- [GD12] K. R. Guruprasad and P. Dasgupta. Distributed voronoi partitioning for multi-robot systems with limited range sensors. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 3546–3552, 2012.
- [GM02a] Brian P Gerkey and Maja J Mataric. A market-based formulation of sensor-actuator network coordination. In *Proceedings of the AAAI Spring Symposium on Intelligent Embedded and Distributed Systems*, pages 21–26, 2002.
- [GM02b] Brian P Gerkey and Maja J Mataric. Sold!: Auction methods for multirobot coordination. *Robotics and Automation, IEEE Transactions on*, 18(5):758–768, 2002.
- [GM04] Brian P Gerkey and Maja J Matarić. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research*, 23(9):939–954, 2004.
- [GP08] Vrinda Gupta and Rajoo Pandey. Data fusion and topology control in wireless sensor networks. *WSEAS Transactions on Signal Processing*, 4(4):150–172, 2008.
- [GS62] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.
- [GS69] K Ruben Gabriel and Robert R Sokal. A new statistical approach to geographic variation analysis. *Systematic Biology*, 18(3):259–278, 1969.
- [GWAC05] Arunabha Ghosh, David R Wolter, Jeffrey G Andrews, and Runhua Chen. Broadband wireless access with wimax/802.16: current performance benchmarks and future potential. *Communications Magazine, IEEE*, 43(2):129–136, 2005.
- [Hil03] Jason Lester Hill. *System architecture for wireless sensor networks*. PhD thesis, University of California, Berkeley, 2003.

- [HKP06] Jiawei Han, Micheline Kamber, and Jian Pei. *Data mining, southeast asia edition: Concepts and techniques*. Morgan Kaufmann, 2006.
- [HMS02] Andrew Howard, Maja J Matarić, and Gaurav S Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *Distributed Autonomous Robotic Systems 5*, pages 299–308. Springer, 2002.
- [HNR68] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics, IEEE Transactions on*, 4(2):100–107, 1968.
- [HV05] Nojeong Heo and Pramod K Varshney. Energy-efficient deployment of intelligent mobile sensor networks. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 35(1):78–92, 2005.
- [JLM11] N. Janicijevic, M. Lukic, and I. Mezei. Routing protocol for low-power and lossy wireless sensor networks. In *Telecommunications Forum (TELFOR), 2011 19th*, pages 234–237. IEEE, 2011.
- [JSAC01] Christine E Jones, Krishna M Sivalingam, Prathima Agrawal, and Jyh Cheng Chen. A survey of energy efficient network protocols for wireless networks. *wireless networks*, 7(4):343–358, 2001.
- [KAS98] Govind Krishnamurthi, Murat Azizoğlu, and Arun K Somani. Optimal location management algorithms for mobile networks. In *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pages 223–232. ACM, 1998.
- [KFHM02] Michael Käsemann, Holger Füßler, Hannes Hartenstein, and Martin Mauve. *A reactive location service for mobile ad hoc networks*. Citeseer, 2002.
- [KKC⁺06] Kyungmi Kim, Hyunsook Kim, Young Choi, Sukgyu Lee, and Kijun Han. A zone-based clustering method for ubiquitous robots based on wireless sensor networks. In Paul Havinga, Maria Lijding, Nirvana Meratnia, and Maarten Wegdam, editors, *Smart Sensing and Context*, volume 4272 of *Lecture Notes in Computer Science*, pages 25–38. Springer Berlin Heidelberg, 2006.
- [KSV05] Jelle R Kok, Matthijs TJ Spaan, and Nikos Vlassis. Non-communicative multi-robot coordination in dynamic environments. *Robotics and Autonomous Systems*, 50(2):99–114, 2005.
- [KT10] Anne-Mari Kermarrec and Guang Tan. Greedy geographic routing in large-scale sensor networks: A minimum network decomposition approach. In *Proceedings of the Eleventh ACM International*

- Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc '10*, pages 161–170, New York, NY, USA, 2010. ACM.
- [Kuh55] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.
