

UNIVERZITET U NOVOM SADU  
FAKULTET TEHNIČKIH NAUKA

mr Ivan Mezei

Aukcijski agregacioni algoritmi za izbor izvršioca u  
bežičnim muti-hop mrežama elektronskih senzora i  
aktuatora

-doktorska disertacija-

Novi Sad, 2012.

Mentor: **Dr Ivan Stojmenović**, redovni profesor  
Fakultet tehničkih nauka, Novi Sad i  
SEECs, Univerzitet u Otavi, Kanada

Komisija za javnu odbranu: **Dr Veljko Malbaša**, redovni profesor  
Fakultet tehničkih nauka, Novi Sad  
**Dr Veljko Milutinović**, redovni profesor  
Elektrotehnički fakultet, Beograd  
**Dr Vojin Šenk**, redovni profesor  
Fakultet tehničkih nauka, Novi Sad  
**Dr Staniša Dautović**, docent  
Fakultet tehničkih nauka, Novi Sad

---

Dr Ivan Stojmenović

---

Dr Veljko Malbaša

---

Dr Veljko Milutinović

---

Dr Vojin Šenk

---

Dr Staniša Dautović

## Ključna dokumentacijska informacija

Редни број, <b>РБР:</b>												
Идентификациони број, <b>ИБР:</b>												
Тип документације, <b>ТД:</b>	Монографска публикација											
Тип записа, <b>ТЗ:</b>	Текстуални штампани материјал											
Врста рада, <b>ВР:</b>	Докторска дисертација											
Аутор, <b>АУ:</b>	Мр Мезеи Иван											
Ментор, <b>МН:</b>	Др Иван Стојменовић, ред. проф.											
Наслов рада, <b>НР:</b>	Аукцијски агрегациони алгоритми за избор извршиоца у бежичним мулти-хоп мрежама електронских сензора и актуатора											
Језик публикације, <b>ЈП:</b>	Српски											
Језик извода, <b>ЈИ:</b>	Српски											
Земља публикавања, <b>ЗП:</b>	Србија											
Уже географско подручје, <b>УГП:</b>	Србија/Војводина											
Година, <b>ГО:</b>	2012.											
Издавач, <b>ИЗ:</b>	Ауторски репринт											
Место и адреса, <b>МА:</b>	Факултет Техничких Наука, Трг Доситеја Обрадовића 6, 21000 Нови Сад											
Физички опис рада, <b>ФО:</b> (поглавља/страна/ цитата/табела/слика/графика/прилога)	9 / 111 / 110 / 5 / 51 / - / 3											
Научна област, <b>НО:</b>	Електротехника											
Научна дисциплина, <b>НД:</b>	Електроника											
Предметна одредница/Кључне речи, <b>ПО:</b>	аукцијски агрегациони алгоритми, избор извршиоца, откривање сервиса, откривање локације извршиоца, бежичне сензорске и актуаторске мреже											
<b>УДК</b>												
Чува се, <b>ЧУ:</b>	Библиотека Факултета Техничких Наука, Трг Доситеја Обрадовића 6, 21000 Нови Сад											
Важна напомена, <b>ВН:</b>												
Извод, <b>ИЗ:</b>	<p>Полазећи од простог аукцијског алгоритма (САП) за избор извршиоца у бежичним сензорским и актуаторским мрежама у овој дисертацији су уведена два побољшања тог алгоритма (мулти-хоп комуникација и локализација). Предложено је и пет нових аукцијских агрегационих алгоритама који користе технику агрегације да би смањили комуникационе трошкове. Трећа група доприноса ове дисертације се односи на примену аукција за побољшање постојећег iMesh алгоритма за проналажење извршиоца. Експериментални резултати показују својства алгоритама и њихове предности у односу на постојеће.</p>											
Датум прихватања теме, <b>ДП:</b>	23. децембар 2010.											
Датум одбране, <b>ДО:</b>	јули 2012.											
Чланови комисије, <b>КО:</b>	<table border="1"> <tr> <td>Председник:</td> <td>Др Вељко Малбаша, ред. проф., ФТН, Нови Сад</td> <td rowspan="5">Потпис ментора</td> </tr> <tr> <td>Члан:</td> <td>Др Вељко Милутиновић, ред. проф. ЕТФ, Београд</td> </tr> <tr> <td>Члан:</td> <td>Др Војин Шенк, ред. проф., ФТН, Нови Сад</td> </tr> <tr> <td>Члан:</td> <td>Др Станиша Даутовић, доцент, ФТН, Нови Сад</td> </tr> <tr> <td>Члан, ментор:</td> <td>Др Иван Стојменовић, ред. проф., ФТН, Нови Сад</td> </tr> </table>	Председник:	Др Вељко Малбаша, ред. проф., ФТН, Нови Сад	Потпис ментора	Члан:	Др Вељко Милутиновић, ред. проф. ЕТФ, Београд	Члан:	Др Војин Шенк, ред. проф., ФТН, Нови Сад	Члан:	Др Станиша Даутовић, доцент, ФТН, Нови Сад	Члан, ментор:	Др Иван Стојменовић, ред. проф., ФТН, Нови Сад
Председник:	Др Вељко Малбаша, ред. проф., ФТН, Нови Сад	Потпис ментора										
Члан:	Др Вељко Милутиновић, ред. проф. ЕТФ, Београд											
Члан:	Др Војин Шенк, ред. проф., ФТН, Нови Сад											
Члан:	Др Станиша Даутовић, доцент, ФТН, Нови Сад											
Члан, ментор:	Др Иван Стојменовић, ред. проф., ФТН, Нови Сад											

Accession number, <b>ANO</b> :												
Identification number, <b>INO</b> :												
Document type, <b>DT</b> :	Monographic publication											
Type of record, <b>TR</b> :	Textual material, printed											
Contents code, <b>CC</b> :	PhD Thesis											
Author, <b>AU</b> :	Mezei Ivan, M.Sc.											
Mentor, <b>MN</b> :	Dr Ivan Stojmenović, Professor											
Title, <b>TI</b> :	Auction Agregation Algorithms for Task Assignment in Wireless Multihop Electronic Sensor and Actuator Networks											
Language of text, <b>LT</b> :	Serbian											
Language of abstract, <b>LA</b> :	Serbian											
Country of publication, <b>CP</b> :	Serbia											
Locality of publication, <b>LP</b> :	Serbia/Vojvodina											
Publication year, <b>PY</b> :	2012											
Publisher, <b>PB</b> :	Author's reprint											
Publication place, <b>PP</b> :	Faculty of Technical Sciences, Trg Dositeja Obradovića 6, 21000 Novi Sad											
Physical description, <b>PD</b> : (chapters/pages/ref./tables/pictures/graphs/appendixes)	9 / 111 / 110 / 5 / 51 / - / 3											
Scientific field, <b>SF</b> :	Electrical engineering											
Scientific discipline, <b>SD</b> :	Electronics											
Subject/Key words, <b>S/KW</b> :	auction aggregation algorithms, task assignment, service discovery, location discovery, wireless sensor and actuator networks											
<b>UC</b>												
Holding data, <b>HD</b> :	Library of the Faculty of Technical Sciences, Trg Dositeja Obradovića 6, 21000 Novi Sad											
Note, <b>N</b> :												
Abstract, <b>AB</b> :	Two improvements of the simple auction protocol (SAP) for the task assignment in wireless sensor and actuator networks is proposed in this dissertation (multi-hop communications and localization). Five new auction aggregation algorithms for the task assignment are proposed with the goal to minimize the communication costs. One of the auction aggregation algorithms is also used to improve existing iMesh service discovery algorithm. Experimental results show the characteristics of the proposed algorithms and advantages over the existing ones.											
Accepted by the Scientific Board on, <b>ASB</b> :	23.12.2010.											
Defended on, <b>DE</b> :	july 2012.											
Defended Board, <b>DB</b> :	<table border="1"> <tr> <td>President:</td> <td>Dr Veljko Malbaša, professor, FTN, Novi Sad</td> <td rowspan="5">Menthor's sign</td> </tr> <tr> <td>Member:</td> <td>Dr Veljko Milutinović, professor, ETF, Beograd</td> </tr> <tr> <td>Member:</td> <td>Dr Vojin Šenk, professor, FTN, Novi Sad</td> </tr> <tr> <td>Member:</td> <td>Dr Staniša Dautović, assis. prof., FTN, Novi Sad</td> </tr> <tr> <td>Member, Mentor:</td> <td>Dr Ivan Stojmenović, professor, FTN, Novi Sad</td> </tr> </table>	President:	Dr Veljko Malbaša, professor, FTN, Novi Sad	Menthor's sign	Member:	Dr Veljko Milutinović, professor, ETF, Beograd	Member:	Dr Vojin Šenk, professor, FTN, Novi Sad	Member:	Dr Staniša Dautović, assis. prof., FTN, Novi Sad	Member, Mentor:	Dr Ivan Stojmenović, professor, FTN, Novi Sad
President:	Dr Veljko Malbaša, professor, FTN, Novi Sad	Menthor's sign										
Member:	Dr Veljko Milutinović, professor, ETF, Beograd											
Member:	Dr Vojin Šenk, professor, FTN, Novi Sad											
Member:	Dr Staniša Dautović, assis. prof., FTN, Novi Sad											
Member, Mentor:	Dr Ivan Stojmenović, professor, FTN, Novi Sad											

Obrazac Q4.09.21 - Izdanje 1

## Sadržaj

Lista skraćenica .....	7
Lista slika.....	8
Lista tabela.....	10
Rezime.....	11
1 Uvod .....	14
1.1 Bežične senzorske mreže.....	14
1.2 Multirobotski sistemi.....	16
1.3 Multi-hop bežične senzorske i aktuatorске mreže.....	17
1.4 Definicija problema izbora izvršioca i dodele posla .....	21
1.5 Postojeća rešenja.....	26
1.6 Aukcije.....	27
1.7 Nova rešenja ili poboljšanja postojećih.....	28
1.8 Mrežni model i parametri simulacija.....	30
1.9 Doprinosi disertacije.....	31
1.10 Pregled disertacije.....	33
2 Pregled literature.....	35
2.1 Problem izbora izvršioca u slučaju više izvršilaca .....	35
2.2 Različiti metodi za izbor izvršioca .....	38
2.3 Otkrivanje lokacije izvršioca .....	41
3 Izbor izvršioca – idealne komunikacije .....	44
3.1 Pojedinačni događaji .....	44
3.2 Višestruki događaji .....	46
3.2.1 Pojedinačna dodela zadatka – pojedinačni izvršioci .....	47
3.2.2 Višestruka dodela zadataka – pojedinačni izvršioci .....	50
3.2.3 Pojedinačna dodela zadatka – zadaci koji zahtevaju više izvršilaca .....	51
3.2.4 Višestruka dodela zadataka – zadaci koji zahtevaju više izvršilaca .....	52
4 Izbor izvršioca – nezanemarljivi komunikacioni troškovi .....	53
4.1 Pojedinačni događaji .....	53
4.2 Višestruki događaji .....	57
5 Unapređenja prostog aukcijskog algoritma .....	59
5.1 Lokalizovani aukcijski algoritam ( <i>k</i> -SAP) .....	60
5.2 Proširenje lokalizovanog aukcijskog algoritma tzv. pohlepnim ( <i>greedy</i> ) algoritmom ( <i>k</i> -SAPG) .....	64
6 Aukcijski agregacioni algoritmi.....	70
6.1 Prosti aukcijski agregacioni algoritam (SAAP) .....	71
6.2 Lokalizovani prosti aukcijski agregacioni algoritam ( <i>k</i> -SAAP).....	74
6.3 Proširenje lokalizovanog prostog aukcijskog agregacionog algoritma tzv. pohlepnim ( <i>greedy</i> ) algoritmom ( <i>k</i> -SAAPG) .....	77
6.4 Aukcijski agregacioni algoritam (AAP) .....	80
6.5 Lokalizovani aukcijski agregacioni algoritam ( <i>k</i> -AAP).....	81

7	Primena aukcija za poboljšanje <i>iMesh</i> algoritma za pronalaženje izvršioca.	87
7.1	<i>iMesh</i> algoritam	87
7.2	Poboljšanje korišćenjem <i>k</i> -SAAP algoritma	90
8	Zaključak i budući rad	96
9	Literatura	103
	Dodaci	109
	D1 Primeri povezanih pseudo slučajnih grafova mreža izvršilaca generisanih na bazi MIN-DPA algoritma	109
	D2 Kompletan globalni simulacioni algoritam za SAP, <i>k</i> -SAP, SAAP, <i>k</i> -SAAP, <i>k</i> -AAP, <i>k</i> -SAPG i <i>k</i> -SAAPG algoritme	110
	D3 Kompletan globalni simulacioni algoritam za za nadgradnju <i>iMesh</i> algoritma upotrebom <i>k</i> -SAP i <i>k</i> -SAAP aukcijskih algoritama	111

## Lista skraćenica

AAP – aukcijski agregacioni algoritam  
ADC – analogno-digitalni konvertor  
BSAM – bežične senzorske i aktuatorске mreže  
BSM – bežične senzorske mreže  
BSRM – bežične senzorske i robotske mreže  
GAP – *generalized assignment problem*  
GCLP – *geography based content location protocol*  
GHT – geografska heš tabela  
GLS – *grid location service* algoritam  
GPS – globalni sistem pozicioniranja  
k-AAP – lokalizovani aukcijski agregacioni algoritam  
k-SAP – lokalizovani prosti aukcijski algoritam  
k-SAPG – lokalizovani prosti aukcijski algoritam proširen tzv. pohlepnim algoritmom  
k-SAAP – lokalizovani prosti aukcijski agregacioni algoritam  
k-SAAPG – lokalizovani prosti aukcijski agregacioni algoritam proširen tzv. pohlepnim algoritmom  
MARA – *multi-agent resource allocation*  
MIN-DPA – *minimum degree proximity algorithm*  
MINLP – *mixed integer non-linear programming*  
MMA – *maximum matching algorithm*  
MMRS – mobilni multi-robotski sistemi  
MRS – multi-robotski sistemi  
MRTA – *multi-robot task allocation*  
MTA-MAT – *multiple task assignment-multiple actuator tasks*  
MTAP – *multi-actuator task allocation problem*  
MTA-SAT – *multiple task assignment-single actuator tasks*  
OAP – *optimal assignment problem*  
RFTA – *routing with face traversal auctions*  
SAAP – prosti aukcijski agregacioni algoritam  
SAP – prosti aukcijski algoritam  
SCD – *selected to closest distance*  
SCI – *science citation index*  
STA-MAT – *single task assignment-multiple actuator tasks*  
STA-SRT – *single task assignment-single actuator tasks*  
UAV – *unmanned aerial vehicles*  
UDG – *unit disc graph*

## Lista slika

- 1.1.1 Tipična bežična senzorska mreža
- 1.1.2 Arhitektura senzorskih čvorova
- 1.3.1 Ilustracija neautomatizovane BSAM-e
- 1.3.2 Ilustracija poluautomatizovane BSAM-e
- 1.3.3 Ilustracija automatizovane BSAM-e
- 1.3.4 Ilustracija multi-hop komunikacije koja polazi od čvora A1
- 1.3.5 Ilustracija komunikacionih paradigmi u BSAM
- 1.4.1 Primer nepovezane mreže izvršilaca (roboti)
- 1.4.2 Senzorska mreža sa tzv. rupom otežava pronalaženje izvršioca
- 1.4.3 Tri paralelna događaja i različite mogućnosti dodele posla
- 4.1.1 Konstrukcija kvoruma i način pretraživanja
- 4.1.2 iMesh struktura sa 5 izvršilaca i žutim označenim zonama znanja
- 4.1.3 Arhitektura na bazi agenata (senzori se ne vide već samo izvršioci - roboti)
- 5.1 Ilustracija zasebnih povratnih putanja svakog aktuatora (SAP algoritam)
- 5.1.1 Ilustracija 2-SAP algoritma i prednosti u odnosu na SAP
- 5.1.2 Poređenje optimalnosti izbora izvršioca  $k$ -SAP algoritma u odnosu na SAP u funkciji  $k$
- 5.1.3 Poređenje optimalnosti izbora izvršioca  $k$ -SAP algoritma u odnosu na SAP u funkciji gustine mreže  $d$
- 5.1.4 Poređenje komunikacionih troškova SAP i  $k$ -SAP algoritma u funkciji  $k$
- 5.1.5 Poređenje komunikacionih troškova SAP i  $k$ -SAP algoritma zavisno od gustine mreže
- 5.2.1 Unapređenje  $k$ -SAP algoritma na bazi tzv. pohlepnog algoritma
- 5.2.2 Poređenje procenta optimalnog odabira izvršioca za SAP,  $k$ -SAP i  $k$ -SAPG u funkciji od  $k$
- 5.2.3 Poređenje procenta optimalnog odabira izvršioca za SAP,  $k$ -SAP i  $k$ -SAPG u funkciji od gustine mreže
- 5.2.4 Poređenje prosečnog broja poruka po izvršiocu za SAP,  $k$ -SAP i  $k$ -SAPG u funkciji od  $k$
- 5.2.5 Poređenje prosečnog broja poruka po izvršiocu za SAP,  $k$ -SAP i  $k$ -SAPG u funkciji od gustine mreže
- 5.2.6 Prosečan odnos udaljenosti izabranog i najbližeg izvršioca u funkciji od  $k$  za SAP,  $k$ -SAP i  $k$ -SAPG
- 5.2.7 Prosečan odnos udaljenosti izabranog i najbližeg izvršioca u funkciji od gustine mreže za SAP,  $k$ -SAP i  $k$ -SAPG
- 6.1.1 Ilustracija aukcijske agregacije kod SAAP algoritma
- 6.1.2 Za guste i male mreže SAAP ne donosi skoro nikakvo poboljšanje
- 6.1.3 Za velike, a posebno velike i retke mreže SAAP donosi značajno smanjenje komunikacionih troškova
- 6.2.1 poređenje komunikacionih troškova SAP,  $k$ -SAP i  $k$ -SAAP algoritama
- 6.3.1 Poređenje prosečnog broja poruka po izvršiocu za svih pet do sada obrađenih algoritama u funkciji od  $k$
- 6.3.2 Poređenje prosečnog broja poruka po izvršiocu za svih pet do sada obrađenih algoritama u funkciji od gustine mreže



- 6.5.1 Ilustracija 2-AAP algoritma
- 6.5.2  $k$ -AAP daje 100% optimalnost izbora izvršioca za male i retke mreže
- 6.5.3 Ilustracija poboljšanja optimalnosti izbora izvršioca koja se dobija sa  $k$ -AAP algoritmom
- 6.5.4 Uticaj gustine mreže na procenat optimalnosti izbora izvršioca
- 6.5.5 Komunikacioni troškovi  $k$ -AAP za male i retke mreže
- 6.5.6  $k$ -AAP ima dobre osobine malog komunikacionog opterećenja i za velike i guste mreže
- 6.5.7 Prosečan SCD u funkciji od  $k$
- 6.5.8 Prosečan SCD u funkciji gustine mreže  $d$
- 7.1.1 Ilustracija *iMesh* algoritma
- 7.1.2 Scenario u kojem se ne pronalazi najbliži izvršilac
- 7.2.1 Efikasnost pronalaženja optimalnog izvršioca za prvi scenario u zavisnosti od broja hopova
- 7.2.2 Efikasnost pronalaženja optimalnog izvršioca za prvi scenario u zavisnosti od gustine mreže
- 7.2.3 Efikasnost pronalaženja optimalnog izvršioca za drugi scenario u zavisnosti od broja hopova pri promenama broja izvršilaca
- 7.2.4 Efikasnost pronalaženja optimalnog izvršioca za drugi scenario u zavisnosti od broja hopova pri promenama komunikacionog radijusa
- 7.2.5 Prosečan broj potrebnih poruka za jedan metar skraćanja pređenog puta izvršilaca za prvi scenario
- 7.2.6 Prosečan broj potrebnih poruka za jedan metar skraćanja pređenog puta izvršilaca za drugi scenario
- 8.1 Ilustracija odabira određenog podskupa izvršilaca koji čine 'kičmu' mreže
- 8.2 Ilustracija RFTA algoritma za Gabrielov graf po UDG modelu
- 8.3 iMesh struktura za scenario sa 5 izvršilaca sa proizvoljnom ali uniformnom strukturom i gustom mrežom

## **Lista tabela**

1.4.1. Vrste događaja, zadataka, dodele posla i rešenja

3.2.1 Primeri dodele posla

5.1.1 Poređenje komunikacionih troškova za velike i retke mreže

6.1 Procenat izbora najbližeg izvršioca kao kolektora probabilističkom metodom

D.1 Primeri povezanih pseudo slučajnih grafova mreža izvršilaca generisanih na bazi MIN-DPA algoritma

## Rezime

U oblasti bežičnih senzorskih i aktuatorskih mreža jedan od važnih aspekata funkcionisanja mreže je i koordinacija, a u okviru koordinacije izbor izvršioca – aktuatora koji će da reaguje na događaj koji je neki od senzora detektovao i javio nekom od izvršilaca. Postavlja se pitanje koji je izvršilac najbolji da reaguje na dati događaj i kako da se uradi odlučivanje. U ovoj disertaciji se podrazumeva da je najbolji onaj izvršilac koji je najbliži. Ovo je jedna od najčešćih metrika, mada postoje i druge (npr. maksimizacija rezidualne energije, balansiranje utrošene energije među izvršiocima itd.).

Kao sredstvo odlučivanja se već dugo u neekonomskim oblastima koriste i aukcije. Prosti aukcijski algoritam (SAP), gde jedan izvršilac organizuje aukciju i očekuje odgovore od svih ostalih u vidu licitacije sa cenom ako bi on reagovao na događaj, se već dugo koristi i jedan je od prvih algoritama u oblasti bežičnih senzorskih i aktuatorskih mreža koji je uveo aukcije u oblast odlučivanja o izboru izvršioca. U oblasti multirobotskih sistema postoji sličan algoritam pod nazivom *Murdoch*. SAP i *Murdoch* algoritmi su inicijalno zamišljeni da rade u mrežama gde izvršioci formiraju tzv. kompletan graf. Drugim rečima, to znači da svaki izvršilac može direktno da komunicira sa svakim drugim izvršiocom. Ovako jaka pretpostavka ograničava upotrebu ovog algoritma na male mreže ili podrazumeva veoma veliki komunikacioni domet izvršilaca.

U ovoj disertaciji prvo uvodimo sledeća poboljšanja SAP algoritma. Prvo poboljšanje, odnosno relaksacija prethodno navedene pretpostavke, urađena je uvođenjem multi-hop komunikacije među izvršiocima. Zatim su istražene osobine lokalizovanog prostog aukcijskog algoritma ( $k$ -SAP) kod kojeg izvršioci mogu da komuniciraju međusobno ukoliko su udaljeni do  $k$  hopova. Pokazuje se da se ovim značajno smanjuju komunikacioni troškovi u odnosu na SAP u nekim scenarijima čak i do 200 puta manje. Ovaj algoritam međutim za razliku od SAP algoritma ne pronalazi najbližeg izvršioca u svim scenarijima i optimalnost odabira izvršioca nije 100% (osim

za male mreže ili velike vrednosti  $k$ ), već se kreće u granicama od 20-70%. Radi poboljšanja procenta izbora najbližeg izvršioca uvodi se proširenje  $k$ -SAP algoritma na bazi tzv. pohlepnog (eng. *greedy*) algoritma ( $k$ -SAPG) koje dovodi do poboljšanja optimalnosti izbora izvršioca (47-98%) uz malo dodatno komunikaciono opterećenje (prosečno 0.4 dodatne poruke po izvršiocu).

Drugi deo doprinosa u ovoj disertaciji bazira na uvođenju pet novih aukcijskih agregacionih algoritama. Tehnika agregacije se odavno koristi u bežičnim senzorskim mrežama kao tehnika za smanjenje količine prenešenih podataka objedinjavanjem senzorskih očitavanja pre slanja. U ovoj disertaciji uvodimo novi način agregacije prilikom licitiranja sa ciljem smanjenja komunikacionog opterećenje tokom licitiranja. Ovo je primenjeno kod prostog aukcijskog agregacionog algoritma (SAAP) uz uvođenje koncepta roditelj-dete gde svaki izvršilac u mreži ima samo po jednog roditelja i gde u toku licitacije izvršioci umesto da odgovaraju na aukcijski poziv zasebnim komunikacionim putanjama oni lokalno vrše agregaciju licitacija tako što sva deca šalju licitacije svojim roditeljima i onda roditelj vraća samo najbolju licitaciju putem jedne poruke. Na ovaj način je odlučivanje distribuirano po celoj mreži, a komunikaciono opterećenje smanjeno u odnosu na SAP u nekim slučajevima i do šest puta. Dalje smanjenje komunikacionih troškova daje lokalizovana verzija SAAP algoritma ( $k$ -SAAP). Za ovaj algoritam može se pokazati da je broj poruka po izvršiocu uvek veći ili jednak od  $2k/N$  (uz  $k < N$ ,  $N$  – broj izvršilaca) i manji ili jednak dva. Radi poboljšanja optimalnosti izbora izvršioca istraživano je i proširenje pohlepnim algoritmom ( $k$ -SAAPG).