- [L⁺04] Franck L Lewis et al. Wireless sensor networks. *Smart environments: technologies, protocols, and applications*, pages 11–46, 2004.
- [LBS15] M. Lukic, A. Barnawi, and I. Stojmenovic. Robot coordination for energy-balanced matching and sequence dispatch of robots to events. *Computers, IEEE Transactions on*, 64(5):1416–1428, May 2015.
- [LJDC⁺00a] Jinyang Li, John Jannotti, Douglas S. J. De Couto, David R. Karger, and Robert Morris. A scalable location service for geographic ad hoc routing. In *Proceedings of the 6th annual international conference on Mobile computing and networking, MobiCom '00*, pages 120–130, New York, NY, USA, 2000. ACM.
- [LJDC⁺00b] Jinyang Li, John Jannotti, Douglas SJ De Couto, David R Karger, and Robert Morris. A scalable location service for geographic ad hoc routing. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 120–130. ACM, 2000.
- [LJGM06] Kristina Lerman, Chris Jones, Aram Galstyan, and Maja Mataric. Analysis of dynamic task allocation in multi-robot systems. *International Journal of Robotics Research*, 25:225–242, 2006.
- [LJM11] M. Lukic, N. Janicijevic, and I. Mezei. Improved decision making in wsn based on localized auctions and fuzzy logic. In *Telecommunications Forum (TELFOR), 2011 19th*, pages 238–241. IEEE, 2011.
- [LKLT12] Chia-Hung Lin, Jian-Jhih Kuo, Bing-Hong Liu, and Ming-Jer Tsai. Gps-free, boundary-recognition-free, and reliable double-ruling-based information brokerage scheme in wireless sensor networks. *IEEE Transactions on Computers*, 61(6):885–898, 2012.
- [LM12] Milan Lukic and Ivan Mezei. Distributed distance sensitive imesh based service discovery in dense wsan. In *Ad-hoc, Mobile, and Wireless Networks*, pages 435–448. Springer Berlin Heidelberg, 2012.
- [LM15] Milan Lukic and Ivan Mezei. Localised querying and location update service in wireless sensor and robot networks with arbitrary topology. *International Journal of Ad Hoc and Ubiquitous Computing*, 2015. (in press).

- [LMK⁺05] Michail G Lagoudakis, Evangelos Markakis, David Kempe, Pinar Keskinocak, Anton J Kleywegt, Sven Koenig, Craig A Tovey, Adam Meyerson, and Sonal Jain. Auction-based multi-robot routing. In *Robotics: Science and Systems*, volume 5, 2005.
- [LMP⁺05] Philip Levis, Sam Madden, Joseph Polastre, Robert Szewczyk, Kamin Whitehouse, Alec Woo, David Gay, Jason Hill, Matt Welsh, Eric Brewer, et al. Tinyos: An operating system for sensor networks. In *Ambient intelligence*, pages 115–148. Springer, 2005.
- [LNS10] Xu Li, Amiya Nayak, and Ivan Stojmenovic. Location service in sensor and mobile actuator networks. *Wireless Sensor and Actuator Networks*, page 209, 2010.
- [LPMS09] Milan Lukic, Bogdan Pavkovic, Nathalie Mitton, and Ivan Stojmenovic. Greedy geographic routing algorithms in real environment. In *Mobile Ad-hoc and Sensor Networks, 2009. MSN'09. 5th International Conference on*, pages 86–93. IEEE, 2009.
- [LS03] Xu Lin and Ivan Stojmenovic. Location-based localized alternate, disjoint and multi-path routing algorithms for wireless networks. *Journal of Parallel and Distributed Computing*, 63(1):22–32, 2003.
- [LS13] Milan Lukic and Ivan Stojmenovic. Energy-balanced matching and sequence dispatch of robots to events: Pairwise exchanges and sensor assisted robot coordination. In *The Tenth IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2013)*, HangZhou, ZheJiang Province, P.R. China, October 2013.
- [LSS07] Xu Li, Nicola Santoro, and Ivan Stojmenovic. Mesh-based sensor relocation for coverage maintenance in mobile sensor networks. In *Ubiquitous Intelligence and Computing*, pages 696–708. Springer, 2007.
- [LSS09] Xu Li, Nicola Santoro, and Ivan Stojmenovic. Localized distance-sensitive service discovery in wireless sensor and actor networks. *IEEE Trans. Comput.*, 58(9):1275–1288, September 2009.
- [Mam12] Quazi Mamun. A qualitative comparison of different logical topologies for wireless sensor networks. *Sensors*, 12(11):14887–14913, 2012.
- [MB01] Brent A Miller and Chatschik Bisdikian. *Bluetooth revealed: the insider's guide to an open specification for global wireless communication*. Prentice Hall PTR, 2001.

- [MBB09] A.N. Mian, R. Baldoni, and R. Beraldi. A survey of service discovery protocols in multihop mobile ad hoc networks. *Pervasive Computing, IEEE*, 8(1):66–74, Jan.-March 2009.
- [MKPS05] Seapahn Megerian, Farinaz Koushanfar, Miodrag Potkonjak, and Mani B Srivastava. Worst and best-case coverage in sensor networks. *Mobile Computing, IEEE Transactions on*, 4(1):84–92, 2005.
- [MKSL04] Miklós Maróti, Branislav Kusy, Gyula Simon, and Ákos Lédeczi. The flooding time synchronization protocol. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 39–49. ACM, 2004.
- [MLM14] Ivan Mezei, Milan Lukic, and Veljko Malbasa. *Wireless Sensor and Robot Networks*, chapter 3: Robot-Robot Coordination, pages 51–69. World Scientific, 2014.
- [MLMS13] Ivan Mezei, Milan Lukic, Veljko Malbasa, and Ivan Stojmenovic. Auctions and imesh based task assignment in wireless sensor and actuator networks. *Computer Communications*, 36(9):979–987, 2013.
- [MMS10] Ivan Mezei, Veljko Malbasa, and Ivan Stojmenovic. Robot to robot. *Robotics & Automation Magazine, IEEE*, 17(4):63–69, 2010.
- [MPGA07] Tommaso Melodia, Dario Pompili, Vehbi C Gungor, and Ian F Akyildiz. Communication and coordination in wireless sensor and actor networks. *Mobile Computing, IEEE Transactions on*, 6(10):1116–1129, 2007.
- [MWH01] Martin Mauve, Jorg Widmer, and Hannes Hartenstein. A survey on position-based routing in mobile ad hoc networks. *Network, IEEE*, 15(6):30–39, 2001.
- [NK13] S Nithyakalyani and S Suresh Kumar. An approach to data aggregation in wireless sensor network using voronoi fuzzy clustering algorithm. *Journal of Scientific & Industrial Research*, 72(5):287–293, 2013.
- [NNY89] Yoshihiko Nakamura, Kiyoshi Nagai, and Tsuneo Yoshikawa. Dynamics and stability in coordination of multiple robotic mechanisms. *The International Journal of Robotics Research*, 8(2):44–61, 1989.
- [NS10] Amiya Nayak and Ivan Stojmenovic. Wireless sensor and actuator networks. *Algorithms and Protocols for Scalable Coordination and Data Communication* WILEY, 2010.
- [OP05] Bob O’hara and Al Petrick. *IEEE 802.11 handbook: a designer’s companion*. IEEE Standards Association, 2005.

- [PA05] Jufeng Peng and Srinivas Akella. Coordinating multiple robots with kinodynamic constraints along specified paths. *The International Journal of Robotics Research*, 24(4):295–310, 2005.
- [Pap03] Christos H Papadimitriou. *Computational complexity*. John Wiley and Sons Ltd., 2003.
- [Par98] Lynne E Parker. Alliance: An architecture for fault tolerant multirobot cooperation. *Robotics and Automation, IEEE Transactions on*, 14(2):220–240, 1998.
- [Par08] Lynne E Parker. Multiple mobile robot systems. In *Springer Handbook of Robotics*, pages 921–941. Springer, 2008.