Poslednja dva algoritma, aukcijski agregacioni algoritam (AAP) i lokalizovana verzija AAP algoritma ( $k$ -AAP), baziraju na konceptu topološkog predznanja o mreži koje se dobija u toku konfigurisanja mreže i koje omogućava da se lokalno odlučuje da se u informacionoj fazi poruke šalju dalje u mrežu samo ukoliko u se zna da u blizini ima bolji izvršilac. U suprotnom se poruke dalje ne šalju. Time se smanjuje komunikaciono opterećenje, a poboljšava se i optimalnost izbora izvršioca. Za mnogo manje  $k$  može se dobiti 100% optimalnost izbora izvršioca nego u slučaju ostalih lokalizovanih aukcijskih agregacionih algoritama.

Treći deo doprinosa ove disertacije je u uvođenju lokalizovanog prostog aukcijskog algoritma ( $k$ -SAAP) kao poboljšanja postojećeg algoritma za otkrivanje

lokacije izvršioca baziranog na informacionoj mreži (*iMesh* algoritam). Pokazuje se da se polaznih 95% efikasnosti pronalaženja najbližeg izvršioca, koliko osnovni *iMesh* algoritam ima u proseku, uvođenjem *k*-SAAP može povećati na 99.3% u proseku.

**Ključne reči:** aukcije, aukcijski agregacioni algoritmi, izbor izvršioca, otkrivanje servisa, otkrivanje lokacije izvršioca, bežične senzorske i aktuatorске mreže

# 1 Uvod

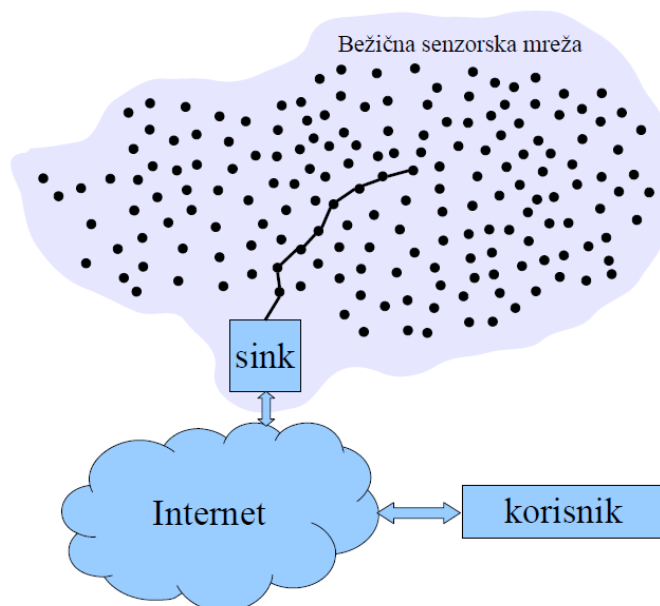
U uvodnom poglavlju dat je generalni prikaz oblasti koje su od interesa za ovu disertaciju. To su bežične senzorske mreže, multirobotski sistemi i bežične senzorske i aktuatorске mreže. Nakon toga je definisan problem kojim se bavi ova disertacija; izbor izvršioca i dodela posla u bežičnim mrežama elektronskih senzora i aktuatora. Nakon kratkog pregleda postojećih rešenja, pojma aukcija i mogućnosti njihove primene dat je kratak pregled novih rešenja ili poboljšanja postojećih. Mrežni model i parametri simulacija su takođe navedeni u jednom odeljku. Spisak doprinosa i pregled organizacije disertacije zaokružuju uvodno poglavlje.

## 1.1 Bežične senzorske mreže

Bežične senzorske mreže (BSM) su mreže koje se sastoje od mnoštva senzorskih čvorova koji kooperativno mere neke veličine na određenom prostoru i koji mogu međusobno da komuniciraju bežičnim putem. Ova oblast je veoma dobro istražena i objavljeno je mnogo preglednih članaka, istraživačkih članaka i knjiga (npr. [ASSC02], [S05], [SMZ07], [YMG08] [DP10]). Razlika između senzora i senzorskih čvorova je u tome što senzor ima sposobnost samo da meri neku veličinu, a senzorski čvor pored merenja te veličine ima i sposobnost obrade signala, memorisanja i bežične komunikacije. Dakle pored merenja neke veličine, senzorski čvorovi mogu tu vrednost da obrade, sačuvaju ili pošalju putem bežične komunikacije. Merene veličine mogu biti veoma različite, od jednostavnih ka kompleksnim koje kombinuju različite tehnike i tehnologije merenja (npr. optičke, magnetne, zvučne itd.). U ovoj disertaciji, ukoliko se kaže senzor, podrazumeva se senzorski čvor.

Kooperativno merenje određenih veličina na nekom prostoru obično se radi da bi se te informacije prosledile korisniku BSM. Na Slici 1.1.1 prikazana je tipična BSM pomoću koje se određeni podaci od interesa šalju od senzorskih čvorova (mali crni

krugovi) do sink čvora i potom putem interneta ka korisniku. Podaci koji se šalju mogu da se dobijaju po potrebi nakon zahteva korisnika za njima, periodično ili po pojavi nekog događaja od interesa. To su tri osnovna načina dobijanja podataka u BSM.

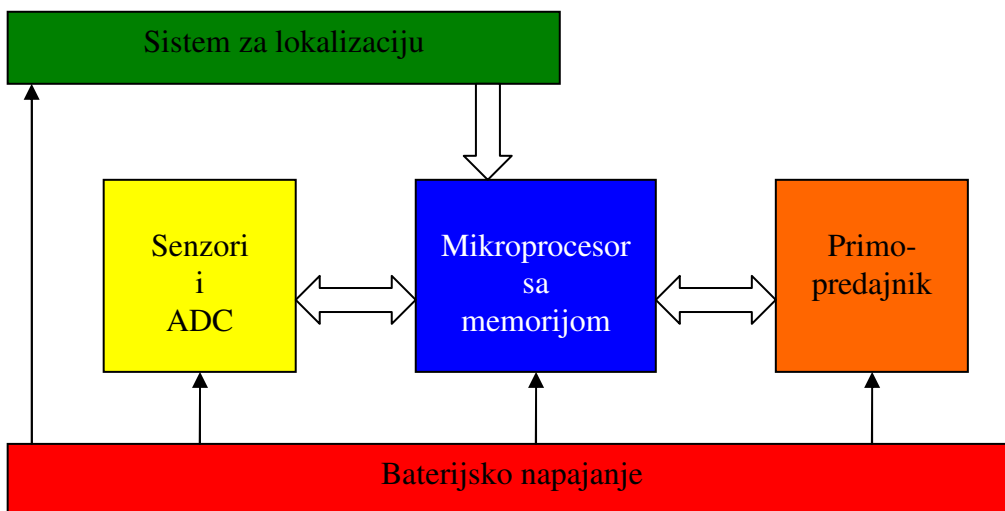


Slika 1.1.1 Tipična bežična senzorska mreža

Svaki čvor sastoji se od najmanje četiri dela. To su: napajanje, senzori sa analogno–digitalnim konvertorom, primopredajnik i mikroprocesor sa memorijom. Pored ova četiri dela postoje senzorski čvorovi koji sadrže i neki od sledeća tri dela [AV10]. To su: sistem za kretanje (ukoliko je čvor pokretan), sistem za lokalizaciju i generator energije. Kada je reč o senzorskim čvorovima u okviru ove disertacije, podrazumeva se da imaju strukturu koja je se sastoji od baterijskog napajanja, senzorskog dela, procesora sa memorijom, primopredajnika sa određenim komunikacionim radijusom i sistema za lokalizaciju koji omogućava geografsko lociranje čvora. Ovakva arhitektura prikazana je na Slici 1.1.2.

Baterijsko napajanje senzorskih čvorova, koje ima konačnu energiju, daje glavni istraživački motiv čiji je cilj da se postigne visoka energetska efikasnost BSM. Kada se pogledaju elementi strukture svakog senzorskog čvora sa Slike 1.1.2 kao i merenja koja su obavljena za razne komercijalne senzorske čvorove može se reći da najveću potrošnju energije ima primopredajni blok [AV10]. Ovaj blok omogućava bežičnu

komunikaciju pa je zato poželjno osmisliti takve komunikacione protokole i algoritme koji zahtevaju što manje komunikacije među čvorovima i samim tim najmanji utrošak energije. Na ovaj način se produžava radni vek mreže.



Slika 1.1.2 Arhitektura senzorskih čvorova

Primena BSM je u mnogim oblastima. BSM-e se koriste za vojne primene, za aplikacije praćenja parametara okruženja, u oblasti zdravstva (tzv. telesne mreže [Y06]), u oblasti kućne automatizacije i kontrole, za industrijske aplikacije [GH09], u zaštiti i praćenju stanja zgrada [JH10] i drugih zaštićenih objekata [WY+10], u savremenoj poljoprivredi i industriji hrane [RG+09], podvodnim istraživanjima [PKL07], savremenoj električnoj distributivnoj mreži [GBH10] itd. Lista različitih aplikacija u BSM data je u [SMZ07].

## 1.2 Multirobotski sistemi

Slično kao i oblasti BSM i oblast *multi-robotskih sistema* (MRS) je veoma dobro proučena i u literaturi postoji više preglednih članaka, taksonomija i poglavlja u monografijama i knjigama (npr. [DJMW96], [DJM02], [SW02], [FIN04], [P08]). Fokus većine radova koji su u vezi sa MRS je na koordinaciji i saradnji (kooperaciji). Krajnji cilj istraživanja u ovoj oblasti je povećanje efikasnosti u poređenju sa sistemima sa samo jednim robotom. Poboľšanja su moguća korišćenjem efikasnosti koja se javlja u



paralelizmu, tj. koordiniranom paralelnom izvođenju operacija. Termin *network robot system* (umreženi robotski sistem) je predložen od strane *Network Robot Forum*-a [N1], a izraz *networked robots* (umreženi roboti) je odobren od strane *IEEE* tehničkog komiteta za umreženu robotiku 2004 [N2]. Umrežena robotika je proširenje multi-robotskih sistema i prema preglednom članku o statusu robotike (Bekey and Yuh [BY08]), komunikacija između entiteta u MRS je fundamentalna za koordinaciju i saradnju i predstavlja najvažniji aspekt umreženih robota. Više o temi umreženih robota sa fokusom na probleme međupovezanosti robota, senzora i veštačke inteligencije dao je Kumar u poglavlju [KRS08]. Primena MRS i umreženih robota je sledeća. Roboti se mogu povezati i udružiti radi izvršenja zadataka koje ne može jedan robot da obavi ili bi bio potreban veći robot koji bi to mogao. Primeri takvih udruživanja su istraživanje terena, mapiranje, pretraga i izviđanje [BM+05] ili robotizovani fudbal koji je primer udruživanja u oblasti zabave, a koji je istraživački veoma atraktivan [KK+05].

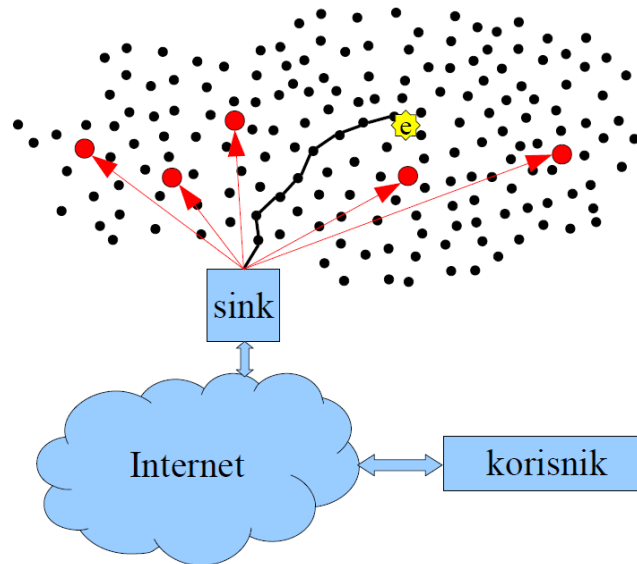
### **1.3 Multi-hop bežične senzorske i aktuatorске mreže**

Oblast bežičnih senzorskih i aktuatorskih mreža (BSAM) je nešto mlađa od oblasti BSM i MRS. Istraživanja su značajnije intenzivirana nakon što su Akyldiz i Kasimoglu objavili članak [AK04] u kojem su dali pregled istraživačkih izazova u ovoj oblasti.

U BSAM pored elektronskih senzora postoje i aktuatori kao izvršni elementi. Oni su povezani bežičnim medijumom radi distribuiranog prikupljanja podataka o okruženju, procesiranja prikupljenih podataka, donošenja odluka i njihovog izvršenja [NS10], [VDMC08]. Postoje različite primene ovakvih distribuiranih mreža. Nadgledanje poljoprivrednih dobara, tzv. precizna poljoprivreda [HS+10], nadgledanje i delovanje na delove okruženja [WS+09], praćenje parametara u 'pametnim' kućama [CCXS10], prevencija pojave požara [KCM12] i koordinacija više vozila [MB+10] su neki od primera primene BSAM mreža. Aktuatori mogu biti jednostavni uređaji (npr. prskalice, ventili, motori itd.) ili složeni sistemi kao što su roboti. U slučaju kada su aktuatori roboti, tada je reč o bežičnim senzorskim i robotskim mrežama (BSRM). Ova oblast je istraživački aktivna tek u poslednjih nekoliko godina [CO+11].

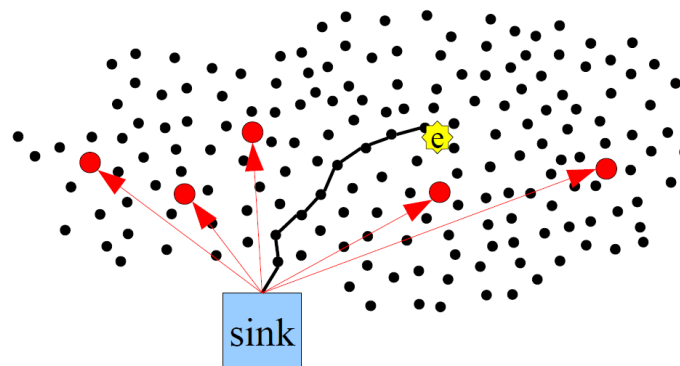
Arhitekture BSAM se dele na poluautomatizovane, automatizovane [AK04],

neautomatizovane i kooperativne [NS10]. Na Slici 1.3.1 imamo primer neautomatizovane arhitekture sa pet aktuatora (kružići crvene boje). Ova arhitektura je praktično potpuno centralizovana i sve zavisi od odluka koje donosi korisnik. Dakle, po pojavi događaja  $e$ , on se javlja korisniku putem senzorske mreže, sink čvora i interneta, a potom korisnik donosi odluku i šalje je nazad prema nekom od aktuatora.



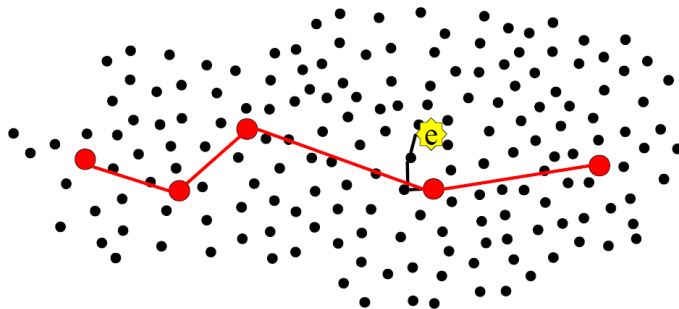
Slika 1.3.1 Ilustracija neautomatizovane BSAM-e

Na Slici 1.3.2 dat je prikaz poluautomatizovane BSAM-e gde se po pojavi događaja i javljanju sinku, odlučivanje prebacuje na sink čvor koji potom javlja odgovarajućem aktuatoru da reaguje na događaj. Javljanje aktuatorima na ovom primeru je direktno, a može biti realizovano i putem senzorske mreže. Kooperativne BSAM-e imaju istu arhitekturu s' tim što se ovde po potrebi konsultuje sink čvor pri odlučivanju, ali nije obavezno.



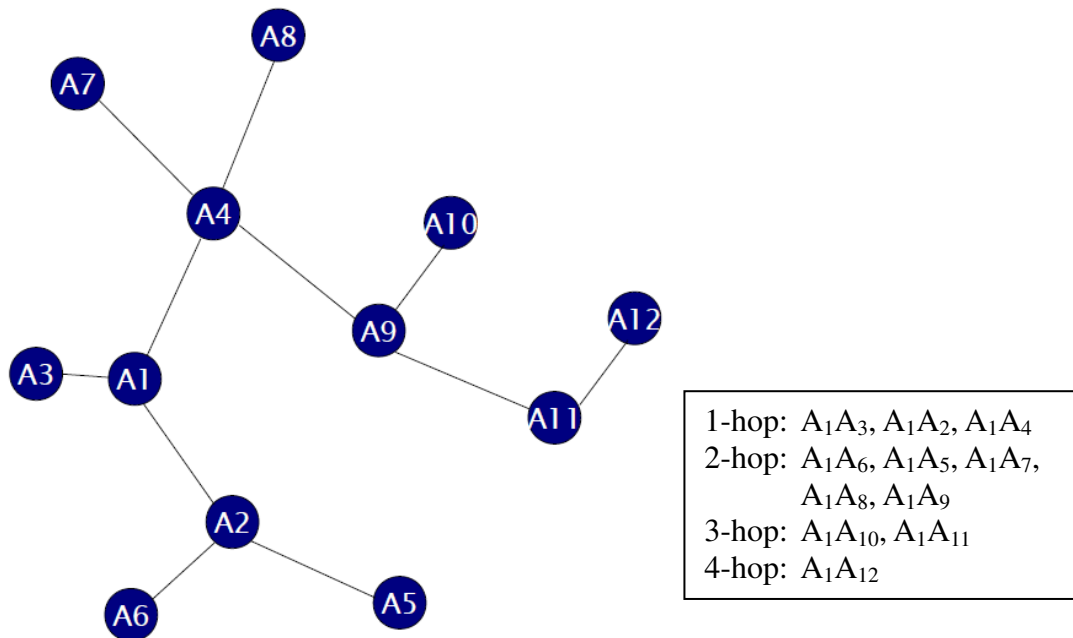
Slika 1.3.2 Ilustracija poluautomatizovane BSAM-e

Na Slici 1.3.3 prikazana je automatizovana arhitektura BSAM. Ovde nemamo sink čvor i odlučivanje u mreži je raspoređeno na aktuator, a i senzori učestvuju u procesu pronalaženja aktuatora. Aktuatori međusobno moraju biti u povezanoj mreži (kako je prikazano na slici) ili moraju biti povezani putem senzorske mreže.



Slika 1.3.3 Ilustracija automatizovane BSAM-e

U slučaju mreža koje imaju robote kao izvršioce, u većini radova se podrazumeva da oni mogu međusobno da komuniciraju direktno svaki sa svakim što podrazumeva da roboti formiraju kompletan graf mreže. Ovo je praktično veoma teško održiva pretpostavka, pogotovo ako se radi o scenarijima sa više robota i na većem prostoru.



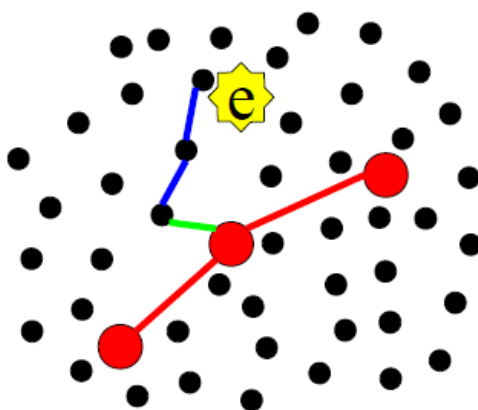
Slika 1.3.4 Ilustracija multi-hop komunikacije koja polazi od čvora A1

U ovoj disertaciji se ne podrazumeva kompletan graf izvršilaca već se podrazumeva da je graf izvršilaca povezan, dakle da se mreža ne sastoji iz više delova.

To znači i da je komunikacija između svaka dva izvršioca u mreži moguća slanjem komunikacionih poruka (čiji je ukupan broj konačan) preko ostalih izvršilaca koji samo prosleđuju ovu poruku ka određišanom izvršiocu. Ovo je ilustrovano na Slici 1.3.4 za scenario sa dvanaest izvršilaca (senzori su izostavljeni sa slike) na kojoj se posmatra multi-hop komunikacija između čvora A1 i ostalih čvorova u mreži. Za ovakav scenario, aktuator A1 može sa svakim drugim čvorom u mreži da komunicira u najviše četiri hopa (aktuator A12 je najdalji). Interesantno je uočiti da za slučaj posmatranja bilo koja dva čvora u mreži možemo odrediti maksimalno potrebni broj hopova koji za primer sa slike iznosi sedam.

Ovakav način komunikacije se naziva multi-hop komunikacija. Ono što je dovelo do potrebe za ovakvim načinom komunikacije je činjenica da je komunikacioni radijus svakog čvora, koji zavisi od konkretne realizacije primopredajnog dela čvora u BSAM, veličine od nekoliko metara pa do nekoliko desetina metara najčešće. Drugim rečima, komunikacioni radijus je konačan i za praktične primene nije moguće imati direktnu komunikaciju između svih čvorova mreže. Multi-hop komunikaciju koristimo i u ovoj disertaciji i zato je ovde reč o multi-hop bežičnim mrežama elektronskih senzora i aktuatora.

Za razliku od bežičnih senzorskih mreža gde imamo samo komunikaciju među sensorima, u BSAM imamo još dve paradigme, a to su komunikacija i koordinacija između senzora i aktuatora [W11] i među aktuatorima [AK04]. Ovo je ilustrovano na Slici 1.3.5.



Slika 1.3.5 Ilustracija komunikacionih paradigmi u BSAM

Komunikacija među sensorima je označena plavim linijama, komunikacija između senzora i aktuatora zelenim, a komunikacija među aktuatorima crvenim linijama. U ovoj disertaciji nas pre svega zanima komunikacija među aktuatorima kao izvršiocima, a komunikacija između senzora i aktuatora ima za cilj da prosledi informacije o događaju.

#### **1.4 Definicija problema izbora izvršioca i dodele posla**

U opštem slučaju problem izbora izvršioca i dodele posla može se definisati na sledeći način.

##### Definicija

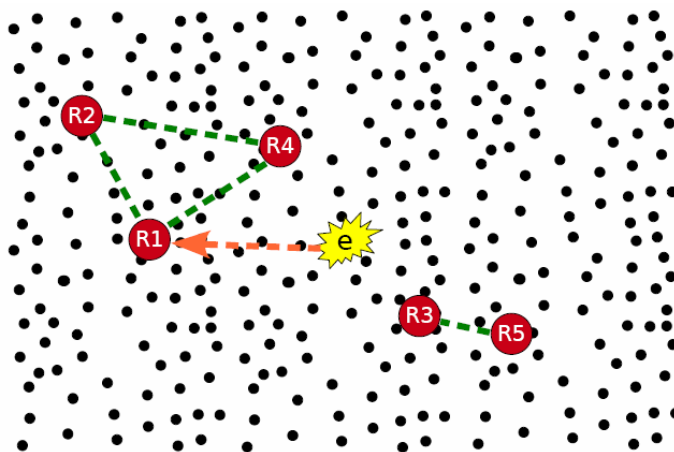
Za zadati skup senzora i njihovih karakteristika, izvršilaca (aktuatora ili robota) i njihovih karakteristika, događaja, raspoloživih resursa, troškova i karakteristika okruženja, problem koji treba rešiti je *pronaći najboljeg izvršioca (ili izvršioce) da reaguju na događaj i dodeliti mu (im) zadatak*.

Uzimajući u obzir da senzori, aktuatori i roboti imaju ograničene resurse, cilj je minimizovati troškove unutar postojećih energetske resursa. Ovaj problem se sastoji iz dva dela. Prvi deo predstavlja pronalaženje izvršioca polazeći od senzora koji je detektovao događaj. Ovaj problem se naziva otkrivanje servisa (*service discovery*) [LSS09] ili otkrivanje lokacije izvršioca (*location service*) [LNS10]. Uzimajući u obzir mobilnost izvršioca, pronaći jednog od izvršilaca nije jednostavan problem, zbog promena topologije mreže, posebno u slučaju kada komunikacioni troškovi nisu zanemarljivi. Nakon što je pronađen neki od izvršilaca, moguće je koristiti rutiranje na bazi pozicije [GS04]. Drugi deo problema predstavlja pronalaženje izvršioca koji je *najbolji* da reaguje na događaj.