- [Per98] Charles E Perkins. Mobile networking through mobile ip. *Internet Computing, IEEE*, 2(1):58–69, 1998.
- [PHS97] R Prakash, ZJ Haas, and M Singhal. Load balanced location management for mobile systems using dynamic hashing and quorums. *submitted for publication*, 1997.
- [Pol04] Daniel Polani. *RoboCup 2003: Robot Soccer World Cup VII*, volume 3020. Springer Science & Business Media, 2004.
- [PRM⁺12] Bogdan Pavkovic, Jovan Radak, Nathalie Mitton, Franck Rousseau, Ivan Stojmenovic, et al. Emulation of large scale wireless sensor networks: From real neighbors to imaginary destination. In *11th International Conference on Ad-Hoc Networks and Wireless*, 2012.
- [PY90] Suparek Premvuti and S Yuta. Consideration on the cooperation of multiple autonomous mobile robots. In *Intelligent Robots and Systems' 90.'Towards a New Frontier of Applications', Proceedings. IROS'90. IEEE International Workshop on*, pages 59–63. IEEE, 1990.
- [RB07] Martijn N Rooker and Andreas Birk. Multi-robot exploration under the constraints of wireless networking. *Control Engineering Practice*, 15(4):435–445, 2007.
- [RKY⁺02] Sylvia Ratnasamy, Brad Karp, Li Yin, Fang Yu, Deborah Estrin, Ramesh Govindan, and Scott Shenker. Ght: a geographic hash table for data-centric storage. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, WSNA '02, pages 78–87, New York, NY, USA, 2002. ACM.
- [RM98] V. Rodoplu and T.H. Meng. Minimum energy mobile wireless networks. In *Communications, 1998. ICC 98. Conference Record. 1998 IEEE International Conference on*, volume 3, pages 1633–1639 vol.3, Jun 1998.

- [RMO93] K Ahuja Ravindra, Thomas L Magnanti, and James B Orlin. Network flows: theory, algorithms, and applications, 1993.
- [RS08] Wei Ren and Nathan Sorensen. Distributed coordination architecture for multi-robot formation control. *Robotics and Autonomous Systems*, 56(4):324–333, 2008.
- [RSZ04] Cauligi S Raghavendra, Krishna M Sivalingam, and Taieb Znati. *Wireless sensor networks*. Springer Science & Business Media, 2004.
- [SBK05] Bharath Sundararaman, Ugo Buy, and Ajay D Kshemkalyani. Clock synchronization for wireless sensor networks: a survey. *Ad Hoc Networks*, 3(3):281–323, 2005.
- [SCS06] Xingfa Shen, Jiming Chen, and Youxian Sun. Grid scan: A simple and effective approach for coverage issue in wireless sensor networks. In *Communications, 2006. ICC'06. IEEE International Conference on*, volume 8, pages 3480–3484. IEEE, 2006.
- [SKK13] Shamneesh Sharma, Dinesh Kumar, and Keshav Kishore. Wireless sensor networks- a review on topologies and node architecture. *International Journal of Computer Sciences and Engineering*, 1(2):19–25, 2013.
- [SLJ08] Ivan Stojmenovic, Dandan Liu, and Xiaohua Jia. A scalable quorum based location service in ad hoc and sensor networks. *Int. J. Commun. Netw. Distrib. Syst.*, 1(1):71–94, February 2008.
- [SMZ07] Kazem Sohraby, Daniel Minoli, and Taieb Znati. *Wireless sensor networks: technology, protocols, and applications*. John Wiley & Sons, 2007.
- [SN06] Milos Stojmenovic and Amiya Nayak. Localized routing with guaranteed delivery and a realistic physical layer in wireless sensor networks. *Computer Communications*, 29(13-14):2550–2555, August 2006.
- [SRL06] Ivan Stojmenovic, Anand Prakash Ruhil, and DK Lobiyal. Voronoi diagram and convex hull based geocasting and routing in wireless networks. *Wireless communications and mobile computing*, 6(2):247–258, 2006.