Ono što je neophodno još definisati je šta znači *najbolji* izvršilac. Pretpostavljajući da svi izvršioci imaju sposobnost da obave zadatak i uzimajući u obzir osnovnu osobinu skoro svih bežičnih senzorskih i aktuatorskih mreža, a to je da imaju ograničenu energiju, možemo definisati da je najbolji izvršilac onaj koji obavi zadatak uz najmanji utrošak energije za najkraće vreme. Drugim rečima najbolji je onaj izvršilac koji ima minimalne troškove za obavljanje zadatka. U ovoj disertaciji, kao i u većini

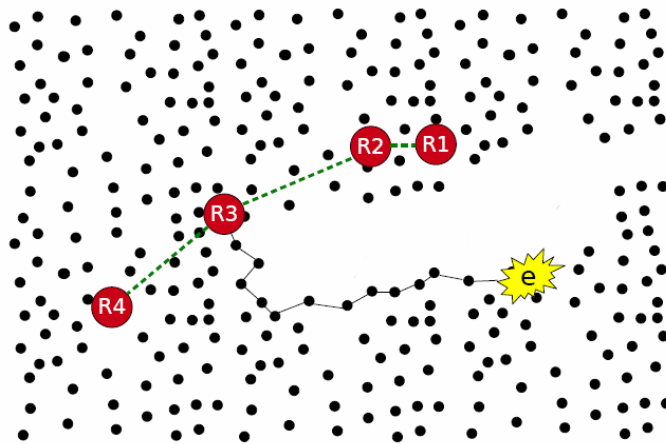
radova koji uzimaju u obzir i energetska efikasnost prilikom odabira izvršioca, je usvojen kriterijum da je najbolji onaj izvršilac koji je najbliži. Ovde treba napomenuti da je neophodno poznavanje lokacija događaja i izvršilaca da bi se moglo govoriti o blizini nekog izvršioca događaju i na osnovu toga procenjivati. Ovo predstavlja posebnu istraživačku oblast u BSM (lokalizacija), a u ovoj disertaciji se pretpostavlja da su lokacije poznate (npr. na bazi dodatnog uređaja za lokalizaciju – GPS ili nekog od algoritama za lokalizaciju). Alternativno, postoje i drugačiji kriterijumi, a to su, na primer, da izvršilac ima najviše energije, da ima najviše preostale energije nakon obavljenog zadatka, da se odabere izvršilac tako da ukupna energija izvršilaca u mreži bude ravnomerno raspoređena itd. Naravno mogući su i multikriterijalni pristupi gde se, na primer, traži optimalan odabir za minimalno rastojanje i maksimalnu energiju itd. Ovakvi kriterijumi mogu značajno da povećaju složenost postupka odlučivanja.

Razlozi zašto odmah nije pronađen najbolji izvršilac mogu biti različiti. Pre svega razlog može biti neoptimalan algoritam za pronalaženje izvršioca koji ne garantuje da će pronaći najboljeg. Ako izuzmemo centralizovana rešenja, kod kojih je moguće uvek pronaći najboljeg izvršioca, među lokalizovanim to najčešće nije moguće i za sada jedini lokalizovani algoritam koji garantuje pronalaženje je ILSR algoritam i čiji generalni opis će biti objavljen u [LYNS12].



Slika 1.4.1 Primer nepovezane mreže izvršilaca (robota)

Zatim razlozi mogu biti topološke prirode. Na Slici 1.4.1. je prikazana situacija gde je mreža izvršilaca nepovezana. Algoritam za pronalaženje izvršioca nalazi robota R1 koji konsultuje R2 i R4 i odlučuje da je R4 najbliži, a stvarno najbliži je R3, ali se do njega ne može doći jer je mreža nepovezana.



Slika 1.4.2 Senzorska mreža sa tzv. rupom otežava pronalaženje izvršioca

Na Slici 1.4.2 imamo tzv. rupu u senzorskoj mreži zbog koje pronađeni izvršilac (u primeru na slici to je robot R3) može biti dosta udaljen. Međutim konsultovanjem ostalih izvršilaca on može da odluči da je robot R1 najbliži i samim tim najbolji.

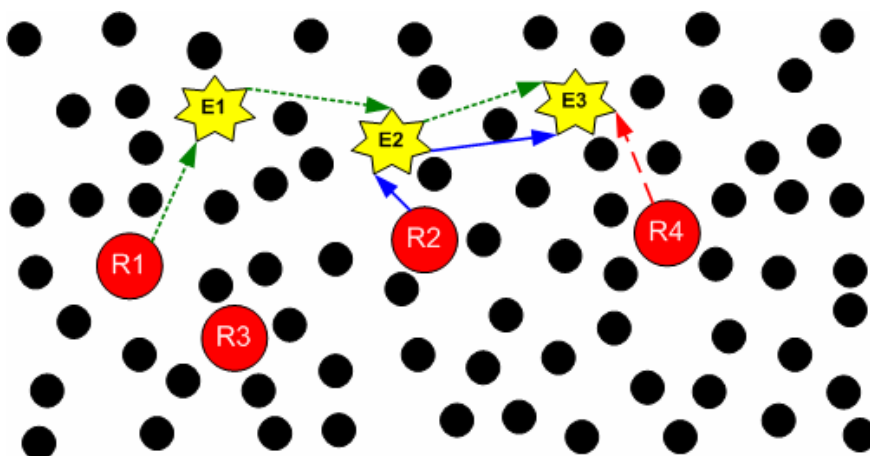
Događaji na koje se očekuje odgovarajuća reakcija, mogu biti različiti. Pretpostavka je da aktuatori i roboti uvek mogu da izvršavaju samo po jedan zadatak u jednom trenutku. Događaji mogu biti jednostruki ili višestruki. Izvršilac može da obavi samo jedan zadatak ili više zadataka u nizu. U scenariju gašenja požara, senzori detektuju događaj pojave požara (npr. merenjem temperature i količine dima), dojavljuju informacije o događaju nekom od izvršilaca i tada je potrebno doneti odluku ko će da reaguje i da ugasi požar. Ovo je primer jednostrukog događaja i jednostruke dodele zadatka izvršiocu. Ponekad je potrebno više izvršilaca da bi se obavio zadatak (npr. veći požar, ili pomeranje teškog objekta). To su primeri jednostrukih događaja sa višestrukom dodelom zadataka izvršiocima. U slučaju višestrukih događaja, veoma je važno kako se događaji pojavljuju i da li postoji vremensko ograničenje za izvršenje zadatka. Ukoliko nema ograničenja za izvršenje zadatka ili to nije problem, tada se svi višestruki događaji svode na iterativnu verziju jednostrukih događaja. Tako se može zaključiti da se višestruki događaji razlikuju od jednostrukih samo u urgentnim situacijama.

Višestruki događaji se mogu odvijati u potpunosti sekvencijalno, sekvencijalno po principu "grupa po grupa" ili paralelno. Najjednostavniji primer višestrukog događaja sa jednostrukim izvršiocom je gašenje požara gde jedan izvršilac treba da obiđe više lokacija i ugasi požar. Pitanje je naravno da li će to uspeti da uradi. U tim

situacijama je daleko bolji slučaj koji nazivamo višestruki događaj sa višestrukim dodelama zadataka kada se svakom lokalitetu dodeli po jedan izvršilac da gasi požar. U slučaju da ima više događaja nego izvršioca tada je potrebno nekima od njih dodeliti niz uzastopnih zadataka. Scenario sa tri paralelna događaja (npr. požara) je prikazan na Slici 1.4.3. Na ovoj slici su prikazane i tri različite mogućnosti dodele posla. Prva mogućnost je da izvršilac R1 obiđe sve tri lokacije jednu za drugom i samostalno obavi sva tri zadatka. Druga mogućnost je da se svakom događaju pridruži po jedan izvršilac, a treća mogućnost je kombinacija prethodne dve. Kako optimalno raspodeliti više poslova u slučaju ako imamo jednog aktuatora ili ako ih je više je razmatrano u [WT+11]. Problem gde treba simultano pomeriti više prepreka spada u istu grupu. Simultana postavka novih senzora na više lokacija takođe je jedan od primera. Tabela 1.4.1 prikazuje tipove događaja, zadataka, dodele posla i vrste rešenja.

Tabela 1.4.1 Vrste događaja, zadataka, dodele posla i rešenja

Događaji	Zadaci	Dodela posla	Rešenje
jednostruki		1 zadatak – 1 izvršilac	centralizovano
višestruki sekvencijalni	jedan izvršilac	višestruki zadaci – 1 izvršilac	lokalizovano
višestruki “grupa po grupa”	više izvršilaca	1 zadatak – višestruki izvršioci	distribuirano
višestruki paralelni		višestruki zadaci–višestruki izvršioci	"tržišno"



Slika 1.4.3 Tri paralelna događaja i različite mogućnosti dodele posla

U većini radova se problem izbora izvršioca i dodele posla (eng. *task assignment* ili *task allocation*) formalizuje kao neka od varijanti celobrojnog linearnog



programiranja (npr. [XZQD06], [SDM06], [MPGA07], [AB06]). U ovim najčešće centralizovanim formulacijama cilj optimizacije je minimizacija utroška energije, minimizacija ukupno pređenog puta, maksimizacija preostale energije, maksimizacija efekata i sl. Mobilnost često nije razmatrana (izuzeci su npr. [MPA06], [MPA10] i [SDM06]) kao ni komunikacioni aspekti izbora izvršioca (izuzeci su npr. [SM07], [AB06]). Komunikacioni troškovi se najčešće ne uzimaju u obzir, pogotovo u oblasti robotike, jer se smatra da su oni najmanje za jedan ili dva reda veličine manji od potrošnje energije za kretanje. Pored toga česta je i pretpostavka da roboti mogu sve vreme direktno da komuniciraju međusobno. Smatramo da su ove pretpostavke pregrube i da pitanje komunikacija, a samim tim i izbora izvršioca nije jednostavno. Jedan od retkih radova koji se bavi izborom izvršioca u BSRM, a koji uzima u obzir i komunikacione troškove i mobilnost je [CV10].

U BSAM, praćenje parametara okruženja i akciju obavljaju grupa senzorskih i aktuatorskih čvorova. Senzori su tipično jeftini, malih dimenzija, sa ograničenom procesorskom snagom, komunikacionim mogućnostima i resursima uopšte. Aktuatori, a pogotovo roboti, opremljeni su napajanjem koje može da traje duže i većeg je kapaciteta, a komunikacioni dometi i procesorska snaga su veći u poređenju sa sensorima. Senzori su po pravilu statični, a aktuatori mogu biti statični ili pokretni. Tipično, broj senzora je mnogo veći od broja aktuatora.

Dve su jedinstvene karakteristike BSAM: potreba za radom u realnom vremenu i koordinacija. Zahtev za radom u realnom vremenu se obično svodi na potrebu da vreme od trenutka pojave nekog događaja, preko izbora izvršioca, do izvršenja ne traje duže od nekog unapred zadatog vremenskog kašnjenja. Za razliku od bežičnih senzorskih mreža, gde se podaci skupljaju i šalju prema jednom čvoru, i koordinacija se vrši pomoću sensor-senzor koordinacije, u BSAM imamo dva nova mrežna fenomena. To su, sensor-aktuator koordinacija i aktuator-aktuator koordinacija [AK04]. U slučaju BSRM imamo sensor-robot i robot-robot koordinaciju.

Koordinacija između senzora i aktuatora (ili robota) omogućava da se informacija o događaju od interesa prosledi od senzora do aktuatora ili robota. Obzirom da se podrazumeva da su roboti u najvećem broju slučajeva mobilni, a i aktuatori, onda unapred nije poznata njihova geografska pozicija unutar prostora i potrebno je pronaći bar jednog ili više, a potom pronaći i najboljeg među njima da odgovori na događaj.

Pronalaženje aktuatora ili robota naziva se, kao što je već rečeno, problem otkrivanja servisa ili lokalizacija servisa (*service discovery* ili *location service*). Ovaj problem je izučavan u oblasti mobilnih *ad hoc* mreža već više od decenije [MBB09] i neka od rešenja mogu se direktno primeniti i u oblasti BSAM i BSRM. Odlučivanje o tome koji aktuatori ili roboti su najbolji za reagovanje na dati događaj se vrši u okviru koordinacije među aktuatorima ili robotima.

Na osnovu svega prethodno navedenog možemo dati definiciju problema izbora izvršioca i dodele posla koji se izučava u okviru ove disertacije.

### Definicija

Za zadati skup senzora i aktuatora sa njihovim unapred poznatim lokacijama i karakteristikama, poznatu površinu okruženja, datim jednim događajem i poznatom njegovom lokacijom, problem koji treba rešiti je *pronaći algoritam za izbor najbližeg izvršioca da reaguje na događaj i dodelu zadatka*.

## **1.5 Postojeća rešenja**

Dodela posla i raspodela posla unutar koordinacije među izvršiocima jedni su od najvažnijih problema koje treba rešiti. Potrebno je odlučiti koji izvršilac treba da izvrši koji zadatak. Postoje *centralizovana*, *distribuirana*, *lokalizovana* i rešenja na bazi primene *tržišnih* metoda. Na rešenje ovih problema imaju uticaj sledeći faktori. Energija izvršioca može biti neograničena (npr. solarno napajanje) ili ograničena. U slučaju rada u realnom vremenu, vreme izvršenja može biti značajan faktor. Kako vreme potrebno da se započne izvršenje (responsivnost), tako i vreme potrebno da se zadatak izvrši. Na bazi ovih faktora moguće je odrediti koji algoritam treba koristiti. Na primer, ukoliko je potreban brz odziv vrlo je verovatno da centralizovano rešenje neće biti dobro zbog toga što se prilikom odlučivanja konsultuje cela mreža i tada je neka vrsta distribuiranog rešenja neophodna.

Centralizovana rešenja podrazumevaju da se iz skupa izvršilaca bira jedan od njih da bude glavni, dobija sve potrebne podatke i vrši odluke o dodeli posla. Na ovaj način, teoretski, moguće je uvek naći optimalno rešenje jer glavni izvršilac ima sve potrebne podatke za odlučivanje. Međutim, ovo može biti računski teško, posebno ako je veći

broj izvršilaca što vodi lošoj skalabilnosti i sporom odzivu. Ovakva rešenja imaju i nizak nivo otpornosti na otkaz. Pored toga, odlikuju se i visokima komunikacionim troškovima jer glavni izvršilac mora da dobije *sve* relevantne informacije iz cele mreže.

Distribuirana rešenja koriste raspodelu odgovornosti i odlučivanja među izvršiocima. Pri odlučivanju korišćenjem lokalizovanih rešenja, izvršioci koriste samo lokalno dostupne informacije. Osnovne karakteristike lokalizovanih rešenja su: brz odziv, malo računsko opterećenje, mali komunikacioni troškovi, robustnost, dobra skalabilnost i otpornost na greške. Problemi koji se mogu javiti su posledica činjenice da se odluke donose na bazi lokalnih informacija, a nisu svi problemi u dodeli zadataka jednostavni za dekompoziciju. Ovo može dovesti do veoma sub-optimalnih rešenja. Koliko je nama poznato, za sada jedini lokalizovani algoritam koji obezbeđuje osobinu garantovane isporuke komunikacione poruke od senzora ka nekom pokretnom čvoru (izvršiocu ili sink čvoru) je ILSR algoritam koji će biti uskoro publikovan u [LYNS12].

Rešenja na bazi tržišnih metoda predstavljaju varijantu distribuiranog rešenja (nekad se naziva i polu-distribuirano) i to nije lokalizovano rešenje. Lokalizovano podrazumeva ograničenu količinu komunikacija i to samo među komšijama. Ovaj tip rešenja bazira na ekonomskom modelu tržišta. Izvršioci rade kao slobodni agenti da bi maksimizovali individualni profit. Oni se dogovaraju i licitiraju za dodelu zadataka koristeći se *aukcijama* koje će biti objašnjene u narednom odeljku. Aukcije su glavno sredstvo za odlučivanje. Zbog analogije sa realnom ekonomijom, izvršioci nekad saraduju, a nekad se takmiče. Cilj je postići globalnu efikasnost obavljanjem postavljenih zadataka na željeni način uz minimizaciju svih troškova. Glavni doprinos tržišnog pristupa je u tome što je po svojoj prirodi polu-distribuiran i što nastoji da usvoji pozitivne osobine centralizovanih rešenja uz minimizaciju negativnih.

## 1.6 Aukcije

Kada se spomene reč aukcije još uvek se obično misli na aukcije na kojima se prodaju umetnička dela po visokim cenama (npr. u *Sotheby*<sup>1</sup>), gde jedan čovek vodi aukciju, potencijalni kupci licitiraju posebnim znacima (npr. namigivanjem, kašljanjem i sl.) i gde pobjednika proglašavaju uz udarac drvenim čekićem. Pored ovih klasičnih

---

<sup>1</sup> Jedna od najpoznatijih svetskih aukcijskih kuća

aukcija [MM87], postoji mnogo elektronskih servisa na Internetu koji funkcionišu po principima aukcija i gde se trguje raznom robom (npr. eBay ili Limundo).

Danas su aukcije našle svoju primenu u mnogim oblastima koje nisu nužno u direktnoj vezi sa ekonomijom. Aukcije kao sredstvo odlučivanja u bežičnim senzorskim i aktuatorskim mrežama su predložili Gerkey i Mataric u [GM02a] i to je jedan od prvih radova koji uvode aukcije u ovu oblast.

Aukcije mogu biti:

- jednodimenzionalne ili multidimenzionalne, u zavisnosti da li je pri licitaciji bitna samo jedna veličina (npr. cena) ili i neke druge (npr. vreme isporuke)
- jednostrane ili dvostrane, u zavisnosti da li se na aukciji licitiraju samo kupci (mogu biti i samo prodavci) ili i jedni i drugi
- otvorene ili zatvorene, u zavisnosti od toga da li svi učesnici aukcije imaju informaciju o licitacijama
- po prvoj ceni, ukoliko se plaća izlicitirani iznos, ili po  $k$ -toj ceni ukoliko se plaća iznos koji je označen kao  $k$ -ti
- za pojedinačnu robu ili za više istih komada
- za pojedinačnu robu ili za više komada robe različite vrste (kombinatorijalna aukcija)

Ovu klasifikaciju i objašnjenje za istraživače iz oblasti računarskih nauka su dali Parsons *et al.* u [PRK11], a detaljan teorijski, matematički uvod u aukcije je dao Wolfstetter u [W96].

Za potrebe algoritama u okviru ove disertacije korišćene su jednodimenzionalne (uvek je samo cena element licitacije), jednostrane (samo izvršioc licitiraju), zatvorene (izvršioc međusobno ne znaju licitacije ostalih), po prvoj ceni (posao dobija izvršilac koji da najbolju licitaciju) i aukcije su za pojedinačnu robu (samo jedan događaj se posmatra).

### **1.7 Nova rešenja ili poboljšanja postojećih**

U ovoj disertaciji je predloženo nekoliko unapređenja postojećih algoritama i predloženo je nekoliko novih algoritama. Polazeći od prostog aukcijskog algoritma koji je dat u [MPGA05] u oblasti BSAM i sličnog algoritma pod nazivom *Murdoch* u oblasti

robotike [GM02], gde su izvršioci odgovarali organizatoru aukcije licitiranjem direktnom komunikacijom (jedan hop), urađeno je njegovo poboljšanje u vidu lokalizovanog multi-hop aukcijskog algoritma pod nazivom  $k$ -SAP. Ovaj algoritam je dalje proširen i poboljšan uvođenjem tzv. pohlepnog (eng. *greedy*) algoritma i takav algoritam je označen sa  $k$ -SAPG. Ovaj algoritam poboljšava efikasnost pronalaženja najbližeg izvršioca po ceni malog dodatnog komunikacionog opterećenja.

Dalje je predloženo nekoliko novih aukcijskih algoritama koji uključuju tehniku agregacije. Prvi od njih je prosti aukcijski agregacioni algoritam – SAAP. Ovaj algoritam smanjuje komunikacione troškove centralizovanog multi-hop prostog aukcijskog algoritma (SAP) na taj način što smanjuje komunikacione troškove u fazi licitacije. Ovo se postiže uvođenjem koncepta roditelj-dete u mrežnom stablu gde se potom vrši agregacija licitiranih vrednosti kod roditelja. Roditelji vrše lokalnu aukciju među svi izvršiocima koji su im deca. Dakle, umesto da svi izvršioci licitiraju posebnim putanjama, kao kod SAP algoritma, ovde samo roditelji odgovaraju nazad prema organizatoru aukcije. Lokalizovana verzija prostog aukcijskog agregacionog algoritma (u oznaci  $k$ -SAAP) dalje omogućava smanjenje komunikacionih troškova u slučajevima kada nije neophodno uvek pronaći najbližeg izvršioca. Kod ovog algoritma je takođe dodatno poboljšanje urađeno korišćenjem tzv. pohlepnog (eng. *greedy*) algoritma (u oznaci  $k$ -SAAPG).

Najkompleksniji od algoritama je aukcijski agregacioni algoritam (AAP) koji ima praktičnog smisla samo u lokalizovanoj varijanti  $k$ -AAP. Osnovna ideja je da u toku konfigurisanja mreže svaki izvršilac dobije informacije o poziciji susednih izvršilaca koji su  $k$ -hopova udaljeni. Tada u informacionoj fazi svaki izvršilac može lokalno da odluči da li da prosleđuje informaciju o događaju dalje ili ne, zavisno od toga da li postoji neki od izvršilaca na  $k$ -hop udaljenosti koji ima bolju cenu.

U ovoj disertaciji je istražena i mogućnost primene aukcija za poboljšanje jednog od postojećih algoritama za pronalaženje izvršioca na bazi informacione mreže (*iMesh* algoritam [LSS09]). Iako ovaj algoritam ima visoku efikasnost pronalaženja najbližeg izvršioca (oko 95% u proseku), pokazuje se da se primenom lokalizovanog prostog aukcijskog algoritma ( $k$ -SAAP) ta vrednost može još povećati, uz malo povećanje komunikacionih troškova, u nekim scenarijima čak i do 100%.

## 1.8 Mrežni model i parametri simulacija

U ovom odeljku su date zajedničke karakteristike svih modela i parametara koji su korišćeni za potrebe simulacije algoritama za izbor izvršioca. Ukoliko nije drugačije navedeno podrazumeva se sledeći mrežni model. Mreža se sastoji od skupa sa  $m$  senzora  $S = \{S_1, \dots, S_m\}$  i skupa sa  $n$  aktuatora  $A = \{A_1, \dots, A_n\}$ . Pri tome je broj  $n$  znatno manji od broja  $m$ , barem za red veličine. Sensori imaju komunikacioni radijus  $r_S$ , a aktuatori komunikacioni radijus  $r_A$ . Pri tome važi da je komunikacioni radijus senzora manji ili jednak komunikacionom radijusu aktuatora, tj.  $r_S \leq r_A$ . Komunikacija između dva senzora je moguća ukoliko su oba na rastojanju ne većem od radijusa  $r_S$ , komunikacija između dva aktuatora je moguća ukoliko su na rastojanju ne većem od radijusa  $r_A$ , a komunikacija između senzora i aktuatora je moguća ukoliko je senzor udaljen od aktuatora za ne više od  $r_S$ .

Mrežni graf je definisan kao  $G = (V, E)$ , gde je  $V = A$  skup čvorova grafa koji uključuju samo aktuatore jer nas u ovoj disertaciji interesuje komunikacija među izvršiocima, i postoji grana grafa  $e$  između čvorova  $u$  i  $v$ ,  $e = (uv) \in E$ , koja je element skupa grana grafa  $E$ , ako i samo ako je Euklidsko rastojanje među njima manje ili jednako komunikacionom radijusu, tj.  $|uv| \leq r_A$ .

U simulacijama se za formiranje povezane mreže izvršilaca (aktuatora ili robota) koristi MIN-DPA algoritam [OSY08], ukoliko nije drugačije naglašeno. Ovaj algoritam generiše mrežu po UDG (*Unit Disc Graph*) modelu na pseudo slučajan način i raspoređuje čvorove u povezanu mrežu što je moguće uniformnije. U dodatku D1 u Tabeli D.1 dato je osam konkretnih primera generisanih mreža korišćenjem MIN-DPA algoritma za različite veličine i gustine mreže. Prostor u kojem se nalazi bežična mreža elektronskih senzora i izvršilaca (aktuatora ili robota) je dimenzija  $100 \times 100m$ . Broj izvršilaca  $N$  bira se iz diskretnog skupa  $I = \{10, 20, 50, 100\}$ . Prilikom formiranja grafa mreže izvršilaca, prosečan broj susednih čvorova  $d$  menja se u opsegu od 4 do 10. Za deset različitih grafova ponavljano je po sto eksperimenata gde je po jedan događaj na koji se očekuje odziv ubačen u mrežu na pseudo slučajan način. Izvršioci učestvuju na aukcijama i licitiraju sa cenom koja zavisi isključivo od udaljenosti izvršioca od događaja. Putanje slanja odgovora kolektoru (izvršiocu koji je inicirao aukciju) računane su u simulacijama po algoritmu najkraće putanje tj. putanje sa najmanjim brojem

hopova.