- [SSNJ06] David Sweetser, Victoria Sweetser, and Jay Nemeth-Johannes. A modular approach to ieee-1451.5 wireless sensor development. In *Sensors Applications Symposium, 2006. Proceedings of the 2006 IEEE*, pages 82–87. IEEE, 2006.

- [Sto99] Ivan Stojmenovic. Home agent based location update and destination search schemes in ad hoc wireless networks. *Computer Science, SITE, University of Ottawa, TR-99-10*, 1999.
- [Sto05] Ivan Stojmenovic. *Handbook of sensor networks: algorithms and architectures*, volume 49. John Wiley & Sons, 2005.
- [SX07] A. Shrestha and Liudong Xing. A performance comparison of different topologies for wireless sensor networks. In *Technologies for Homeland Security, 2007 IEEE Conference on*, pages 280–285, May 2007.
- [SY13] Tien-Wen Sung and Chu-Sing Yang. Distributed voronoi-based self-redeployment for coverage enhancement in a mobile directional sensor network. *International Journal of Distributed Sensor Networks*, 2013, 2013.
- [SYTX06] Weihua Sheng, Qingyan Yang, Jindong Tan, and Ning Xi. Distributed multi-robot coordination in area exploration. *Robotics and Autonomous Systems*, 54(12):945–955, 2006.
- [SZG09] Rik Sarkar, Xianjin Zhu, and Jie Gao. Double rulings for information brokerage in sensor networks. *IEEE/ACM Trans. Netw.*, 17(6):1902–1915, December 2009.
- [TBK09] Guang Tan, M. Bertier, and A.-M. Kermarrec. Convex partition of sensor networks and its use in virtual coordinate geographic routing. In *INFOCOM 2009, IEEE*, pages 1746–1754, April 2009.
- [TJZ⁺13] Guang Tan, Hongbo Jiang, Shengkai Zhang, Zhimeng Yin, and Anne-Marie Kermarrec. Connectivity-based and anchor-free localization in large-scale 2d/3d sensor networks. *ACM Trans. Sen. Netw.*, 10(1):6:1–6:21, December 2013.
- [TLJK05] Craig Tovey, Michail G Lagoudakis, Sonal Jain, and Sven Koenig. The generation of bidding rules for auction-based robot coordination. In *Multi-Robot Systems. From Swarms to Intelligent Automata Volume III*, pages 3–14. Springer, 2005.
- [TP07] Fang Tang and L.E. Parker. A complete methodology for generating multi-robot task solutions using asymptre-d and market-based task allocation. In *IEEE International Conference on Robotics and Automation*, pages 3351–3358, 2007.
- [TV04] Jivodar B Tchakarov and Nitin H Vaidya. Efficient content location in wireless ad hoc networks. In *Mobile Data Management, 2004. Proceedings. 2004 IEEE International Conference on*, pages 74–85. IEEE, 2004.

- [VDA13] Aline Carneiro Viana and Marcelo Dias De Amorim. Coverage strategy for periodic readings in robotic-assisted monitoring systems. *Ad Hoc Networks*, 11(7):1907–1918, 2013.
- [VDdA08] Aline Carneiro Viana and Marcelo Dias de Amorim. Sensing and acting with predefined trajectories. In *Proceedings of the 1st ACM international workshop on Heterogeneous sensor and actor networks*, pages 1–8. ACM, 2008.
- [VDL03] Tijs Van Dam and Koen Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 171–180. ACM, 2003.
- [VDMC10] Roberto Verdone, Davide Dardari, Gianluca Mazzini, and Andrea Conti. *Wireless sensor and actuator networks: technologies, analysis and design*. Academic Press, 2010.
- [Ver05] S Verret. Current state of the art in multirobot systems. *Defence Research and Development Canada-Suffield*, page 3, 2005.
- [VP08] C.N. Ververidis and G.C. Polyzos. Service discovery for mobile ad hoc networks: a survey of issues and techniques. *Communications Surveys Tutorials, IEEE*, 10(3):30–45, 2008.