Skraćena verzija simulacionog algoritma je:

---

---

```
ponovi 10 puta za svaki odabrani algoritam {
    generiši povezanu pseudo slučajnu mrežu izvršioca koristeći
    MIN-DPA algoritam ili na slučajan način
    ponovi 100 puta {
        ubaci jedan slučajan događaj
        odaberi kolektora probabilističkom ili cross lookup metodom
        pokreni aukciju i šalji informaciju o događaju
        prihvati sve licitacije i odredi najboljeg izvršioca
        proveri da li je izabrani izvršilac najbliži događaju
        sračunaj sve komunikacione troškove
        sačuvaj u fajlovima sve relevantne podatke
    }
    izvrši potrebna računanja za potrebe crtanja grafova
}
```

---

---

Kompletni globalni simulacioni algoritmi su dati u dodatku D2 za SAP, *k*-SAP, *k*-SAPG, SAAP, *k*-SAAP, *k*-SAAPG i *k*-AAP, a u dodatku D3 za nadgradnju *iMesh* algoritma upotrebom *k*-SAP i *k*-SAAP aukcijskih algoritama. Simulacije su rađene u programskom jeziku C i ne zavise od platforme na kojoj se radi. Kompajliranje je rađeno uz pomoć GCC kompajlera u Cygwin okruženju na računarima na kojima je Windows okruženje, a rađeno je i na računarima koji rade sa Ubuntu Linux okruženjem. Korišćen je rad u konzoli i Eclipse razvojno okruženje. Izvorni fajlovi imaju ukupno oko 1750 linija koda u nekoliko fajlova. Trajanje simulacija na računarima klase Pentium 4, zavisno od parametara, može da se kreće od nekoliko minuta, preko nekoliko sati do maksimum desetak sati za simulacije *iMesh* algoritma i njegovog poboljšanja uvođenjem aukcija.

## 1.9 Doprinosi disertacije

U okviru ove disertacije objavljeno je jedno poglavlje u istaknutoj monografiji od međunarodnog značaja (kategorija M13):

H. Liu, V. Malbasa, **I. Mezei**, A. Nayak, I. Stojmenovic, "Coordination in Sensor, Actuator and Robot Networks", In: *Wireless Sensor and Actuator Networks: Algorithms and Protocols for Scalable Coordination and Data Communication* (Ed.: A. Nayak and I. Stojmenovic), Wiley Blackwell, Jan. 2010, ISBN 978-0-470-17082-3, pp. 233-262

Četiri rada su publikovana u međunarodnim časopisima sa SCI liste, od čega jedan M21 kategorije, dva M23 kategorije, a jedan je objavljen u časopisu koji je nekada bio na SCI listi ali više nije pa se prema preporuci Ministarstva za obrazovanje i nauku kategoriše kao vodeći nacionalni časopis M51 kategorije.

**I. Mezei**, V. Malbasa, I. Stojmenovic, "Robot to Robot: Communication Aspects of Coordination in Robot Wireless Networks", *IEEE Robotics and Automation Magazine*, Vol. 17, No. 4, decembar 2010, pp. 63-69, ISSN 1070-9932, M21

**I. Mezei**, V. Malbasa, I. Stojmenovic, "Greedy Extension of Localized Auction Based Protocols for Wireless Actuator Task Assignment", prihvaćeno za publikovanje u *Ad Hoc and Sensor Wireless Networks*, ISSN 1551-9899, M23

**I. Mezei**, M.Lukic, V. Malbasa, I. Stojmenovic, "Auctions and iMesh Based Task Assignment in Wireless Sensor and Actuator Networks", na drugom krugu recenzije u časopisu *Computer Communications*, Elsevier, ISSN 0140-3664, M23

**I. Mezei**, V. Malbasa, I. Stojmenovic, "Auction Aggregation Protocols for Wireless Robot-Robot Coordination", P.M. Ruiz and J.J. Garcia-Luna-Aceves (Eds.): ADHOC-NOW 2009, *Lecture Notes on Computer Science 5793*, Springer-Verlag Berlin Heidelberg 2009, ISSN 0302-9743, pp. 180 - 193, M51

Na međunarodnim konferencijama objavljeno je šest članaka, a na nacionalnim jedan članak koji predstavljaju deo rezultata rada na ovoj disertaciji.

M. Lukic, N. Janicijevic, **I. Mezei**, "Improved Decision Making in WSN Based on Localized Auctions and Fuzzy Logic", 19th Telecommunications Forum (Telfor), pp.



238-241, 2011, M33

B. Gasparovic, **I. Mezei**, “Auction Aggregation Protocols for Agent-based Task Assignment in Multi-hop Wireless Sensor and Robot Networks”, Proc. on IEEE/ASME International Conference on Advanced Intelligent Mechatronics AIM2011, 3-7 July 2011., Budapest, Hungary, pp. 247-252, M33

**I. Mezei**, N. Janicijevic, “Decision Making Based on Localized Auctions in Wireless Sensor Networks”, Proc. on the IEEE Eurocon2011 International Conference on a Computer as a Tool, 27-29.04.2011. Lisbon, Portugal, pp. 1-4, M33

N. Janicijevic, **I. Mezei**, “Decision making by localized auctions and data fusion in building energy monitoring applications”, Proc. on X Int. Conf. on Systems, Automatic Control and Measurements, SAUM 2010, Nis, Serbia, 10-12 Nov. 2010, pp. 246-249, M33

B. Gašparović, **I. Mezei**, “Aukcijski Agregacioni Protokoli Bazirani na Agentima u Multi-hop Bežičnim Robotskim Mrežama”, (in Serbian) Infoteh 2010, Jahorina, Bosnia and Herzegovina, 17-19.03.2010, Vol. 9, Ref. B-II-14, pp. 237-240, M63

**I. Mezei**, V. Malbasa and I. Stojmenovic, “Greedy Extension of Localized Auction Based Protocols for Wireless Robot-Robot Coordination”, Proc. on 7th International symposium on intelligent systems and informatics (SISY 2009), ISBN 978-1-4244-5349-8, pp. 53-57, M33

M.Lukic, **I.Mezei**, „Distributed Distance Sensitive iMesh based Service Discovery in Dense WSN“, prihvaćeno za prezentovanje na konferenciji AdHocNow2012 u julu 2012, M33.

Na bazi ideja obrađenih u ovoj disertaciji, njihovim proširivanjem urađena su i sledeća dva diplomska-master rada.

N. Janićijević, “Primena lokalizovanih aukcija pri odlučivanju u bežičnim senzorskim mrežama”, diplomski-master rad, Fakultet tehničkih nauka, Novi Sad, 2010.

B. Gašparović, “Aukcijski agregacioni protokoli bazirani na agentima u multi-hop bežičnim robotskim mrežama”, diplomski-master rad, Fakultet tehničkih nauka, Novi Sad, 2010.

## 1.10 Pregled disertacije

Nakon uvodnih razmatranja, pozicioniranja oblasti problema, definisanja problema izbora izvršioca, generalnog pregleda postojećih rešenja, opisa predloženih rešenja u ovoj disertaciji kao i doprinosa disertacije, ostatak disertacije organizovan je na sledeći način. U poglavlju 2 dat je pregled literature podeljen u tri oblasti koje su od interesa za ovu disertaciju. To su problem izbora izvršioca u slučaju više izvršilaca, različiti metodi za izbor i otkrivanje lokacije izvršioca.

U poglavlju 3 i 4 dat je pregled mogućnosti za izbor izvršioca u zavisnosti od toga da li se podrazumevaju idealne komunikacije ili postoje komunikacioni troškovi. Prikazana su rešenja u zavisnosti od vrsta događaja.

Dva poboljšanja prostog aukcijskog algoritma, prikazana su u poglavlju 5. To su lokalizovani prosti aukcijski algoritam ( $k$ -SAP) i njegovo proširenje tzv. pohlepnim algoritmom ( $k$ -SAPG).

Pet novih aukcijskih agregacionih algoritama (SAAP,  $k$ -SAAP,  $k$ -SAAPG, AAP i  $k$ -AAP) prikazani su u poglavlju 6.

U poglavlju 7 je data primena lokalizovanog aukcijskog agregacionog algoritama za poboljšanje postojećeg algoritma za nalaženje izvršioca na bazi informacione mreže (*iMesh* algoritam).

Disertaciju zaokružuju zaključci, usmerenja za budući rad, spisak literaturnih referenci i dodaci.

## 2 Pregled literature

U okviru ovog poglavlja dat je pregled literature grupisan u tri podoblasti koje su od interesa u ovoj disertaciji. To su:

- Problem izbora izvršioca u slučaju više izvršilaca
- Različiti metodi za izbor izvršioca i
- Otkrivanje lokacije izvršioca

### 2.1 Problem izbora izvršioca u slučaju više izvršilaca

Gerkey i Mataric su dali formalnu analizu i taksonomiju [GM04] dodele posla u multi-robotskim sistemima (*multi-robot task allocation* (MRTA)). Taksonomiju su organizovali na bazi tri faktora: jednostruki/multi-robotski zadaci, roboti koji mogu da izvrše jednostruke/višestruke zadatke i trenutna/vremenski produžena dodela posla. Analizirali su sve kombinacije ovih faktora i prikazali moguća rešenja takvih problema. Njihov rad predstavlja ključan rad u domenu taksonomije izbora izvršioca i dodele posla, ali u njemu nije obuhvaćen komunikacioni aspekt.

Dudek *et al.* su dali opširnu taksonomiju [DJMW96] i poglavlje u knjizi [DJM02] koja obuhvata multi-agentnu robotiku i proučava problematiku veličine i kompleksnosti ovakvih sistema uvođenjem sedam taksonomskih osa. To su: veličina sistema (broj robota: 1, 2, više ili beskonačno), komunikacioni opseg (nema ga, blizak ili beskonačan), komunikaciona topologija (slanje svima, na bazi adresa, na bazi stabla ili grafa), komunikacioni propusni opseg (besplatan, troškovi jednaki troškovima kretanja, veoma visoka cena, bez komunikacija), rekonfigurabilnost (statička, koordinirana ili dinamička), računске mogućnosti (nelinearno sumiranje, konačni automati, ekvivalenti Turingovoj mašini) i osobine članova (identični, homogeni, različiti). Fokus njihovog rada je na generalnim aspektima komunikacija i računskih

mogućnosti.

Chevaleyre *et al.* su dali opširni pregled problema alokacije resursa u slučaju više agenata (*multi-agent resource allocation* (MARA)) [IMRA06]. Oni posmatraju problem izbora izvršioca kao specijalni slučaj MARA. U vezi našeg rada, oni razlikuju dve klase procedura za izbor izvršioca, centralizovane i distribuirane. Neki od protokola na bazi aukcija su navedeni kao primeri centralizovanih rešenja, a protokoli na bazi dogovaranja kao primer distribuiranih. U radu su naznačeni i generalni problemi vezani za komunikacije i računsku složenost.

Stone i Veloso su dali pregled multi-agentnih i multi-robotskih sistema [SW02]. Pregled je organizovan u dve dimenzije: stepen heterogenosti agenata i količina komunikacija među njima. Polazeći od najjednostavnijeg multi-agentnog scenarija (homogeni agenti koji ne komuniciraju), razmatran je potpuni spektar mogućih multi-agentnih sistema, zaključno sa visoko heterogenim agentima koji imaju mogućnost komunikacije. Scenario koji je korišćen kao ilustracija je igra Predator/Plen (*Predator/Prey*).

Farinelli *et al.* su dali klasifikaciju u multi-robotskim sistemima (MRS) čiji je fokus na koordinaciji u različitim domenima primene [FIN04]. Da bi klasifikovali MRS, predložili su dve grupe dimenzija: koordinacione i sistemske. Prema njima MRS su veoma dobro, slabo ili nisu uopšte koordinirani i organizovani su kao potpuno ili delimično centralizovani ili kao distribuirani. U sistemske dimenzije spadaju: komunikacije, kompozicija tima, arhitektura sistema i veličina tima.

Parker u poglavlju u knjizi [P08] daje opis mobilnih multi-robotskih sistema (MMRS). Arhitekture, komunikacije, heterogenost, izbor izvršioca, učenje i primena MMRS su prikazani. Problem izbora izvršioca u MMRS je grubo podeljen na metode na bazi ponašanja (*behavioral-based*) i na bazi tržišne ekonomije (*market-based*).

Mosteo i Monteno su prikazali pregled metoda za izbor izvršioca u multi-robotskim sistemima u tehničkom izveštaju [MM10]. Ovaj izveštaj predstavlja pregled osnovnih pristupa u rešavanju problema izbora izvršioca u kontekstu mobilnih robota, sa jakim akcentom na servisnoj robotici.

Selvaradjou *et al.* su formulisali problem izbora izvršioca u slučaju višestrukih aktuatora (*multi-actuator task allocation problem* (MTAP)) kao optimizacioni problem

u klasi mešovitog celobrojnog nelinearnog programiranja (*mixed integer non-linear programming* (MINLP)) ([SDM06] i [SM10]), koristeći scenarije sa mobilnim aktuatorima. Može se pokazati da je ovaj problem NP-kompletnan, jer je sličan poznatom NP-kompletnom problemu trgovačkog putnika. Optimizacioni cilj je minimizacija ukupno pređenog puta aktuatora uz ispunjenje zahteva za rad u realnom vremenu i ograničenja potrošnje resursa. Autori predlažu upotrebu nekoliko heuristika radi pronalaženja suboptimalnih rešenja. Ovakva centralizovana rešenja podrazumevaju da su sve informacije o celoj mreži poznate *a priori*. U sličnim centralizovanim formulacijama kao cilj optimizacije pojavljuje se u radovima i minimizacija utrošene energije, maksimizacija korisnosti ili maksimizacija rezidualne energije. Problem u većini ovakvih radova je što se komunikacioni troškovi ili ne uzimaju u obzir ili se podrazumeva kompletan graf koji formiraju izvršioci. Nedostatak studija koje uključuju i probleme vezane za komunikacije su istaknuli Ruiz-Ibarra i Villasenor-Gonzalez za različite oblike kooperacije među izvršiocima (aktuatorima) u bežičnim senzorskim i aktuatorskim mrežama [RV09]. Pregled prednosti i mana centralizovanih rešenja i rešenja na bazi tržišne ekonomije za MTAP su dali Al-Yafi i Lee [AL09], a razmatranje o centralizovanim i distribuiranim rešenjima u oblasti BSAM su dali Cao *et al.* u [CCXS10].

Melodia *et al.* su MTAP takođe formulisali kao MINLP u [MPA06]. Za razliku od ostalih njihovih članaka, ovde su uzeli u obzir i mobilnost aktuatora. Njihov pristup podrazumeva potpuno povezanu mrežu, a komunikacioni troškovi su modelovani kao prosto koordinaciono kašnjenje u MINLP uslovima. Pored toga nije rečeno kako rešiti MTAP, a samim tim kako izvršiti odabir izvršioca.

Yu i Prassana su koristili sličan pristup koristeći varijantu MINLP formulacije i nekoliko heuristika za rešavanje ovog problema u bežičnim senzorskim mrežama [YP05]. Međutim u obzir su uzete samo komunikacije među prvim komšijama (eng. *one hop*).

Da sumiramo, pronalaženje rešenja za izbor izvršioca u konkretnim scenarijima kada se obzir uzima mobilnost izvršioca, kada komunikacioni troškovi nisu zanemarljivi i kada je komunikacione protokole potrebno specificirati i dalje predstavlja istraživački izazov.

## 2.2 Različiti metodi za izbor izvršioca

Xu *et al.* su problem izbora jednog aktuatora formulisali kao varijantu celobrojnog linearnog programiranja (*integer linear programming*) u [XZQD06]. Za rešenje problema su predložene tri heuristike. Prve dve, *Minimum Path Length Cost Tree* i *Minimum Energy Cost Tree*, predstavljaju približne, centralizovane algoritme izbora sa višestrukim ograničenjima koji minimizuju kašnjenja i utrošenu energiju, respektivno. Treća heuristika, *Hop-Bounded and Minimum Energy Cost Tree Algorithm*, je distribuirani algoritam koji ima za cilj da pronađe aktuatora uz nekoliko ograničenja. To su udaljenje aktuatora mereno brojem komunikacionih hopova od događaja i mogućnost pronalaska putanje za koju je potreban minimalan utrošak energije da bi se informacije o događaju prenele od senzora do aktuatora. Dakle, ovo su rešenja koja vrše izbor najpogodnijeg izvršioca u oblasti događaja.

Komunikacioni aspekti izbora izvršioca se često zanemaruju ili kao što ih Atay i Bayazit u slučaju centralizovane MRTA [AB06] modeluju dodavanjem jednog činioca u optimizacionu funkciju koji reprezentuje broj parova robota koji mogu da komuniciraju i koje treba maksimizovati. Pri tome nije naveden komunikacioni protokol i upotrebljena je samo direktna komunikacija sa prvim komšijama.

Komunikaciono efikasan multi-robotski (*communication efficient multi-robot - CEMR*) algoritam za izbor izvršioca i planiranje izvršenja u heterogenim MRS dali su Shah i Meng u [SM07]. CEMR podrazumeva direktnu komunikaciju između robota i centralne jedinice, kao i direktnu komunikaciju među pojedinim parovima robota. Svaki robot obaveštava centralnu jedinicu o svom statusu kao i o detektovanim zadacima koje treba izvršiti. Centralna jedinica dodeljuje poslove koristeći se metodom na bazi aukcija sa fitnes funkcijom koja uključuje mogućnost izvršenja zadatka, udaljenost od zadatka i sposobnost da se zadatak izvrši. Roboti primaju informaciju o statusu (uključujući i pozicije) svih ostalih robota od centralne jedinice i mogu da traže pomoć direktno od bliskih robota. Mogućnost multi-hop komunikacije nije razmatrana u ovom radu. Simulacije su vršene sa 6 robota, a kao alternativno rešenje se predlaže slanje poruka

svim ostalim robotima, umesto samo nekima od njih, koji bi mogli da budu slobodni, a na bazi prethodnog znanja.

Sujit *et al.* su istraživali tri decentralizovana algoritma za izbor izvršioca u slučaju bespilotnih letilica (*unmanned aerial vehicles* – UAVs) u scenarijima 'uništavanja ciljeva' koristeći koncepte dogovaranja, teorije igara i teorije timova [SSG07]. Uočeno je da potpuna komunikacija među letilicama dovodi do velikih komunikacionih troškova. Ukazana je potreba za lokalizovanom komunikacijom (komunikacija među komšijama) kao i potreba za multi-hop komunikacijama ukoliko su agenti izvan komunikacionog opsega. Međutim, simuliran je samo slučaj kompletnog grafa između letilica. Multi-hop komunikacija je pomenuta u nekim od scenarija, ali nije dat prikaz ni jednog komunikacionog protokola koji je korišćen.

U nekim slučajevima problem izbora izvršioca moguće je posmatrati kao jedan od matematičkih problema dodeljivanja (*assignment problems*) (npr. Kaleci *et al.* [KPOK10]). Povodom pedeset godina postojanja ovih problema Pentico je dao njihov pregled u [P07], a pregled generalizovanih problema dodeljivanja (*generalized assignment problems*) su prikazali Cattryse i Wassenhove još u [CW92]. Gerkey i Mataric u [GM04] problem dodele jednog zadatka jednom robotu formulišu kao problem optimalnog dodeljivanja (*optimal assignment problem* – OAP), koji je rešavan koristećenjem optimizacione heuristike (*Hungarian algorithm* [K55]) i poređen sa distribuiranom verzijom na bazi aukcijskog algoritma koji je prvi predložio Bertsekas [B90]. Simulacioni rezultati su dati za jednak broj robota i zadataka. Kwok *et al.* analiziraju OAP u [KDPT02] za slučajeve više ciljeva i više agenata. Farinelli *et al.* pokazuju da se u nekim slučajevima OAP može formulisati kao generalizovani problem dodeljivanja (*generalized assignment problem* - GAP) [FINZ06]. To je centralizovani model koji je NP-težak koju je moguće rešavati upotrebom neke heuristike.

Izbor izvršioca na bazi motivacionih metoda koristi unutrašnje motivacione mehanizme za promenu ponašanja. Parkerov *Alliance* [P98] i stagnacioni oporavak od Kube i Zhanga [KZ94] su dva najpoznatija primera. Gage *et al.* su obrađivali pristup na bazi emocija (afektivni) u [GMVL04]. Pored ovih metoda postoje i metode koje koriste genetske algoritme [HX11], biološki inspirisani postupci [DA10], metodi inspirisani hormonima [WW11] itd.

Već nekoliko godina postoji puno istraživanja metoda na bazi tržišnog pristupa u oblasti robotike, BSM i BSAM. Svi ovi radovi su inspirisani jednim od prvih članaka kojim su Walsh i Wellman uveli tržišnu paradigmu u decentralizovane probleme izbora izvršioca i dodele posla [WW98]. Jedan od prvih članaka kojim su Gerkey i Mataric uveli tržišni pristup u oblasti koordinacije u BSAM [GM02a] daje odličan opis oblasti istraživanja i koji problemi postoje u primeni tržišnog pristupa, ali ne daje rešenje za ove probleme.

U oblasti koordinacije među robotima, tržišni pristup su predložili Tovey *et al.* [TLJK05]. Baziran je na aukcijama koje organizuje robot ili sink (centralna jedinica), tj. onaj član u mreži koji dobije informaciju o događaju, ceni izvršenja zadatka za svakog izvršioca i potencijalnu dobit. Roboti koji se nalaze u lokalnom okruženju učestvuju, ali nivo lokalnosti nije unapred određena već zavisi od konkretnog događaja. Roboti koji učestvuju u akciji odlučuju da li da pozovu ili ne još robota, jer pozivanje samo po sebi povećava komunikacione troškove.

Dias *et al.* su sumirali istraživanja u oblasti koordinacije među robotima na bazi tržišnog pristupa u preglednom članku [DZKS06]. Aukcije mogu biti centralizovane (za sve robote) ili lokalizovane, kada samo bliski roboti učestvuju. Nisu dati podaci o komunikacionim troškovima u slučaju većih robotskih timova osim jednostavnog iskaza (Tabela 3 u [DZKS06]) da su komunikacioni troškovi proporcionalni broju robota.

Aukcije kao alat u koordinaciji se koriste još od kada je Smith objavio *Contract Net Protocol* [S81] i danas postoji više sličnih protokola, pogotovo u oblasti robotike. Jedan od dobro poznatih je aukcijski protokol *Murdoch* (Gerkey i Mataric [GM02]). On koristi metod slanja poruka svima kao sredstvo komunikacije. Pri tome *Murdoch* podrazumeva kompletan graf među robotima. Slično, Michael *et al.* u [MZKP08] i Nanjanath i Gini u [NG06], lokalne aukcije koriste kao distribuirano rešenje za dinamičku MRTA. Međutim, pretpostavka je da svi roboti koji učestvuju u aukciji mogu da komuniciraju direktno sa onim ko je pokrenuo aukciju što podrazumeva kompletan graf među robotima.

Jedan od retkih radova gde centralizovane aukcije koje pokreću roboti uz prisustvo senzora u mreži, istraživali su Coltin i Veloso [CV10]. U ovom radu naglašene su prednosti decentralizovanih rešenja.



U oblasti BSAM su Melodia *et al.* predložili prosti aukcijski protokol (SAP) u [MPGA05] i [MPGA07] sličan *Murdoch* algoritmu. Statični aktuatori su svi u međusobnom dometu podrazumevajući kompletan graf mreže i licitiraju sa cenama da reaguju na događaj u oblasti delovanja u kojoj se nalaze. Ovaj algoritam će biti polazna osnova za unapređenja i predloge novih protokola u ovoj disertaciji.

### 2.3 Otkrivanje lokacije izvršioca

*Grid Location Service* (GLS) algoritam radi podelu polja senzora u pravilne rešetkaste zone i konstruiše strukture stabla tako da svaki roditelj u stablu ima tačno četiri deteta (eng. *quad-tree*) [LJ+00]. Ovaj algoritam koristi heš funkciju, projektovanu na bazi gore objašnjenog tipa stabla, sa ciljem da upari svaki čvor (na osnovu njegovog identifikatora ID) sa jedinstvenim podskupom čvorova koji se zovu serveri lokacija (eng. *location servers*). Svaki čvor šalje informaciju o svom tekućem položaju svim svojim serverima lokacija. Čvor može dobiti informaciju o lokaciji bilo kog drugog čvora traženjem informacije od servera lokacija tog čvora. Ovaj protokol zahteva predznanje o senzorskom polju radi particionisanja. Može se desiti da je komunikaciono opterećenje veliko jer se poruke o trenutnim lokacijama i poruke sa zahtevom za lokacijom nekom čvora kreću po zigzag linijama. Pored toga GLS nema informaciju o udaljenosti jer dužina putanje otkrivanja ne mora da bude proporcionalna udaljenju izvršioca od događaja.

Upotreba geografskih heš tabela (GHT) prikazana je u [RK+02]. Čvor čuva u heš tabeli podatke na jedinstvenoj lokaciji u zavisnosti od tipa podataka i rutira podatke ka toj lokaciji korišćenjem kombinovanog *greedy-face* protokola za rutiranje. Čvorovi koji zaokružuju neku težu lokaciju planarnim grafom, čuvaju podatke o toj lokaciji; ostali čvorovi mogu da dobiju te podatke od njih. Glavni nedostatak ovakve šeme je nepostojanje lokalnog znanja jer može da se desi da neki čvor koji se nalazi blizu izvora informacije ipak šalje zahtev koji će ići mnogo daljom putanjom da dođe do informacije koja mu je u neposrednom blizini. Pored toga, moguće je postojanje zona 'uskih grla' kada se neki tip podataka učestalo traži ili se šalju podaci o njemu pa se u tim zonama javlja povećan komunikacioni saobraćaj.