- [W⁺01] Douglas Brent West et al. *Introduction to graph theory*, volume 2. Prentice Hall Upper Saddle River, 2001.
- [WCLP06] Guiling Wang, Guohong Cao, and Tom La Porta. Movement-assisted sensor deployment. *Mobile Computing, IEEE Transactions on*, 5(6):640–652, 2006.
- [WHY06] Qin Wang, Mark Hempstead, and Woodward Yang. A realistic power consumption model for wireless sensor network devices. In *Sensor and Ad Hoc Communications and Networks, 2006. SECON'06. 2006 3rd Annual IEEE Communications Society on*, volume 1, pages 286–295. IEEE, 2006.
- [WPT10] You-Chiun Wang, Wen-Chih Peng, and Yu-Chee Tseng. Energy-balanced dispatch of mobile sensors in a hybrid wireless sensor network. *IEEE Transactions on Parallel and Distributed Systems*, 21:1836–1850, 2010.
- [WSRABD07] Thomas Watteyne, David Simplot-Ryl, Isabelle Augé-Blum, and Mischa Dohler. On using virtual coordinates for routing in the context of wireless sensor networks. In *Personal, Indoor and Mobile*

- Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*, pages 1–5. IEEE, 2007.
- [WWL13] Jiagao Wu, Yinan Wang, and Linfeng Liu. A voronoi-based depth-adjustment scheme for underwater wireless sensor networks. *International Journal on Smart Sensing And Intelligent Systems*, 6(1):244–258, 2013.
- [Xia08] Feng Xia. Qos challenges and opportunities in wireless sensor/actuator networks. *Sensors*, 8(2):1099–1110, 2008.
- [XTH06] Kenan Xu, Glen Takahara, and Hossam Hassanein. On the robustness of grid-based deployment in wireless sensor networks. In *Proceedings of the 2006 international conference on Wireless communications and mobile computing*, pages 1183–1188. ACM, 2006.
- [YA08] Mohamed Younis and Kemal Akkaya. Strategies and techniques for node placement in wireless sensor networks: A survey. *Ad Hoc Networks*, 6(4):621–655, 2008.
- [YG04] Liuqing Yang and Georgios B Giannakis. Ultra-wideband communications: an idea whose time has come. *Signal Processing Magazine, IEEE*, 21(6):26–54, 2004.
- [YGE01] Yan Yu, Ramesh Govindan, and Deborah Estrin. Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks. Technical report, Technical report ucla/csd-tr-01-0023, UCLA Computer Science Department, 2001.
- [YHE02] Wei Ye, John Heidemann, and Deborah Estrin. An energy-efficient mac protocol for wireless sensor networks. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1567–1576. IEEE, 2002.
- [YKR06] Ossama Younis, Marwan Krunz, and Srinivasan Ramasubramanian. Node clustering in wireless sensor networks: recent developments and deployment challenges. *Network, IEEE*, 20(3):20–25, 2006.
- [ZC03] Yi Zou and Krishnendu Chakrabarty. Sensor deployment and target localization based on virtual forces. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, pages 1293–1303. IEEE, 2003.
- [ZJW13] Hongyu Zhou, Miao Jin, and Hongyi Wu. A distributed delaunay triangulation algorithm based on centroidal voronoi tessellation

- for wireless sensor networks. In *Proceedings of the Fourteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '13, pages 59–68, New York, NY, USA, 2013. ACM.
- [ZMN05] Fen Zhu, Matt W Mutka, and Lionel M Ni. Service discovery in pervasive computing environments. *IEEE Pervasive computing*, 4(4):81–90, 2005.
- [ZS14] Tieying Zhu and Milica Stojmenovic. Social-based routing in opportunistic mobile networks: A survey. *Ad Hoc and Sensor Wireless Networks*, 21(1-2):1 – 20, 2014.
- [ZSDT02] Robert Zlot, Anthony Stentz, M Bernardine Dias, and Scott Thayer. Multi-robot exploration controlled by a market economy. 2002.