Geografski protokol lokacija sadržaja (*Geography based Content Location*

*Protocol* – GCLP) koristi servere sadržaja koji periodično šalju informaciju o svom položaju u četiri pravca [TV04]. Čvorovi koji dobiju ove informacije postaju serveri lokacija sadržaja. „Ukoliko server lokacija sadržaja primi višestruke informacije za pojedini izvor, on će proslediti dalje informaciju samo o serveru sadržaja koji mu je najbliži“. Ovakvo pravilo prosleđivanja ja neformalna definicija pravila blokiranja koje je generalizovano, formalizovano i korišćeno prilikom konstrukcije informacione mreže u [LSS09]. Zbog periodičnosti slanja poruka o lokacijama, GCLP generiše veliko komunikaciono opterećenje.

Protokol koji bazira na orijentirima (eng. *landmarks*) prikazan je u [FGG06]. Ovaj protokol konstruiše određen broj globalnih stabala najkraćih putanja od unapred definisanih orijentirnih čvorova, gde svako stablo počinje od orijentirnog čvora. Podaci se čuvaju (zavisno od tipa podataka) u određenom orijentirnom čvoru i distribuiraju stablom najkraćih putanja koje počinje od orijentirnog čvora. Korisnik podataka traži podatke duž istog stabla najkraćih putanja i dobija ih kad dođe do putanje gde se podatak čuva ili do čvora koji čuva taj podatak. Ovakav način rada uzrokuje puno skupih globalnih operacija i javlja se dosta tačaka sa velikim komunikacionim saobraćajem. Pored toga, kod ovog protokola se ne obraća pažnja na energetska efikasnost i na skaliranje.

Primena pseudo-kvoruma je predložena u [AS02]. Kod ovog protokola, mreža se deli u podskupove koji se zovi kvorumi tako da se svaka dva kvoruma presecaju na način da ni jedan kvorum ne sadrži neki drugi. Da bi podržali mobilnost čvorova i skalabilnost mreže, predloženo je da izvršioc i senzori periodično i sistemski šalju poruke sa informacijama formirajući pseudo-kvorum i da se uparivanje potrage za izvršiocom i informacije o izvršiocu rade u presečnim tačkama. Mana ovog protokola je što ne garantuje da će izvršilac biti pronađen.

Tehnika 'dvostrukih pravila' (eng. *double rulings*) je predložena u [SZG09]. Podaci o izvršiocu se čuvaju duž krive umesto u nekom izolovanom čvoru ili nekoliko njih (kao npr. u GHT). Zahtev za otkrivanjem izvršioca se šalje duž druge krive koja garantovano preseca krivu sa informacijama o izvršiocu i tako obezbeđujući pronalaženje izvršioca. Ovaj protokol je u stvari proširenje GHT protokola ali sa manjim komunikacionim opterećenjem i dat je za statičke mreže.

Algoritam koji koristi hijerarhijsku dekompoziciju (HD) prikazan je u

[FGNW06]. Ovde, čvorovi mreže se klasifikuju u hijerarhijske klasterne tako da svaki čvor pripada jednom klasteru na svakom nivou. Podaci o izvršiocima se čuvaju u susednim klasterima nekog izvršioca na svim nivoima. Podaci o izvršiocu se mogu dobiti traženjem informacija od čvorova u klasteru u kojem se korisnik nalazi povećavajući nivoe sve dok se ne dođe do informacije. HD algoritam garantuje otkrivanje lokacije izvršioca, ali ima veliko komunikaciono opterećenje i potrebe za većom memorijom na čvorovima jer je kod ovog protokola potrebno znanje o globalnoj topologiji.

Otkrivanje lokacije na bazi kvoruma je predloženo u [SLJ08]. Svaki čvor distribuira svoju tekuću poziciju duž „kolona“ u mreži. Kada neki čvor želi da pronađe lokaciju nekog drugog čvora, on šalje poruku radi pretrage duž „redova“ u mreži. Red preseca kolone svih ostalih čvorova, tako obezbeđujući pronalaženje. Mana ovog protokola je što se osvežavanje informacija o tekućim lokacijama kao i pretraživanje vrši kroz celu mrežu i granični čvorovi u mreži moraju biti uključeni da bi presek bio garantovan. Pored toga, u scenariju gde su svi čvorovi kolinearni duž kolone, svaki čvor mora da čuva podatke o svim ostalim u koloni, što dovodi do potrebe čuvanja velike količine podataka i povećanja memorijskog kapaciteta čvorova.

Da sumiramo, svi algoritmi (osim GCLP [TV04]) imaju neke, ako ne sve, sledeće nedostatke: zahtev za predznanjem o mreži, učestale globalne operacije, nekonstantno memorijsko opterećenje po čvoru, pojava visoko komunikaciono opterećenih zona u mreži i nepostojanje lokalnog znanja o postojanju servisa u blizini. Prvi algoritam koji nema ovih nedostataka je algoritam na bazi informacione mreže (iMesh algoritam) [LSS09]. Ovaj algoritam će biti detaljnije objašnjen i korišćen kao algoritam za izbor izvršioca u poglavlju 7. Kombinovan sa aukcijama omogućava efikasnije pronalaženje izvršilaca i samim tim efikasniju upotrebu bežičnih senzorskih i aktuatorskih mreža.

## 3 Izbor izvršioca – idealne komunikacije

U scenarijima u ovom poglavlju uvedene su sledeće pretpostavke. Podrazumeva se da svaki izvršilac (aktuator ili robot) ima informaciju o svojoj geografskoj poziciji (koristeći npr. GPS ili neki od algoritama lokalizacije). Geografska pozicija događaja je takođe poznata najbližem senzoru koji je detektovao događaj (određena pomoću algoritama za lokalizaciju ili GPS uređaja integrisanog na senzoru). Nema izvršioca koji mogu istovremeno da izvršavaju više zadataka, tj. mogu da izvrše samo jedan zadatak u određenom vremenskom intervalu i potom da pređu na neki drugi. Senzorska mreža i mreža izvršilaca je homogena. Komunikacioni troškovi su zanemarljivi, pa tako komunikacije među sensorima, među izvršiocima i međusobno između senzora i izvršilaca ne uzrokuju nikakve troškove. Iako je otkrivanje servisa besplatno u ovim scenarijima jer nema komunikacionih troškova, metode za otkrivanje slobodnog i povoljnog izvršioca ne moraju biti trivijalne.

### 3.1 Pojedinačni događaji

Uzimajući u obzir da su komunikacije bez troškova (među sensorima, među izvršiocima i međusobno), svi troškovi potiču od mobilnosti izvršioca i ukupna minimizacija troškova se svodi na minimizaciju kretanja. Funkcija troškova izvršioca  $c_i$  je funkcija udaljenja lokacije događaja od izvršioca.

Nakon što je događaj uočen, senzori prosleđuju informaciju do najbližeg izvršioca. Ovde imamo dva različita scenarija:

1. Senzori znaju lokaciju najbližeg izvršioca. Ovo je moguće ako izvršioци periodično šalju poruke o svojim pozicijama bliskim sensorima. Tada senzori mogu odmah da obave rutiranje ka tom izvršiocu i proslede informacije o događaju.

2. Sensori ne znaju lokaciju najbližeg izvršioca. Tada senzor koji je otkrio događaj šalje informaciju o njemu celoj mreži. Svaki izvršilac odgovara, prvi odziv se uzima u obzir (u slučaju da količina energije nije kritična) i taj izvršilac dobija zadatak, a ostali se informišu o tome. Na ovaj način se dobija kraće vreme odziva u odnosu na rešenje kada senzor čeka odgovor od svih izvršilaca. Alternativno, izvršilac koji je primio zahtev može da obavesti sve izvršioce u blizini o odluci da on dobija posao i time zaustavi ostale izvršioce. Međutim ima još detalja o kojima treba voditi računa. Koliko dugo treba da čeka neki izvršilac da dobije poruku od potencijalno bližeg? Drugo moguće rešenje je kada nekoliko izvršilaca krene u izvršenje (možda čak svi), ali kasnije samo jedan ostaje koji prvi stigne na mesto događaja. Ovde je naravno problem gubljenja energije na kretanje onih izvršilaca koji neće dobiti taj zadatak.

Ako je energija koju imaju izvršioци na raspolaganju ograničena (npr. baterijsko napajanje koje je najčešće), izbor najboljeg izvršioca se vrši dodelom posla najbližem izvršiocu koji ima dovoljno energije da dođe do mesta događaja i obavi potreban zadatak. U ovom slučaju to ne mora biti i izvršilac najbliži događaju. Taj izvršilac može da ima u svojoj listi poslova još nekoliko zadataka i da potroši energiju iz napajanja pre nego što stigne do mesta događaja. Donošenje odluke u ovakvom scenariju se može postići dogovaranjem među izvršiocima. Svrha dogovaranja je pronalaženje najbližeg izvršioca i samim tim najmanji utrošak energije za tim izvršilaca. Ako je cilj maksimizacija vremena potrebnog da neki od izvršilaca oстане bez energije, tada je potrebna drugačija strategija koja balansira utrošak energije među izvršiocima.

Pojava događaja unapred nije poznata i mi ne pretpostavljamo *a priori* znanja o događajima. Sve odluke se donose nakon pojave događaja (ovo se ponekad naziva reaktivno odlučivanje). Jedna od mogućih strategija koja balansira opterećenje među izvršiocima je da kada izvršioци dobiju informacije o događaju od senzora, sračunaju količnik  $m*d/R_E$  gde je  $m$  konstanta,  $d$  je udaljenje od događaja, a  $R_E$  je preostala energija. Onaj izvršilac koji minimizuje ovaj količnik preuzima posao. Ponovo, detalji o komunikaciji da bi se ovo postiglo najčešće nisu trivijalni. Druga mogućnost je da izvršioци minimizuju  $m*d/(R_E - m*d)$ . Pronalaženje dobre strategije za balansiranje opterećenja među izvršiocima prevazilazi okvire ove disertacije.

U oblasti robotike, puno istraživanja u oblasti koordinacije među robotima je rađeno upotrebom tržišnog pristupa [DZKS06]. Upotreba tržišnog pristupa za scenario jednostrukih događaja je povezana sa prethodno pomenutom strategijom o minimizaciji  $m*d/R_E$ . Osnovna ideja je da svaki robot sračuna  $m*d/R_E$ , i da u skladu sa tim licitira na aukciji koju započinje senzor koji je detektovao događaj na koji se očekuje odziv nekog od robota. Onaj robot koji je dao najvišu licitaciju dobija posao. Alternativno, najbliži robot može da organizuje aukciju umesto senzora. Lokalizovano rešenje bi bilo ono koje u aukcijama uključuje samo *bliske* robote, dok bi kod centralizovanog bili uključeni *svi* roboti.

### 3.2 Višestruki događaji

Kod ovog modela se takođe podrazumeva da su sve relevantne informacije poznate svakom izvršiocu jer su komunikacije besplatne. Izvršiocima mogu biti dodeljeni pojedinačni zadaci ili nekoliko sukcesivnih zadataka. Događaji mogu da se pojavljuju paralelno ili sekvencijalno. Mogući su događaji koji zahtevaju više od jednog izvršioca da bi se obavili uspešno.

Ako se događaji pojavljuju sekvencijalno i vreme između dva događaja je dovoljno dugo, tada se ovaj problem svodi na niz problema pojedinačnih događaja o kojima je bilo reči u prethodnom odeljku. Ako se svi događaji pojavljuju paralelno, postoji nekoliko različitih slučajeva. Pre nego što pređemo na detalje ovih različitih scenarija, u Tabeli 3.2.1 pokazano je koje su to situacije. Primeri su dati za slučaj četiri izvršioca-aktuatora (A1, A2, A3 i A4) i četiri zadatka (Z1, Z2, Z3 i Z4).

Tabela 3.2.1 Primeri dodele posla

	A. STA-SAT				B. MTA-SAT				C. STA-MAT				D. MTA-MAT			
	Z1	Z2	Z3	Z4	Z1	Z2	Z3	Z4	Z1	Z2	Z3	Z4	Z1	Z2	Z3	Z4
<b>A1</b>	1				1	1	1		1		?	?	1	1		
<b>A2</b>			1						1		?	?	1			
<b>A3</b>		1						1		1	?	?			1	
<b>A4</b>				1						1	?	?			1	1

Prvi scenario je pojedinačna dodela zadatka – pojedinačni izvršioc (single task

*assignment-single actuator tasks* STA-SRT). Ovde je moguće imati samo jedan zadatak po izvršiocu i samo jednog izvršioca za zadatak. Drugi scenario je višestruka dodela zadataka – pojedinačni izvršioci (*multiple task assignment-single actuator tasks* MTA-SAT). U ovom scenariju neki od izvršioca dobijaju više uzastopnih zadataka (npr. A1 u B scenariju u Tabeli 3.2.1 ima tri uzastopna zadatka). Treći scenario je pojedinačna dodela zadatka – zadaci koji zahtevaju više izvršilaca (*single task assignment-multiple actuator tasks* STA-MAT). Ovde su zadaci takvi da zahtevaju više izvršilaca, ali je svakom izvršiocu moguće dodeliti samo po jedan zadatak. U C scenariju u Tabeli 3.2.1, gde su zadaci takvi da je potrebno po dva izvršioca, zadaci Z3 i Z4 su ostali neraspodeljeni. Poslednji i najkompleksniji scenario je višestruka dodela zadataka – zadaci koji zahtevaju više izvršilaca (*multiple task assignment-multiple actuator tasks* (MTA-MAT). Ovde zadaci Z1 i Z3 zahtevaju po dva izvršioca, ali i izvršioci mogu da dobiju više od jednog zadatka.

### 3.2.1 Pojedinačna dodela zadatka – pojedinačni izvršioci

Ukoliko se događaji odvijaju paralelno i svaki izvršilac dobija različiti pojedinačni zadatak tada je problem moguće rešavati kao jedan od optimalnih problema dodeljivanja (*optimal assignment problem*) koji su detaljno obrađeni u [GM04]. Glavni cilj je pronaći takvu raspodelu poslova koja minimizuje ukupno pređeni put izvršilaca i na taj način minimizuje potrošnju energije na kretanje. Ako je pri tome broj izvršilaca jednak broju zadataka, takav scenario spada u klasične probleme dodeljivanja (*classical assignment problem*) [P07] koji su poznati i pod nazivom linearno sumirani problem dodeljivanja (*linear sum assignment problem* [BDM09]).

Ovde formulišemo problem na sledeći način da bi odgovarao našem slučaju za BSAM:

Pronaći:  $\mathbf{T} = \{t_{ij}\}$  – matrica dodele;  $i = 1, \dots, N_A, j = 1, \dots, N_D$  (3.1)

$N_D$  – broj događaja,  $N_A$  – broj aktuatora

Minimizovati:  $C_{\text{MIN}} = \sum \sum m [1/\text{meter}] * d_{ij} [\text{m}] * t_{ij}$  (3.2)

$m$  – cena kretanja po jednom metru

$d_{ij}$  – udaljenost između događaja  $j$  i aktuatora  $i$

Uz uslove:

$$\sum_{i=1}^{N_A} t_{ij} = 1, \forall j \quad (3.3)$$

$$\sum_{j=1}^{N_D} t_{ij} = 1, \forall i \quad (3.4)$$

Uslov (3.3) obezbeđuje da nije moguće dodeliti više aktuatora jednom zadatku, dok uslov (3.4) obezbeđuje da nema višestrukih dodela zadataka jednom aktuatoru.

Ako je potrebno uzeti i energiju u obzir, prethodni problem se može proširiti u slučaj problema dodeljivanja sa bočnim uslovima (*assignment problem with side constraints* [P07]):

$T_{ij}$  – zadatak  $j$  dodeljen aktuatoru  $A_i$

$$\sum_{j=1}^{N_D} E(T_{ij}) * t_{ij} \leq E_I(A_i); i = 1, 2, \dots, N_A \quad (3.5)$$

$$\text{gde je } E(T_{ij}) = m * d(T_{ij}, T_{i(j-1)}) \quad (3.6)$$

Ukupno potrošena energija mora biti u okviru raspoložive energije ( $E_I$ ) za svaki aktuator. Ovo je određeno uslovom (3.5). Energija koja se troši na izvršenje zadatka sastoji se od energije koja je potrebna da aktuator dođe na mesto izvršenja zadatka i to je određeno jednačinom (3.6). Ukoliko energija za izvršenje zadatka nije konstantna (npr. postoje različite vrste zadataka), onda je moguće dopuniti jednačinu (3.6). Dodavanje uslova vezanih za energiju (kao relevantnog faktora u BSAM) u model deluje opravdano na prvi pogled. Međutim u praktičnim primerima ostali troškovi i troškovi energije po zadatku su veoma korelisane veličine i potrebno je pokazati da je proširenje modela opravdano.

Određena vremenska ograničenja (vreme odziva, vreme izvršenja itd.) koja su uobičajena za primene i rad u realnom vremenu do sada nisu razmatrane. Ukoliko je to bitan deo, moguće je dodati vreme potrebno da se izvrši određena akcija u formi još jednog uslova u modelu:



$$t_M(T_{ij}) \leq t_r \quad (3.7a)$$

gde je  $t_M$  vreme potrebno da se dođe u zonu akcije (ovo se može izračunati ako se zna brzina kretanja aktuatora), a  $t_r$  vremenska konstanta. Uslov (3.7a) može da se proširi kompletnim uslovom koji uključuje i vreme potrebno za izvršenje zadatka  $t_T$ :

$$\sum_{i=1}^{N_A} \sum_{j=1}^{N_D} (t_M(T_{ij}) + t_T) \leq t_o \quad (3.7b)$$

gde je  $t_o$  posmatrani vremenski interval.

U ovom centralizovanom modelu podrazumeva se da je sve unapred poznato (svi troškovi, svi zadaci, sve pozicije, sve energije i sva vremena). Komunikacioni troškovi se zanemaruju. Pronalaženje matrice dodele posla  $T$  je NP-težak problem, ali je moguće naći rešenje koristeći neku heuristiku za rešavanje ovog optimizacionog problema (npr. *Hungarian* algoritam [K55]). Za primene i rad u realnom vremenu nije moguće sve znati unapred, tako da ovakav model i njegovo rešavanje nije pogodno za realne primene. Međutim može biti dobro za potrebe poređenja sa drugim rešenjima. Pored toga, ukoliko ima više zadataka nego izvršilaca problem nije moguće rešiti. Ako se događaji ne pojavljuju paralelno, već po grupama, tada se može problem rešiti korišćenjem iterativne verzije optimalnog problema dodele.

Ako svi zadaci nisu poznati *a priori* (npr. događaju se dinamički ili čak proizvoljno), problem formulisan u (3.2) ne može se koristiti. Potencijalna heuristika koju je moguće koristiti je neka od varijanata pohlepnog algoritma (*Greedy algorithm*). Algoritam bi bio: kada se pojavi novi zadatak dodeli ga izvršiocu koji unosi najmanji trošak ili onom koji je najbolji za taj zadatak kao u *Murdoch* algoritmu [GM02] (pohlepni tržišni algoritam). Pitanje šta znači najbolji zavisi od metrike koja se koristi.

U [AGB10], autori predlažu centralizovano rešenje koje podrazumeva idealne komunikacije bez troškova. Višestruki događaji se pojavljuju paralelno i broj robota je jednak broju događaja. Pretpostavka je da svi roboti i događaji imaju informacije o svim ostalim robotima i događajima. Roboti se posmatraju kao 'muškarci', a događaji kao 'žene'. Svi oni prave listu poželjnih 'partnera' na bazi udaljenosti tako da su bliži poželjniji. Nakon toga, se vrši uparivanje robota i događaja koristeći *Gale-Shapley*

algoritam stabilnog braka (*Stable Marriage Algorithm* - SM). Ovaj algoritam raspoređuje parove na način tako da su svi elementi 'oženjeni' i da su svi 'brakovi' stabilni. To znači da nema alternativnog uparivanja u kojem bi i 'muškarac' i 'žena' imali poželjnijeg partnera od trenutnog.

U [WPT10], drugačije rešenje istog problema je predloženo uz iste pretpostavke (jednak broj robota i događaja i globalno znanje). Algoritam pod nazivom Algoritam Maksimalnog Uparivanja (*Maximum Matching Algorithm* - MMA) je predložen, koji umesto minimizacije ukupno pređenog puta, minimizuje standardnu devijaciju između pređenih puteva pojedinačnih robota. Ovo vodi ka bolje balansiranom opterećenju robota što je važno za maksimizaciju vremena rada sistema u scenarijima sa više krugova.

### 3.2.2 Višestruka dodela zadataka – pojedinačni izvršioci

Ukoliko imamo scenario u kojem ima više zadataka nego izvršioca, tada je izvršiocima potrebno dodeliti više uzastopnih zadataka. Intuitivno, u slučajevima kada resursi izvršioca i prioritet zadataka nije bitan (npr. nelimitiran kapacitet napajanja), broj zadataka je moguće podeliti u odgovarajući broj podskupova  $w$ . Prvih  $(w - 1)$  podskupova sastoji se od broja zadataka koji je jednak broju izvršioca i tada imamo slučaj klasičnog problema dodele (*classical assignment problem*). Poslednji skup zadataka sastoji se od broja zadataka koji je manji ili jednak broju izvršioca i on se svodi na klasični ili optimalni problem dodeljivanja. Glavni problem kod ovakvog pristupa je kako podeliti događaje u podskupove što može dovesti do veoma suboptimalne dodele zadataka gledano na bazi ukupno pređenog puta ili potrebnog vremena.

Ako su resursi izvršioca važan aspekt ili su ograničeni (što je najčešći slučaj), moguće je problem formulisati i kao generalizovani problem dodele (*Generalized Assignment Problem* – GAP) [P07]. GAP je primer problema dodele koji uključuje i ograničenja u resursima.

Optimizacioni problem koji treba rešiti je minimizovati troškove dodele  $N_D$  zadataka za  $N_A$  izvršilaca, pri čemu se podrazumeva da su resursi izvršilaca ograničeni i da se uvek dodeljuje jedan zadatak jednom izvršiocu. Problem primene GAP za naš

scenario je u tome što GAP podrazumeva konstantne troškove, dok su u našem slučaju troškovi u funkciji udaljenosti koju izvršilac treba da pređe da bi stigao do zone gde se nalazi događaj na koji treba reagovati. Zbog toga troškovi nisu konstantni, a mogu biti i delom nepoznati unapred. U slučajevima kada su troškovi konstantni (npr. kada zavise samo od tipa zadatka) tada se GAP može koristiti kao centralizovano rešenje za višestruku dodelu zadataka pojedinačnim izvršiocima.

Za rešavanje problema višestruke dodele zadataka pojedinačnim izvršiocima moguće je koristiti i sledeću heuristiku. Inicijalno, rešiti problem za slučaj jednakog broja događaja i izvršilaca kao što je razmatrano u prethodnom odeljku i nakon toga koristiti Pohlepni algoritam za dodelu preostalih zadataka kada neki od izvršilaca postane slobodan (ovo je nazvano aproksimacionim ST-SR-TA u [GM04]).

U [WPT10], autori razmatraju problem višestruke dodele zadataka na sledeći način. Broj izvršilaca je  $N_A$  i broj događaja je  $N_D$  (uz  $N_A < N_D$ ). Prvo, skup događaja je podeljen u  $R$  klastera na bazi jednog od predložena tri načina (*k-means*, *MaxMin* ili balansirani). Tada se MMA algoritam, objašnjen u prethodnom odeljku, primenjuje na skup izvršilaca i skup klastera, gde je udaljenost između izvršioca i klastera definisana kao minimalna udaljenost između tog izvršioca i bilo kojeg događaja u tom klasteru. Kada se izvršiocu upare sa klasterima, oni odlaze do najbližeg događaja u klasteru sa kojim su upareni i potom sekvencijalno obilaze sve događaje u klasteru primenom algoritma koji rešava Problem trgovačkog putnika (*Traveling Salesman Problem* – TSP).

### 3.2.3 Pojedinačna dodela zadatka – zadaci koji zahtevaju više izvršilaca

Problem dodele zadataka u slučaju kada su događaji takvi da zahtevaju više od jednog izvršioca je razmatrano u [GM04]. Primeri takvih zadataka su guranje veoma teškog objekta i koordinirano istraživanje terena. Ovakvi problemi se mogu posmatrati kao problemi particionisanja skupova (*Set Partitioning Problem* – SPP). To je strogo NP-težak problem za čije rešavanje postoji nekoliko heuristika. Ostaje da se vidi da li su te heuristike dovoljno dobre da bi se mogle primeniti na probleme dodele posla.

### 3.2.4 Višestruka dodela zadataka – zadaci koji zahtevaju više izvršilaca

Ukoliko izvršiocima može biti dodeljeno više od jednog zadatka i postoje događaji koji traže više od jednog izvršioca, ovakvi scenariji spadaju u veoma teške probleme. Prema našim najboljim saznanjima, za sada, nema radova koji daju rešenje za ovakve probleme i to predstavlja jedan od budućih istraživačkih izazova.

## 4 Izbor izvršioca – nezanemarljivi komunikacioni troškovi

U narednim scenarijima u ovom poglavlju, komunikacioni troškovi nisu zanemarljivi. U tim slučajevima problem pronalaženja lokacije nekog od izvršioca unosi značajne troškove i veoma je važna tema u ovoj klasi scenarija. Pretpostavlja se da je uvek najmanje jedan senzor u komunikacionom dometu izvršioca čime je uvek omogućena komunikacija između izvršilaca i senzora.

### 4.1 Pojedinačni događaji

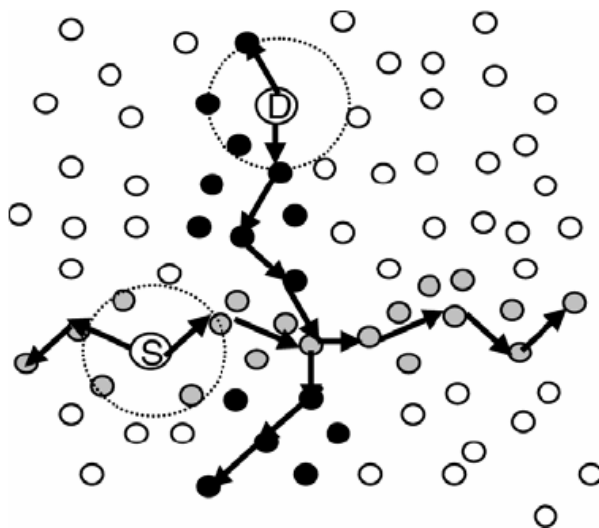
Uzimajući u obzir ranije uočene tri vrste komunikacionih paradigmi u BSAM i BSRM (komunikacija među sensorima, među izvršiocima i između senzora i izvršioca), postavlja se pitanje kako odrediti koji je izvršilac *najbolji*. Zbog mobilnosti izvršilaca, arhitektura mreže nije statička; stoga tekuće pozicije izvršilaca nisu poznate sensorima. Nakon pojave događaja, senzor u blizini rutira informaciju o događaju putem komunikacije među sensorima. Postavlja se pitanje kako rutirati ako lokacije izvršilaca nisu unapred poznate?

Prva ideja koja se javlja je korišćenje plavljenja (eng. *flooding*) mreže radi slanja informacija. U tom slučaju imamo komunikacionu kompleksnost reda  $O(n)$ . U stvari, za najbolji algoritam plavljenja (ako takav postoji), gde se svaka poruka ponovo šalje samo jedanput i svaki čvor prima samo po jednu kopiju poruke, ukupni troškovi izraženi u broju komunikacionih skokova bi bili jednaki broju čvorova uvećanom za broj izvršilaca. Stoga plavljenje odlikuju vrlo visoki komunikacioni troškovi, ograničen propusni opseg, loša skalabilnost i poželjno je potražiti drugačija rešenja.

Lokalizovano (ili delimično) plavljenje sa ograničenjem na doseg u okruženju

(mereno brojem komunikacionih skokova) može da se koristi, ali tada ne može da se garantuje isporuka (npr. svi izvršioci su van dosega). Čak i u slučaju da je izvršilac pronađen mogući su visoki komunikacioni troškovi. Različiti načini, problemi pri korišćenju plavljenja i moguća poboljšanja su razmatrana u [SW04].

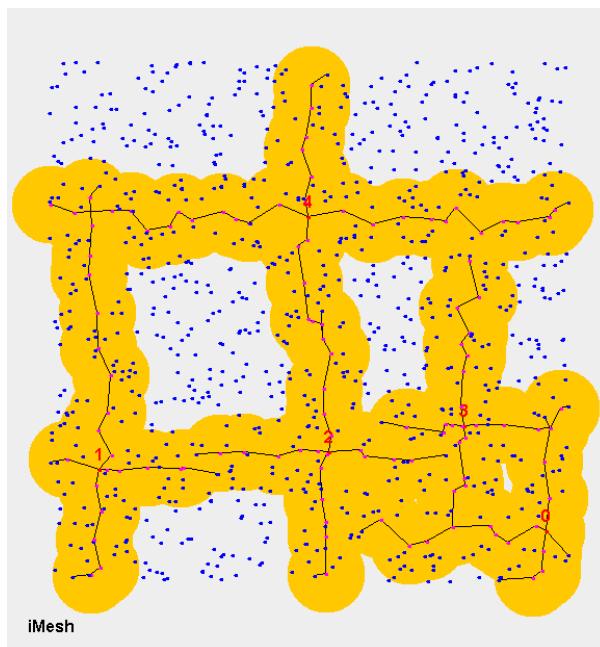
Prethodno navede metode imaju velike komunikacione troškove (komunikaciona kompleksnost je reda  $O(n)$ , gde je  $n$  broj čvorova u mreži) jer ne postoji informacija o trenutnoj lokaciji izvršioca. Da bi prevazišli ovaj nedostatak postoje različiti pristupi. Stojmenović *et al.* su prikazali metode na bazi kvoruma za pronalaženje lokacije izvršioca u [SLJ08]. Ova metoda se karakteriše dobrom skalabilnošću, visokim stepenom uspešnosti u pronalaženju lokacije izvršioca, skoro garantovanom isporukom informacije izvršiocu i kompleksnošću reda  $O(\sqrt{n})$ .



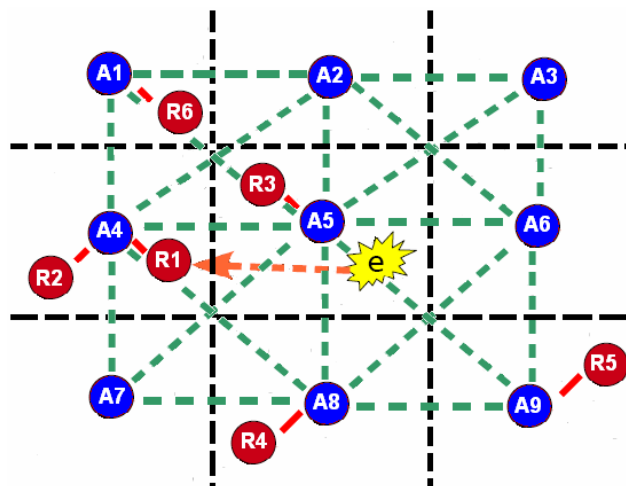
Slika 4.1.1 Konstrukcija kvoruma i način pretraživanja

Pronalaženje lokacije izvršioca na bazi kvorum metode je ilustrovano na Slici 4.1.1. Osnovna ideja je sledeća. Čvor  $S$  ima informaciju o događaju i on šalje zahtev za izvršiocem u pravcu istoka i zapada. Izvršilac  $D$  šalje informacije o svojoj lokaciji u pravcu severa i juga. Može se pokazati da uvek postoji tačka sretanja tj. čvor koji ima informacije i o traženju lokacije od strane senzora  $S$  kao i o tome gde se nalazi izvršilac  $D$ . Mnogo manji komunikacioni troškovi su posledica kompleksnosti reda  $O(\sqrt{n})$  i mnogo manje razmene poruka koja je potrebna da se dođe do lokacije izvršioca nego kod metoda na bazi plavljenja.

Iako prethodno opisan metod nalaženja lokacije izvršioca značajno smanjuje komunikacione troškove, poboljšanje u smislu povećanja verovatnoće pronalaženja *najbližeg* izvršioca je moguće konstrukcijom informacione mreže *iMesh* kako je objašnjeno u [LSS09], a potom primenjeno i u [MLMS12]. U inicijalnoj fazi svi izvršioci šalju informacije o svojim pozicijama kroz statičku senzorsku mrežu u četiri pravca (sever, istok, jug i zapad) tako što šalju registracione poruke. Kada se dve registracione putanje poruka sretnu, tada se primenjuje pravilo blokiranja koje određuje da registraciona poruka koja je dalja od svog izvora prestaje dalje da se šalje dok ona druga nastavlja propagaciju. Čvorovi duž putanje slanja registracionih poruka čuvaju informaciju o pozicijama izvršioca koji je inicirao slanje tako formirajući *iMesh* strukturu. Ovo je prikazano na Slici 4.1.2. Ova struktura se koristi za otkrivanje servisa, tj. slobodnih izvršilaca. Kada se desi neki događaj, sensor koji ga je otkrio započinje pretragu u četiri pravca slanjem poruka na sličan način kao što se šalju i registracione poruke. Kada jedna od poruka za pretragu dođe do čvora koji pripada *iMesh* strukturi, taj čvor šalje odgovor nazad sa informacijom o slobodnom izvršiocu. Kada sensor koji je inicirao pretragu dobije sve odgovore, on određuje lokaciju najbližeg izvršioca sa velikom tačnošću.



Slika 4.1.2 *iMesh* struktura sa 5 izvršilaca i žuto označenim zonama znanja



Slika 4.1.3 Arhitektura na bazi agenata (senzori se ne vide već samo izvršioci - roboti)

Jedna od ideja za pronalaženje izvršioca podrazumeva ćelijsku ili klaster arhitekturu mreže gde se u svakoj ćeliji nalazi po jedan centralizovan statičan senzorski čvor kojeg zovemo *agent* (videti Sliku 4.1.3), a ostali senzori su postavljeni na slučajan način. Agent ima informaciju o lokacijama svih izvršilaca unutar njegove ćelije. Da bi odredili koji je izvršilac najbliži moguća su dva scenarija. Prvi, senzorski čvor koji je otkrio odgovarajući događaj prosleđuje informaciju o događaju agentu unutar svoje ćelije. Ukoliko nema izvršioca koji je slobodan u toj ćeliji, agent započinje pretragu komunicirajući sa susednim agentima koristeći pretragu po koloni/redu ili neki drugi metod. Alternativno senzorski čvor može da šalje informaciju do prvog robota (kao što je ilustrovano na Slici 4.1.3). Pokazano je da je direktno rutiranje ka agentima u poređenju sa rutiranjem prema robotima mnogo bolje u smislu manjih komunikacionih troškova [GM11].

Nakon što je neki od izvršilaca pronađen, ostaje pitanje da li je on *najpogodniji* da reaguje na događaj. U slučaju da je energija izvršioca neograničena i nema vremenskih ograničenja za izvršenje zadatka, izabrani izvršilac jeste najbolji. U opštem slučaju, za potrebe određivanja najpogodnijeg izvršioca, komunikacija među izvršiocima je neophodna i mora se uzeti u obzir. Izvršilac koji je pronađen se naziva kolektor i on šalje  $N - 1$  poruka ostalim izvršiocima gde je  $N$  broj izvršilaca. Isti broj poruka se šalje nazad, ukoliko su svi izvršioci slobodni i svi izvršioci mogu da komuniciraju međusobno *direktno*. Poslednja pretpostavka je smisljena samo u



slučajevima kada je većina izvršilaca međusobno u komunikacionom dometu. U ovom slučaju potrebno je najviše  $2(N - 1)$  poruka da se dođe do odluke ko je najpogodniji i tada su komunikacioni troškovi jednaki  $2(N - 1)$ .

Očigledno je da direktna komunikacija među izvršiocima nije moguća za situacije gde je prostor koji se posmatra znatno veći od komunikacionog radijusa izvršilaca. Rutiranje koje je tada neophodno unosi dodatne komunikacione troškove. Tada izvršioci mogu da komuniciraju samo sa susednim tj. *bliskim* izvršiocima. Definicija bliskosti izvršioca se može dati na bazi komunikacionog dometa ili na bazi broja komunikacionih skokova. Najjednostavniji algoritam u ovom slučaju je lokalizovano plavljenje sve do suseda koji je udaljen  $k$  komunikacionih skokova.

Da bi dalje smanjili komunikacione troškove koji su posledica komunikacije među izvršiocima mogu se koristiti aukcijski agregacioni algoritmi [MMS10]. Oni predstavljaju glavnu temu ove disertacije i biće obrađeni u narednim poglavljima. Pre svega biće prikazana unapređenja prostog aukcijskog algoritma datog u [MPGA05] i sličnog algoritma pod imenom *Murdoch* u [GM02], potom nekoliko novih aukcijskih agregacionih algoritama i na kraju primena aukcijskih agregacionih algoritama radi poboljšanja izbora izvršioca nakon što je on već određen korišćenjem *iMesh* algoritma [LSS09].

## 4.2 Višestruki događaji

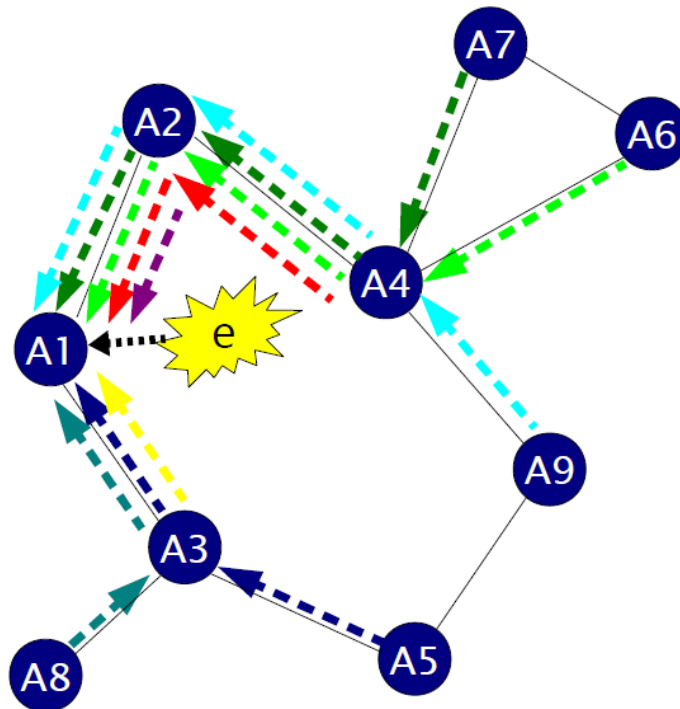
U slučaju kada se višestruki događaji pojavljuju paralelno, distribuirana verzija problema pronalaženja izvršioca može da se podeli u dve faze. U prvoj fazi se radi traženje nekog od izvršilaca korišćenjem npr. kvorum ili *iMesh* metode opisane u prethodnom odeljku. U ovoj fazi, kada svi senzorski čvorovi koji su otkrili odgovarajuće događaje pronađu bar po jednog izvršioca, oni šalju poruke izvršiocima sa zahtevom za dodelu posla. U toku ove faze izvršioci prave lokalne liste zahteva. Problem koji se pojavljuje je činjenica da jedan izvršilac može dobiti zahteve za dodelom posla koji su geografski značajno udaljeni. Pored toga moguće je da jedan izvršilac dobije mnogo zahteva, a neki ni jedan. Zbog toga je potrebna komunikacija među izvršiocima koja omogućava lokalnu razmenu zahteva na način koji omogućava

nalaženje raspodele poslova koja je bliska optimalnoj u smislu što veće energetske efikasnosti tj. što manjih troškova. Da bi se postigao ovaj cilj moguće je koristiti neke od aukcijskih algoritama, ali ovo je oblast koja još nije istražena. Posebno u dve klase ovih problema, a to su: pojedinačna dodela zadatka – zadaci koji zahtevaju više izvršilaca (*single task assignment-multiple actuator tasks* STA-MAT) i najkompleksniji scenario višestruka dodela zadataka – zadaci koji zahtevaju više izvršilaca (*multiple task assignment-multiple actuator tasks* (MTA-MAT)).

# 5 Unapređenja prostog aukcijskog algoritma

Prosti aukcijski algoritam za bežične senzorske i aktuatorске mreže (označavamo ga sa SAP) su predložili Melodia *et al.* u [MPGA05] i [MPGA07] za slučaj kada su svi aktuatori u međusobnom komunikacionom dometu (kompletna mreža). Osnovna ideja je da nakon pojave i detekcije događaja od strane nekog senzora, prvi aktuator koji dobije informaciju (nazvan kolektor), organizuje aukciju među preostalim aktuatorima sa ciljem da se utvrdi koji je najpogodniji da reaguje na dati događaj. Razlog da se utvrdi koji je najbolji je potreban zato što kolektor ne mora da bude i najbolji koji će da reaguje i to zavisi pre svega od kriterijuma po kojem se utvrđuje. Sličan algoritam pod imenom *Murdoch* su predložili u oblasti robotike Gerkey i Mataric u [GM02].

Nakon što svi aktuatori dobiju informaciju o događaju od kolektora (informaciona faza), oni vraćaju povratnu poruku sa cenom koštanja ukoliko bi oni reagovali na događaj (aukcijska faza). U ovoj disertaciji prosti aukcijski protokol koristimo u slučaju mreža koje nisu kompletne već multi-hop mreže. Tada prilikom informacione faze poruke se šalju od kolektora ka svim aktuatorima na način da svaki aktuator kada dobije poruku prosledi je samo jedanput čime se kreira stablo komunikacionog grafa. Ovde se pojavljuje situacija da neki aktuatori dobijaju poruku više puta, ali ta u nekim slučajevima problematična situacija nije razmatrana u okviru ove disertacije. U aukcijskoj fazi svaki aktuator šalje informaciju posebnom putanjom (linije različitih boja na Slici 5.1). Pretpostavka je da je algoritam rutiranja povratnih poruka takav da one idu putanjama sa najmanjim brojem hopova, dakle najkraćim. Kada dobije sve odgovore, kolektor odlučuje koji je aktuator najbolji da reaguje i šalje mu informaciju o tome.



Slika 5.1 Ilustracija zasebnih povratnih putanja svakog aktuatora (SAP algoritam)

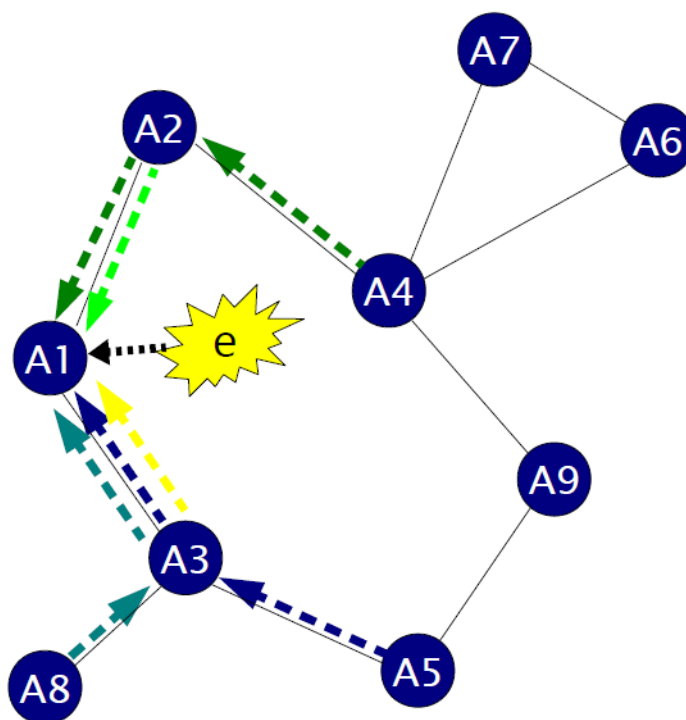
Prosti aukcijski algoritam uvek pronalazi najboljeg izvršioca jer u aukciji učestvuju svi izvršioci tako da ovaj algoritam garantuje mogućnost određivanja najboljeg izvršioca. Sa druge strane, osnovna mana ovog rešenja je što su visoki komunikacioni troškovi jer se čeka odgovor od svih aktuatora. Pored toga je i vreme potrebno da se dođe do odluke koji je aktuator najbolji da reaguje dosta dugačko u poređenju sa ostalim algoritmima.

U ovoj disertaciji smo predložili dva poboljšanja ovog algoritma. Prvi predstavlja lokalizaciju SAP algoritma i ima za cilj smanjenje komunikacionih troškova uz smanjenu optimalnost odlučivanja, a drugi je proširenje upotrebom tzv. pohlepnog (*greedy*) algoritma koji nastoji da poboljša smanjenu optimalnost odlučivanja.

### 5.1 Lokalizovani prosti aukcijski algoritam (*k*-SAP)

U cilju smanjenja komunikacionih troškova, kao i vremena potrebnog da se donese odluka o tome koji je izvršilac najbolji, moguće je koristiti sledeći lokalizovani

algoritam. Umesto da se šalje informacija o događaju svim izvršiocima u mreži i čeka odziv od svih, kod lokalizovanog prostog aukcijskog algoritma informacija se šalje izvršiocima koji su na  $k$  hopova udaljeni od kolektora. Samo od ovih izvršioca se očekuje odgovor. Za slučaj  $k = 2$  i  $N = 9$  ilustracija 2-SAP algoritma data je na Slici 5.1.1. Optimalnost odlučivanja je ista za SAP i za 2-SAP jer je moguće naći najbližeg aktuatora u dva hopa (to je A4). Međutim broj poruka potrebnih za odlučivanje upotrebom SAP algoritma je 17, a upotrebom 2-SAP algoritma je svega 8.

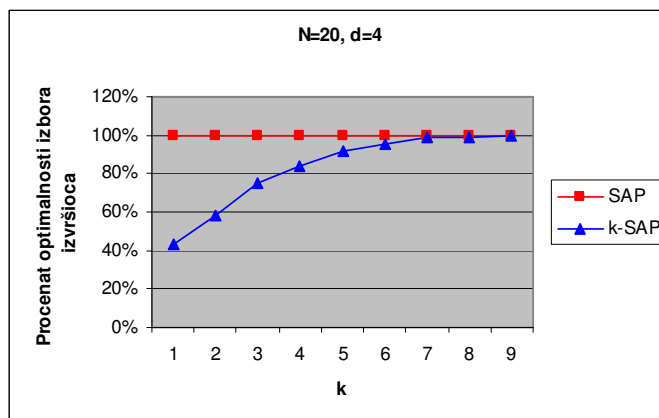


Slika 5.1.1 Ilustracija 2-SAP algoritma i prednosti u odnosu na SAP

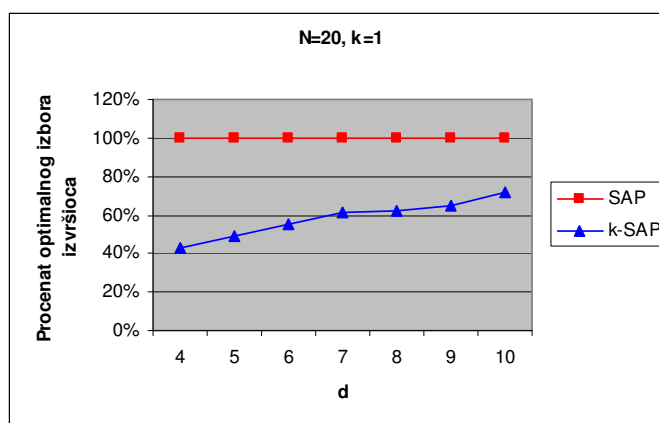
Mana ovog algoritma je što se ne može očekivati optimalan rezultat u svim scenarijima jer na aukciji ne učestvuju svi izvršiocima, za razliku od prostog aukcijskog algoritma. Međutim, komunikacioni troškovi su znatno manji u poređenju sa prostim aukcijskim protokolom.

Radi ilustracije posmatraćemo rezultate za scenario sa  $N = 20$  izvršilaca. SAP algoritam uvek pronalazi optimalnog izvršioca, dok lokalizovana verzija ovog algoritma ( $k$ -SAP) dostiže 100% optimalnost samo za male ( $N \leq 10$ ) i guste mreže ili za velike vrednosti  $k$  (videti Sliku 5.1.2.). Za veće mreže ili manje vrednosti  $k$  ova vrednost se kreće između 20 i 70 procenata (videti Sliku 5.1.3 i rezultate u [MMS09]). Može se

zaključiti da što je mreža gušća, procenat optimalnih izbora je veći.



Slika 5.1.2 Poređenje optimalnosti izbora izvršioca  $k$ -SAP algoritma u odnosu na SAP u funkciji  $k$



Slika 5.1.3 Poređenje optimalnosti izbora izvršioca  $k$ -SAP algoritma u odnosu na SAP u funkciji gustine mreže  $d$

Tabela 5.1.1 Poređenje komunikacionih troškova za velike i retke mreže

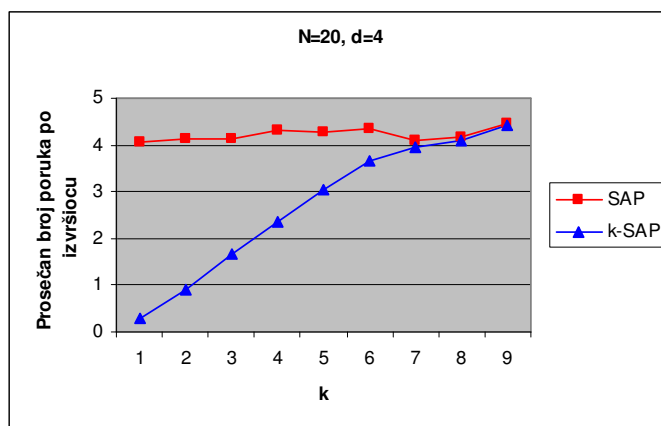
$k$	SAP	$k$ -SAP	SAP/ $k$ -SAP
1	11,82	0,06	197,00
2	11,46	0,2	57,30
3	11,4	0,38	30,00
4	10,91	0,68	16,04
5	11,16	1,04	10,73
6	11,16	1,46	7,64
7	10,71	2,04	5,25
8	12,6	2,26	5,58
9	11,29	3,18	3,55

Optimalnost izbora izvršioca korišćenjem SAP algoritma ima znatno veće komunikacione troškove od  $k$ -SAP (videti Sliku 5.1.4 i Sliku 5.1.5). U nekim slučajevima su ti troškovi i do skoro 200 puta veći što se vidi u Tabeli 5.1.1 gde su dati

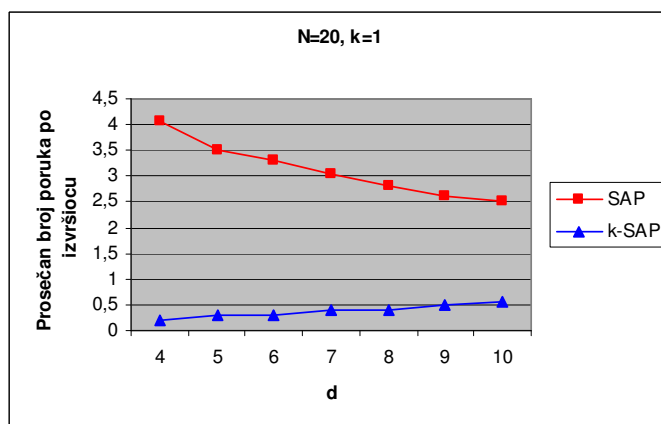
komunikacioni troškovi za slučaj  $N = 100$  i  $d = 4$ , dakle za velike i retke mreže.

Na Slici 5.1.5, može se uočiti osobina SAP algoritma da sa povećanjem gustine mreže dolazi do blagog smanjenja komunikacionog opterećenja. Ovo se može objasniti na sledeći način. SAP algoritam konsultuje sve izvršioce u mreži koja je povezana, a povećanjem gustine mreže se izvršioци raspoređuju na takav način da svaki izvršilac ima u proseku više komšija nego u ređim mrežama. Zbog svega ovoga je potrebno manje poruka da bi se zasebnim putanjama rutiranja stiglo do izvršioca koji je inicirao aukciju. Dakle izvršioци su bliži i lakše komuniciraju pa je potreban manji broj poruka.

Pored toga može se uočiti i tendencija blagog porasta prosečnog broja poruka za  $k$ -SAP algoritam. Ovo se može objasniti činjenicom da, uzimajući u obzir je ovo grafik za  $k = 1$ , se blago povećava broj učesnika na aukciji sa povećanjem gustine mreže što dovodi do blagog povećanja prosečnog broja poruka.



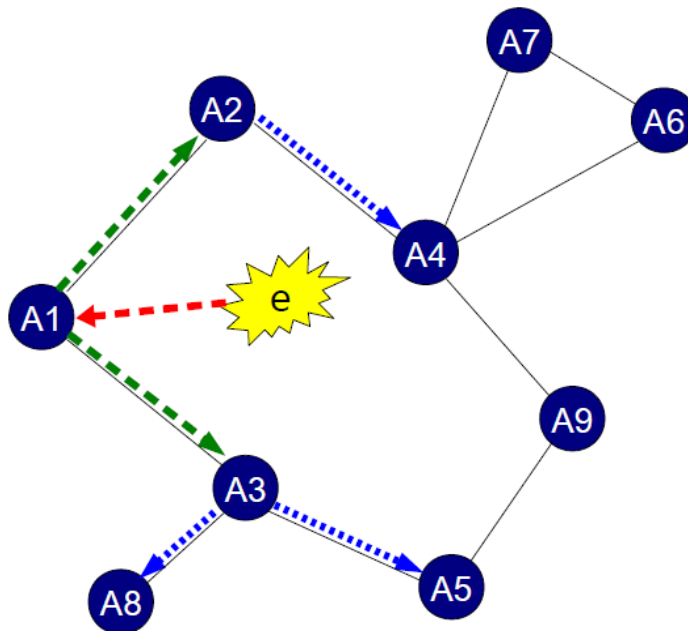
Slika 5.1.4 Poređenje komunikacionih troškova SAP i  $k$ -SAP algoritma u funkciji  $k$



Slika 5.1.5 Poređenje komunikacionih troškova SAP i  $k$ -SAP algoritma zavisno od gustine mreže

## 5.2 Proširenje lokalizovanog aukcijskog algoritma pohlepnim (*greedy*) algoritmom (*k*-SAPG)

Uzimajući u obzir činjenicu da se upotrebom lokalizovanog aukcijskog algoritma dobija značajno smanjenje komunikacionih troškova, ali uz smanjen procenat odabira najboljeg izvršioca, u ovoj disertaciji smo odabrali sledeću ideju radi poboljšanja procenta odabranih najboljih izvršilaca. Da bi poboljšali *k*-SAP dodali smo sledeće poboljšanje na bazi tzv. pohlepnog (eng. *greedy*) algoritma. Nakon što je *k*-SAP algoritam završen i odluka o (potencijalno) najboljem izvršiocu donešena, taj izvršilac pokreće novu aukciju na bazi *l*-SAP algoritma sa ciljem da potraži eventualno boljeg izvršioca među susednim izvršiocima koji su unutar njegovog komunikacionog radijusa. Ukoliko nema boljeg izvršioca, tada ne dolazi do preraspodele posla i on ostaje dodeljen prvobitnom izvršiocu. U slučaju da je pronađen bolji izvršilac, tada se posao njemu prosleđuje i on ponovo inicira *l*-SAP algoritam. Algoritam (označen sa *k*-SAPG) se ponavlja sve dok ima boljih izvršilaca među susedima unutar komunikacionog dometa onog koji inicira *l*-SAP aukciju.



Slika 5.2.1 Unapređenje *k*-SAP algoritma na bazi tzv. pohlepnog algoritma

Da bismo ilustrovali *k*-SAPG algoritam, pretpostavimo da je prvo upotrebljen



*I*-SAP algoritam da se odredi potencijalno najbolji aktuator za scenario prikazan na Slici 5.2.1. Aktuator A1 dobija informaciju o događaju  $e$  i inicira *I*-SAP aukciju na kojoj učestvuju aktuatori A2 i A3 koji su na jednakoj udaljenosti od događaja. Ukoliko je kriterijum odlučivanja udaljenost od događaja, on odlučuje da je A2 pobednik (ili A3 ali ne i oba) obzirom da su i A2 i A3 bliži događaju od A1. Međutim postoji bliži aktuator (A4) kojeg je moguće pronaći primenom gore predloženog proširenja algoritma na bazi tzv. pohlepnog algoritma ukoliko novu aukciju pokrene aktuator A2. Nakon nove lokalizovane *I*-SAP aukcije pronađen je bliži aktuator A4. Obzirom da A4 ne zna da je najbliži i on pokreće *I*-SAP aukciju i ne pronalazi bližeg, čime se algoritam završava i određen je aktuator A4 kao izvršilac. U slučaju da je aktuator A1 odlučio da je A3 onaj koji je pobednik (npr. iako su i A2 i A3 jednako udaljeni, ali A3 ima više energije) tada bi se algoritam završio iniciranjem *I*-SAP aukcije od strane auktuatora A3 jer on ne bi pronašao bližeg. Ovo je na Slici 5.2.1. ilustrovano tačkastim linijama.

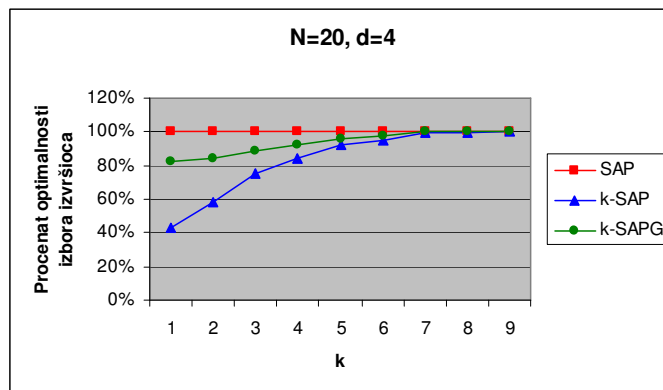
Algoritam *k*-SAPG moguće je formalizovati kao:

- 
- 
1. Pronađi (potencijalno) najboljeg izvršioca  $I_{PN}$  koristeći *k*-SAP algoritam
  2. Pokreni novu *I*-SAP aukciju gde izvršilac  $I_{PN}$  postaje kolektor
  3. Pronađi novog potencijalno najboljeg izvršioca  $I_{PN}$
  4. Ponovi korake 2 i 3 sve dok ima boljih izvršilaca  $I_{PN}$  od prethodnog
  5. Pobednik je poslednji  $I_{PN}$
- 
- 

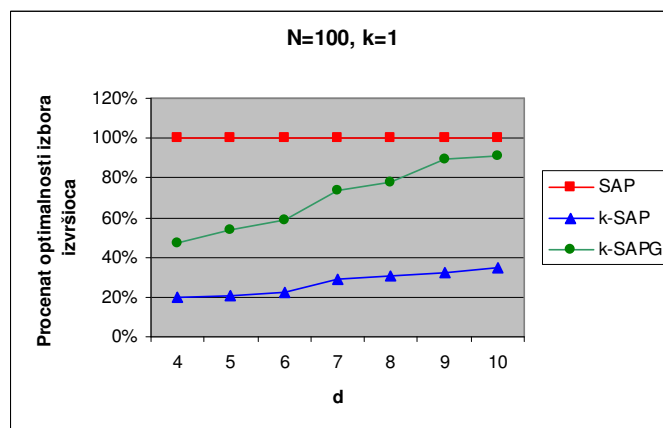
Ovaj algoritam nije moguće, a ni potrebno, primeniti za SAP algoritam jer tada učestvuju svi izvršioci na aukciji i najbolji izvršilac će sigurno biti pronađen.

Analizom rezultata i računanjem prosečnog procenta optimalnosti izbora izvršioca primenom *k*-SAPG, može se uočiti sledeće. Najinteresantniji za analizu, sa stanovišta najmanjeg dodatnog komunikacionog opterećenja, je slučaj za  $k = 1$ . Uz  $d = 4, 5, \dots, 10$  i  $N = 20, 50$  i  $100$ , procenat optimalnog izbora izvršioca se kreće u opsegu od 47 do 98 procenata (videti Sliku 5.2.3). Jedino za male ( $N \leq 10$ ) i guste mreže ili za veće vrednosti  $k$  moguće je dostići 100 procenata (videti Sliku 5.2.2). Uvođenje

proširenja lokalizovanog aukcijskog algoritma na bazi tzv. pohlepnog algoritma pokazuje poboljšanje optimalnosti pri izboru izvršioca koja se kreće do 2,78 puta bolje u odnosu na  $k$ -SAP.



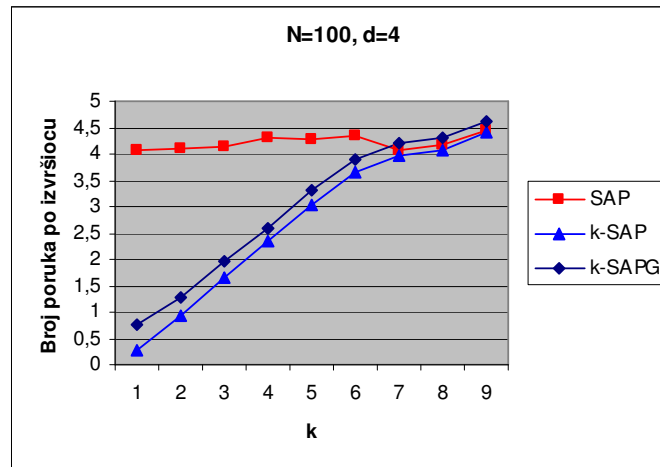
Slika 5.2.2 Poređenje procenta optimalnog odabira izvršioca za SAP,  $k$ -SAP i  $k$ -SAPG u funkciji od  $k$



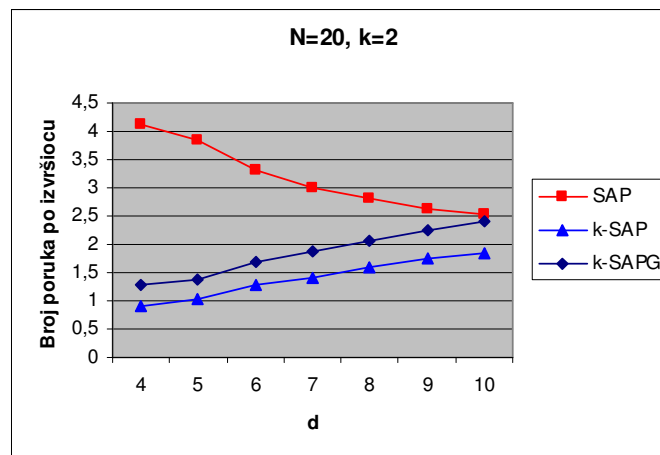
Slika 5.2.3 Poređenje procenta optimalnog odabira izvršioca za SAP,  $k$ -SAP i  $k$ -SAPG u funkciji od gustine mreže

Osnovni cilj  $k$ -SAPG algoritma je povećanje procenta optimalnih odabira izvršilaca. Uvođenje proširenja na bazi tzv. pohlepnog algoritma uzrokuje povećanje komunikacionih troškova. Analizom rezultata uočavaju se sledeće karakteristike algoritma u smislu povećanja komunikacionih troškova. Merena veličina koja daje informaciju o tome je prosečan broj poruka po izvršiocu koji je potreban da se donese odluka o tome koji je najbolji. Izmereno prosečno komunikaciono opterećenje je oko 0.4 poruke po izvršiocu (stvarne vrednosti su se kretale od 0.07 do 1.01). Pri tome,

najbolji rezultat dobija se za velike mreže (ali ne i guste) uz male vrednosti  $k$ . Posebno dobra karakteristika ovog algoritma je što u slučaju većih mreža ( $N > 10$ ) ima približno konstantno dodatno komunikaciono opterećenje u odnosu na  $k$ -SAP algoritam. Ovo je ilustrovano graficima na Slikama 5.2.4 i 5.2.5.



Slika 5.2.4 Poređenje prosečnog broja poruka po izvršiocu za SAP,  $k$ -SAP i  $k$ -SAPG u funkciji od  $k$



Slika 5.2.5 Poređenje prosečnog broja poruka po izvršiocu za SAP,  $k$ -SAP i  $k$ -SAPG u funkciji od gustine mreže

Kao mera kvaliteta odlučivanja o izboru izvršioca korišćen je prosečan odnos udaljenja između izabranog izvršioca i događaja i udaljenja onog koji je stvarno najbliži događaju. Ova mera je označena sa SCD.

$$SCD = d(I_I)/d(I_N) \quad (5.2.1)$$

pri tome je:

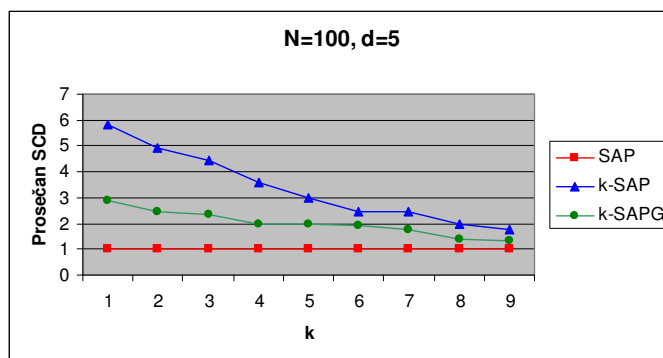
$I_I$  – izabrani izvršilac

$I_N$  – najbolji (najbliži) izvršilac

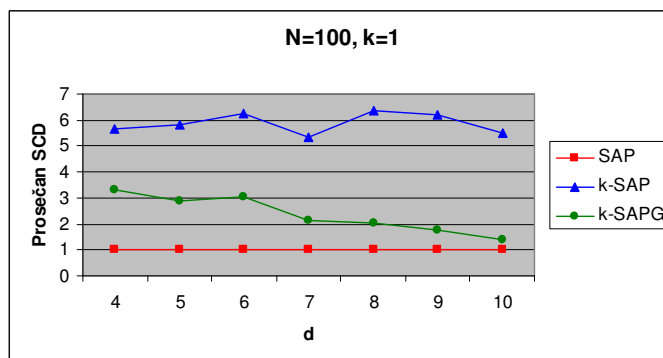
$d(I_x)$  – rastojanje izvršioca  $x$  do događaja

SAP algoritam ima konstanto jediničnu vrednost ovog parametra jer je uvek izabrani izvršilac ujedno i najbolji. Mera kvaliteta drugih algoritama se ogleda u tome koliko su za pojedine scenarije bliski jedinici.

Opravednost uvođenja proširenja na bazi tzv. pohlepnog algoritma se najbolje vidi u slučaju velikih mreža i za  $k = 1$  jer se dobija smanjenje prosečnog SCD u slučaju gustih mreža i do četiri puta. Pri tome je dodatno komunikaciono opterećenje najmanje jer je  $k = 1$ . Naredna dva grafika ilustruju rezultate dobijene za scenario sa  $N = 100$  izvršilaca.



Slika 5.2.6 Prosečan odnos udaljenosti izabranog i najbližeg izvršioca u funkciji od  $k$  za SAP,  $k$ -SAP i  $k$ -SAPG



Slika 5.2.7 Prosečan odnos udaljenosti izabranog i najbližeg izvršioca u funkciji od gustine mreže za SAP,  $k$ -SAP i  $k$ -SAPG

Većina rezultata vezanih za uticaj tzv. pohlepnog algoritma, koji su prikazani u ovom odeljku, publikovana je u članku [MMS12].

## 6 Aukcijski agregacioni algoritmi

O aukcijama kao alatu koji se uveliko koristi u raznim istraživačkim oblastima u računarskim naukama, pa i u oblasti bežičnih senzorskih i aktuatorskih mreža, a koje nemaju direktne veze sa ekonomijom bilo je reči u odeljku 1.6 uvodnog poglavlja. Kao unapređenje osnovne ideje primene aukcija ovde uvodimo i tehniku agregacije koja se u oblasti bežičnih senzorskih mreža često koristi za smanjenje količine poruka koje se šalju prilikom slanja podataka prema sink čvoru.

Pojam agregacija potiče od latinskog *aggregacio* i znači pridruživanje, spajanje u jedno. Pojam agregacije paketa se definiše u komunikacionim mrežama kao proces spajanja više paketa u jedan pre njegovog daljeg slanja radi smanjenja komunikacionih troškova [W1]. Ovo je definicija koja je najbliža suštini agregacije u algoritmima koji će biti prikazani u ovom poglavlju.

Radi štednje energije u bežičnim senzorskim i aktuatorskim mrežama, što je jedan od osnovnih ciljeva zbog najčešće baterijskog napajanja čvorova, jedan od načina uštede energije je i smanjenje komunikacionih troškova. Ovo je osnovni cilj pet novih aukcijskih agregacionih algoritama za izbor izvršioca koji su prikazani u ovom poglavlju:

- 1) Prosti aukcijski agregacioni algoritam (SAAP)
- 2) Lokalizovani prosti aukcijski agregacioni algoritam ( $k$ -SAAP)
- 3) Proširenje lokalizovanog aukcijskog agregacionog algoritma tzv. pohlepnim (eng. *greedy*) algoritmom ( $k$ -SAAPG)
- 4) Aukcijski agregacioni algoritam (AAP)
- 5) Lokalizovani aukcijski agregacioni algoritam ( $k$ -AAP)

Kod svih algoritama se podrazumeva da je neki od izvršilaca – kolektor dobio informaciju o tome da se desio događaj i on inicira neki od pet gore navedenih

algoritama za izbor najboljeg izvršioca. Proces informisanja nekog od izvršilaca naziva se otkrivanje servisa (*service discovery*) ili otkrivanje lokacije izvršioca (*location service*) i složen je problem u opštem slučaju. Pored toga, očigledno je da i dalji rezultati algoritama koji određuju koji je izvršilac najbolji da reaguje veoma zavise od toga koji je izvršilac odabran da bude kolektor. Da rezultati ne bi zavisili od konkretnog algoritma za otkrivanje lokacije izvršioca, opredelili smo se za probablističku metodu određivanja izvršioca – kolektora.

Kolektor se na bazi probablističke metode određuje prema jednačinama (6.1) i (6.2) na sledeći način.

$$P(I_i) = 1/(D * d_i) \quad (6.1)$$

$$D = 1/d_1 + 1/d_2 + \dots + 1/d_N \quad (6.2)$$

gde je:

$P(I_i)$  – verovatnoća izbora izvršioca  $I_i$  kao kolektora

$d_i$  – udaljenje od izvršioca  $I_i$  do događaja

$N$  – broj izvršilaca

Pokazuje se da ukoliko se koristi ovakav način odabira kolektora, on je ujedno i najbliži događaju prosečno u 15% slučajeva. U Tabeli 6.1 prikazana je prosečna vrednost izbora kolektora koji je istovremeno i najbliži izvršilac (označeno sa CC) u zavisnosti od veličine mreže. Ovo su niske vrednosti ukoliko bi se koristio ovakav metod za otkrivanje lokacije izvršioca, ali omogućava dobru osnovu za proveru karakteristika aukcijskih agregacionih algoritama.

Tabela 6.1 Procenat izbora najbližeg izvršioca kao kolektora probablističkom metodom

<b>N</b>	10	20	50	100
<b>CC[%]</b>	24	17	11	8

## 6.1 Prosti aukcijski agregacioni algoritam (SAAP)

Informaciona faza nakon detekcije nekog događaja je kod prostog aukcijskog

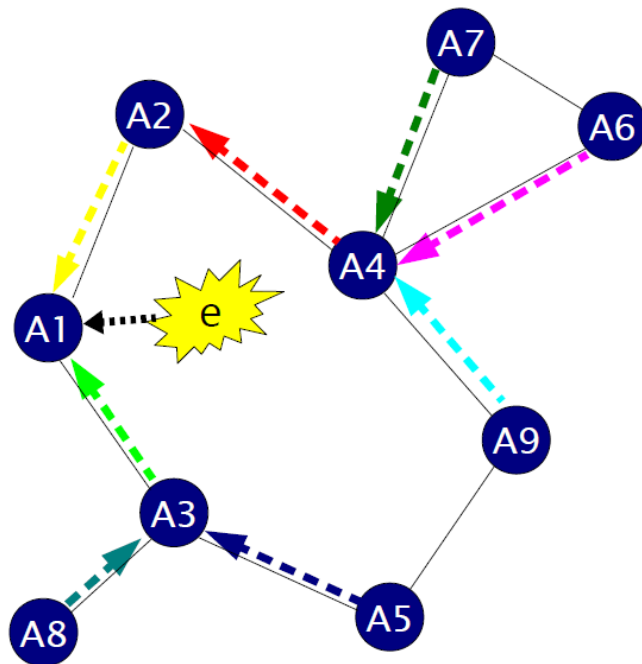
agregacionog algoritma (SAAP) ista kao i kod prostog aukcijskog algoritma (SAP). Informaciju o događaju prosleđuje u mrežu izvršilac kojeg ovde zovemo glavni kolektor. Dalje prosleđivanje poruka vrši se na način da svaki izvršilac kada dobije informacionu poruku, prosleđuje je samo jedanput čime se jasno određuje koji čvor je u stablu na poziciji roditelja, a koji čvor je naslednik – dete. Tokom informacione faze svaki izvršilac uključuje u poruku i svoj identifikacioni broj tako da svaki izvršilac istovremeno dobija i informaciju od kog roditelja je dobio informacionu poruku. Obzirom da je moguće da dobije poruku od više izvršilaca, on bira samo jednog za roditelja i njemu će slati odgovor nazad u aukcijskoj fazi. Informaciona faza se još naziva i faza širenja komunikacionog stabla.

Ono što razlikuje ovaj algoritam od SAP-a je način kako se poruke vraćaju nazad prema glavnom kolektoru u aukcijskoj fazi. Ovu fazu nazivamo i faza sažimanja jer se ovde poruke ne šalju zasebnim putanjama već na sledeći način uvođenjem lokalne agregacije pomoću malih lokalnih *1*-hop aukcija. Dakle, u aukcijskoj fazi svaki izvršilac vraća poruku svome roditelju sa cenom koliko bi koštalo da on reaguje na dati događaj. Taj roditelj postaje lokalni kolektor i čeka da mu odgovore svi izvršiooci koji su njega označili kao roditelja. Kada dobije sve poruke on lokalno odlučuje koji je izvršilac poslao najbolju ponudu i potom šalje samo jednu poruku ka svome roditelju sa najboljom ponudom. Na ovaj način smo dobili efekat agregacije tj. sažimanja. Ovo je ilustrovano na Slici 6.1.1 za scenario sa devet aktuatora. Glavni aktuator je A1, a lokalni kolektori su aktuatori A2, A3 i A4.

Krajnji cilj ovakve aukcijske faze je smanjenje komunikacionih troškova jer se šalje manje poruka. Broj potrebnih poruka za aukcijsku fazu je jednak broju izvršilaca (broju povratnih linija različitih boja na Slici 6.1.1) tj. ukupno devet za ovaj scenario. Za isti taj scenario, ali korišćenjem SAP algoritma potrebno je ukupno šesnaest poruka jer svaki aktuator odgovara glavnom aktuatoru zasebnim putanjama.

Optimalnost odlučivanja je ista za SAP i SAAP algoritme jer na aukciji učestvuju isti izvršiooci. Mana ovog algoritma je produženo vreme odlučivanja. Ovim problemom se nismo bavili obzirom da je u okviru ove disertacije istraživan samo slučaj odlučivanja nakon pojave jednostrukog događaja gde vreme odlučivanja ima ulogu samo ukoliko se radi o urgentnim događajima.



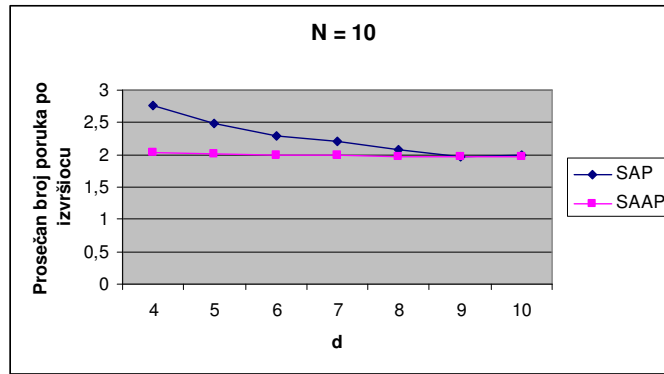


Slika 6.1.1 Ilustracija aukcijske agregacije kod SAAP algoritma

SAAP algoritam moguće je formalizovati na sledeći način:

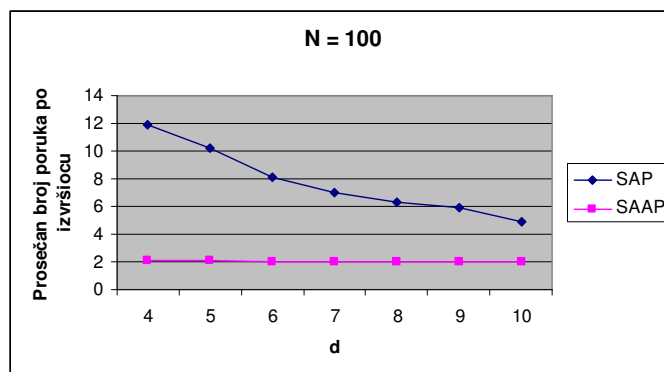
- 
- 
1. Glavni kolektor šalje informacione poruke svim izvršiocima u mreži.
  2. Svaki izvršilac kada poruku primi prvi put šalje je dalje samo jedanput.
  3. U aukcijskoj fazi svaki roditelj u grafu postaje lokalni kolektor, čeka licitaciju svakog svog deteta i lokalno odlučuje čija je ponuda najbolja. Potom i sam licitira prema svom roditelju.
  4. Kada glavni kolektor dobije licitacije od svakog svog deteta u stablu on odlučuje koja je ponuda najbolja.
  5. Glavni kolektor obaveštava izvršioca pobednika na aukciji slanjem odgovarajuće poruke i dodeljuje mu posao.
- 
- 

Analizom rezultata mogu se izvesti sledeći zaključci. SAP i SAAP se ponašaju skoro identično za male i guste mreže i tu ovaj algoritam ne donosi neko poboljšanje, a unosi kašnjenje u odlučivanju pa u tim slučajevima nije pogodan za korišćenje. Ovo je ilustrovano grafikom na Slici 6.1.2 za vrednosti  $d > 8$ .



Slika 6.1.2 Za guste i male mreže SAAP ne donosi skoro nikakvo poboljšanje

Kolika je međutim prednost SAAP u odnosu na SAP za slučaj velikih i retkih mreža se vidi na Slici 6.1.3. Ovde imamo scenario sa  $N = 100$  izvršioca i mrežu koja ima prosečno od  $d = 4$  do  $10$  suseda. Za  $d = 4$ , vidi se da je prosečan broj potrebnih poruka po izvršiocu manji za čak šest puta. Takođe varijacije komunikacionog opterećenja za SAAP su vrlo male tako da se uz minimalnu grešku može reći da je konstantno i iznosi dve poruke po izvršiocu. Ovo se može jednostavno objasniti i činjenicom da kod SAAP algoritma svaki izvršilac šalje u informacionoj fazi poruku jedanput, a u aukcijskoj takođe po jednom vraća poruku kada licitira svome roditelju što ukupno daje dve poruke po izvršiocu. Ovo se može videti i na rezultatima simulacija na Slici 6.1.2 i Slici 6.1.3. Ova osobina SAAP algoritma može biti veoma važna za centralizovane sisteme odlučivanja gde želimo da imamo ravnomerno komunikaciono opterećenje u celoj mreži.



Slika 6.1.3 Za velike, a posebno velike i retke mreže SAAP donosi značajno smanjenje komunikacionih troškova

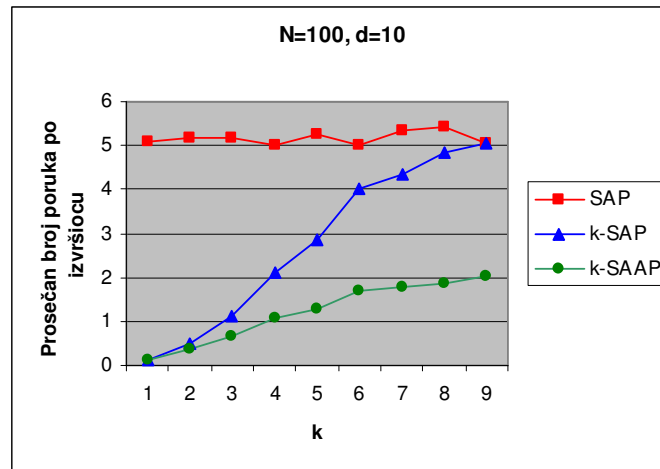
## 6.2 Lokalizovani prosti aukcijski agregacioni algoritam ( $k$ -SAAP)

Uvođenje agregacije dovodi do smanjenja komunikacionih troškova, ali oni mogu biti ipak visoki posebno za velike mreže. Da bi se dalje smanjili troškovi u situacijama kada nije neophodno da imamo izbor najboljeg izvršioca u *svim* slučajevima, može se koristiti lokalizovana varijanta SAAP algoritma (označena sa  $k$ -SAAP). To znači da glavni kolektor u informacionoj fazi šalje poruke na takav način da one stignu samo do susednih izvršilaca koji su do  $k$  hopova udaljeni. Da bi to bilo moguće, u informacionim porukama se uvodi dodatna promenljiva `lokalno_k` koja služi za praćenje broja hopova i svakim narednim slanjem se smanjuje za jedan.

$k$ -SAAP algoritam moguće je formalizovati kao:

- 
- 
1. Glavni kolektor šalje informacione poruke svim susednim izvršiocima uz `lokalno_k = k`.
  2. Svaki izvršilac kada poruku primi prvi put šalje je dalje jedanput uz `lokalno_k = lokalno_k - 1`; Ukoliko je `lokalno_k = 0` poruka se ne šalje dalje.
  3. U aukcijskoj fazi svaki roditelj u grafu postaje lokalni kolektor, čeka licitaciju svakog svog deteta i lokalno odlučuje čija je ponuda najbolja. Potom i sam licitira prema svom roditelju.
  4. Kada glavni kolektor dobije licitacije od svakog svog deteta u stablu on odlučuje koja je ponuda najbolja.
  5. Glavni kolektor obaveštava pobednika na aukciji slanjem odgovarajuće poruke i dodeljuje mu posao.
- 
- 

Koliki je doprinos smanjenju komunikacionog opterećenja lokalizovanog aukcijskog agregacionog protokola u odnosu na prosti i lokalizovani prosti aukcijski protokol vidi se na Slici 6.2.1. U slučajevima velikih i gustih mreža  $k$ -SAAP ima znatno manju komunikacionu potrošnju od  $k$ -SAP uz istu optimalnost odlučivanja. Za male mreže, slično kao i SAAP, ni  $k$ -SAAP ne donosi neko poboljšanje koje bi moglo opravdati povećanje vremena odlučivanja zbog lokalnih aukcija.



Slika 6.2.1 poređenje komunikacionih troškova SAP,  $k$ -SAP i  $k$ -SAAP algoritama

Za komunikaciono opterećenje izraženo prosečnim brojem poruka po izvršiocu moguće je dokazati sledeću teoremu.

**TEOREMA 1** (gornja granica)

Za lokalizovani  $k$ -SAAP algoritam uvek važi da je prosečan broj poruka po izvršiocu manji ili jednak 2.

**Dokaz:**

Pretpostavimo da imamo  $N$  izvršilaca i da je  $k = k_{max}$  odabrano dovoljno veliko tako da se u najviše  $k_{max}$  hopova može stići od bilo kojeg čvora do bilo kojeg drugog čvora u mreži. Ovim je obuhvaćena cela mreža i algoritam se svodi na nelokalizovani SAAP algoritam opisan u prethodnom odeljku. Tada svaki izvršilac šalje po jednu poruku u informacionoj fazi prilikom obaveštavanja ostalih i po jednu u aukcijskoj prilikom licitiranja, tako da se dobija prosečno po dve poruke po izvršiocu. Slučaj gde je  $k > k_{max}$  se svodi na slučaj  $k = k_{max}$ . Za svako  $k$  za koje važi  $k < k_{max}$  jedan deo izvršilaca ne učestvuje u komunikaciji i samim tim je broj poruka manji od slučaja za  $k = k_{max}$ . Iz svega navedenog sledi da je prosečan broj poruka po izvršiocu za  $k$ -SAAP algoritam manji ili jednak 2.

## TEOREMA 2 (donja granica)

Za lokalizovani  $k$ -SAAP algoritam uvek važi da je prosečan broj poruka po izvršiocu veći ili jednak od  $2k/N$ , uz  $k < N$  da bi algoritam bio lokalizovan.

### **Dokaz:**

Prvo ćemo dokazati da uvek važi da je  $2k/N < 2$  (što je tvrdnja pokazana u prethodnoj teoremi). Uzimajući u obzir da je  $k < N$ , tada je  $k/N < 1$ , sledi da je i  $2k/N < 2$ .

U informacionoj fazi je potrebno najmanje  $k$  poruka da bi svi bili informisani jer svaki izvršilac prosleđuje poruku samo po jednom kada je primi. U aukcijskoj fazi je potrebno najmanje poruka za slučaj da svako dete ima samo po jednog roditelja. Tada je potrebno još  $k$  poruka da bi svi izvršioci odgovorili svakom svom roditelju i oni nazad prema kolektoru. Sabiranjem broja potrebnih poruka za obe faze se dobija da je to  $2k/N$  što predstavlja donju granicu za broj poruka.

Na osnovu prethodne dve teoreme moguće je tačno znati u kojim granicama se kreće komunikaciono opterećenje  $k$ -SAAP algoritma što je vrlo važno prilikom računanja energetske efikasnosti algoritma. Pored toga ove informacije kvalifikuju algoritam za poboljšanje nekog već postojećeg algoritma u scenarijima gde je važno da se unapred zna dodatno komunikaciono opterećenje. Ovo će biti pokazano u sedmom poglavlju.

### **6.3 Proširenje lokalizovanog prostog aukcijskog agregacionog algoritma pohlepnim (*greedy*) algoritmom ( $k$ -SAAPG)**

Uzimajući u obzir činjenicu da se upotrebom lokalizovanog prostog aukcijskog agregacionog algoritma dobija značajno smanjenje komunikacionih troškova, ali uz smanjen procenat odabira najboljeg izvršioca, i za ovaj algoritam smo odabrali proširenje na bazi tzv. pohlepnog (*greedy*) algoritma sa ciljem poboljšanja procenta odabranih najboljih izvršilaca. Nakon što je osnovni  $k$ -SAAP algoritam završen i odluka o (potencijalno) najboljem izvršiocu donešena, taj izvršilac pokreće novu aukciju na

bazi  $I$ -SAP algoritma sa ciljem da potraži eventualno boljeg izvršioca među susednim izvršiocima koji su unutar njegovog komunikacionog radijusa. Ukoliko nema boljeg izvršioca, tada ne dolazi do preraspodele posla i on ostaje dodeljen prvobitnom izvršiocu. U slučaju da je pronađen bolji izvršilac, tada se posao njemu prosleđuje i on ponovo inicira  $I$ -SAP algoritam. Ovaj algoritam (označen sa  $k$ -SAAPG) se ponavlja sve dok ima boljih izvršilaca među susedima unutar komunikacionog dometa onog koji inicira  $I$ -SAP aukciju. Ilustracija primene proširenja na bazi pohlepnog algoritma je ilustrovana na Slici 5.2.1 u prethodnom poglavlju.

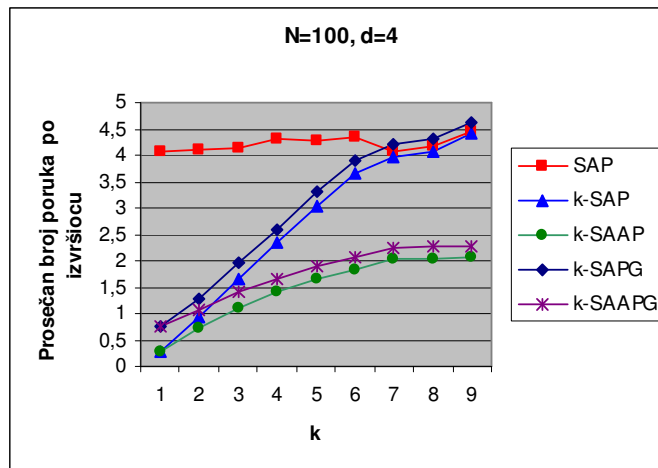
Algoritam  $k$ -SAAPG moguće je formalizovati kao:

- 
- 
1. Pronađi (potencijalno) najboljeg izvršioca  $I_{PN}$  koristeći  $k$ -SAAP algoritam
  2. Pokreni novu  $I$ -SAP aukciju gde izvršilac  $I_{PN}$  postaje kolektor
  3. Pronađi novog potencijalno najboljeg izvršioca  $I_{PN}$
  4. Ponovi korake 2 i 3 sve dok ima boljih izvršioca  $I_{PN}$  od prethodnog
  5. Pobednik je poslednji  $I_{PN}$
- 
- 

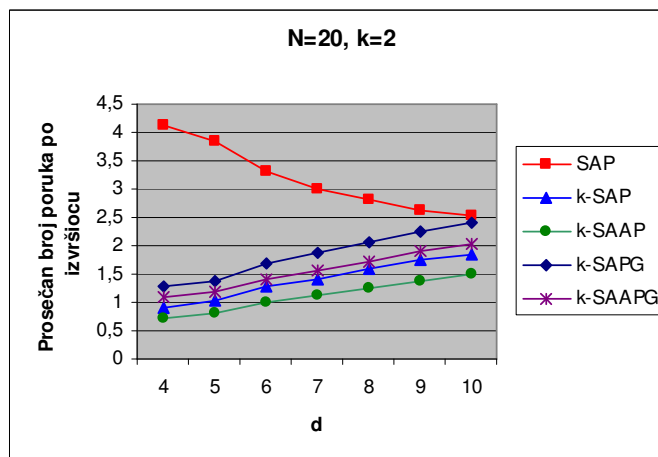
Ovaj algoritam nije moguće, a ni potrebno, primeniti za SAAP algoritam jer tada učestvuju svi izvršioci na aukciji i najbolji izvršilac će sigurno biti pronađen.

Sa stanovišta optimalnosti odabira izvršioca, ovaj algoritam se ponaša identično kao i algoritam  $k$ -SAPG, objašnjen u odeljku 5.2 poglavlja 5. To znači da se uvođenjem proširenja lokalizovanog aukcijskog agregacionog algoritma na bazi tzv. pohlepnog algoritma dobija poboljšanje optimalnosti pri izboru izvršioca koja se kreće do 2,78 puta bolje u odnosu na  $k$ -SAAP.

Osnovni cilj  $k$ -SAAPG algoritma je povećanje procenta optimalnih odabira izvršilaca. Uvođenje proširenja na bazi tzv. pohlepnog algoritma se vrši na isti način kao i u slučaju  $k$ -SAPG algoritma objašnjenog u odeljku 5.2 poglavlja 5. Samim tim je i izmereno dodatno prosečno komunikaciono opterećenje oko 0.4 poruke po izvršiocu (stvarne vrednosti su se kretale od 0.07 do 1.01) isto kao i za  $k$ -SAPG. Ukupno prosečno komunikaciono opterećenje svih pet do sada obrađenih algoritama je ilustrovano graficima na Slikama 6.3.1 i 6.3.2.



Slika 6.3.1 Poređenje prosečnog broja poruka po izvršiocu za svih pet do sada obrađenih algoritama u funkciji od  $k$



Slika 6.3.2 Poređenje prosečnog broja poruka po izvršiocu za svih pet do sada obrađenih algoritama u funkciji od gustine mreže

Sa slika se može videti da uvođenje pohlepnog proširenja dovodi do malog povećanja ukupnog komunikacionog opterećenja, ali uz značajno poboljšanje optimalnosti izbora izvršioca.

Kao mera kvaliteta odlučivanja o izboru izvršioca korišćen je prosečan odnos udaljenja između izabranog izvršioca i događaja i udaljenja onog koji je stvarno najbliži događaju. Ova mera (SCD) definisana je jednačinom (5.2.1) u odeljku 5.2 poglavlja 5. Ovaj algoritam se ponaša na isti način kao i  $k$ -SAPG, tako da rezultati u vezi sa prosečnim SCD prikazani tamo važe i ovde za  $k$ -SAAPG. Tako da opravdanost uvođenja proširenja na bazi tzv. pohlepnog algoritma se najbolje vidi u slučaju velikih mreža i za

$k = 1$  jer se dobija smanjenje prosečnog SCD u slučaju gustih mreža i do 4 puta. Pri tome je dodatno komunikaciono opterećenje najmanje jer je  $k = 1$ .

Većina rezultata vezanih za uticaj tzv. pohlepnog algoritma, koji su prikazani u ovom odeljku, publikovana je u članku [MMS12].

#### **6.4 Aukcijski agregacioni algoritam (AAP)**

Do sada prikazani algoritmi vršili su odlučivanje o tome koji je najbolji izvršilac da reaguje na događaj na bazi dobijenih licitacija od svih izvršilaca u susedstvu (SAP i SAAP), na bazi licitacija od komšija koji su udaljeni do  $k$  hopova ( $k$ -SAP i  $k$ -SAAP) i primenom tzv. pohlepnog algoritma kao proširenja  $k$ -SAP i  $k$ -SAAP algoritama ( $k$ -SAPG i  $k$ -SAAPG). Ovi algoritmi su u informacionoj fazi slali poruke celoj mreži (SAP i SAAP) ili komšijama udaljenim do  $k$  hopova ( $k$ -SAP,  $k$ -SAAP,  $k$ -SAPG i  $k$ -SAAPG). Prilikom prosleđivanja informacionih poruka usvojeno je pravilo da svaki izvršilac u mreži prosleđuje poruku kada je dobije samo jedanput i on nije imao nikakvo prethodno znanje o mreži.

Da bismo olakšali proces odlučivanja o najboljem izvršiocu koji treba da reaguje na neki događaj, kao i samu komunikaciju, kod aukcijskog agregacionog algoritma koristimo znanje o mreži koje je moguće dobiti u toku konfigurisanja mreže i tokom razmene poruka u toku rada mreže. Ideja prikupljanja predznanja na bazi topoloških informacija je u tome da svaki čvor u mreži ima informaciju o susednim čvorovima i da na osnovu te informacije može lokalno da odlučuje da li uopšte da šalje dalje poruke u informacionoj fazi. Odluku vrši na osnovu procene o tome da li postoji bolji izvršilac na nekom udaljenju od sopstvenog. Tu se javlja mogućnost smanjenja potrebnih poruka u informacionoj fazi.

Algoritam predstavljen u ovom odeljku pretpostavlja da svaki čvor ima informaciju o lokaciji i radnom stanju svakog čvora u mreži. Čak i inicijalno potrebna razmena poruka da bi ovakva pretpostavka bila ispunjena stvara ogromno komunikaciono opterećenje, a i održavanje ove pretpostavke u toku rada mreže bi takođe davalo značajno komunikaciono opterećenje, posebno za veće mreže, pa ovaj algoritam nije svrsishodan. Uprkos tome u literaturi se često pojavljuju članci, posebno



u oblasti robotike, u kojima se ova pretpostavka implicitno postavlja time što se podrazumeva da svaki čvor može da komunicira sa svakim čvorom čime se dobija kompletan graf mreže. Naravno u scenarijima koji teže da barem liče na praktične, ovakva pretpostavka nije održiva. Međutim lokalizovana varijanta ovog algoritma jeste svrsishodna i to će biti pokazano u narednom odeljku.

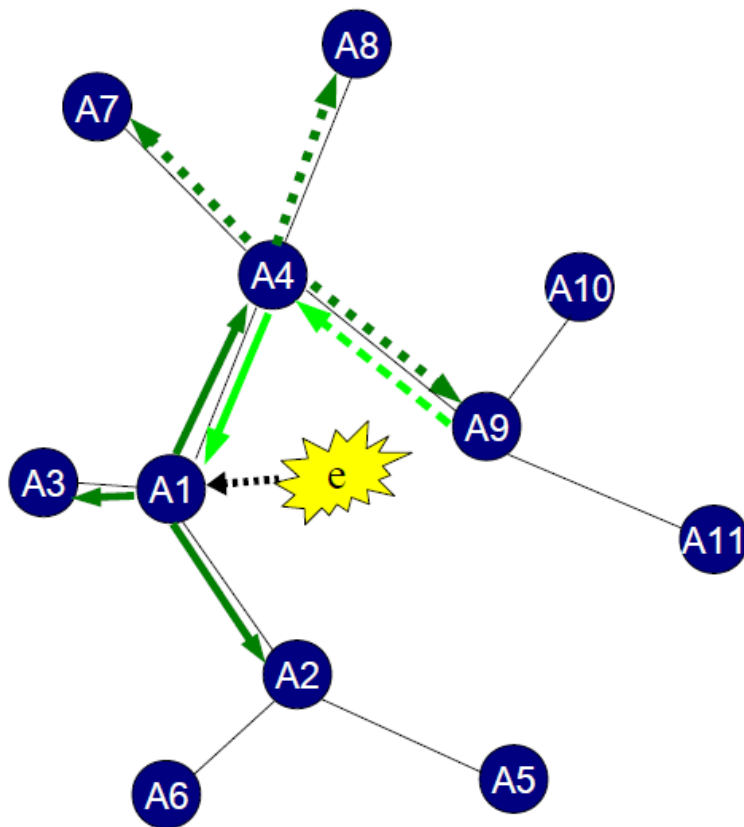
### **6.5 Lokalizovani aukcijski agregacioni algoritam ( $k$ -AAP)**

Usvajanjem lokalnog znanja o susedima u toku konfigurisanja mreže i potom u toku rada mreže dodavanjem topoloških informacija na poruke koje se razmenjuju, dobijamo osnovni preduslov za lokalizovani aukcijski agregacioni algoritam ( $k$ -AAP). Osnovna ideja da smanjujemo broj poruka u toku informacione faze lokalnim odlučivanjem o tome da li da se informaciona poruka šalje dalje ili ne, na osnovu predznanja o susednim čvorovima, je potpuno drugačija u odnosu na dosadašnje ideje pokazane u prethodnim algoritmima gde se smanjenje komunikacionih troškova realizuje u aukcijskoj fazi algoritama.

Na ovaj način dobijamo seriju lokalizovanih algoritama za svako različito celobrojno  $k$ . U svakom algoritmu kolektor započinje aukciju i u trivijalnom slučaju 0-AAP algoritma, on je jedini učesnik aukcije i on odmah prelazi na izvršenje zadatka.

U slučaju 1-AAP algoritma, kolektor (kao i ostali izvršioci u mreži) ima znanje o svim prvim komšijama i tada on može da proceni da li među prvim komšijama ima neki koji bi mogao biti bolji od njega samog i šalje im informacionu poruku sa priključenom informacijom o svojoj ceni reagovanja na dati događaj. U scenariju na Slici 6.5.1 izvršilac A1 je kolektor i on ima informaciju da ni jedan od izvršilaca među prvim susedima nije bolji od njega samog tako da on ne šalje dalje ni jednu informacionu poruku, već dodeljuje samom sebi posao. Ovde vidimo da bi se primenom proširenja na bazi tzv. pohlepnog algoritma moglo doći do najboljeg izvršioca (to je A9 jer je najbliži događaju), međutim ovde to nije potrebno jer već primenom 2-AAP algoritma dobijamo isti rezultat.

Naime, kod 2-AAP algoritma kolektor A1 ima informaciju da među drugim komšijama ima neki koji je bolji i zato šalje poruku sa svojom cenom prvim komšijama (to su A2, A3 i A4). Izvršioci A2 i A3 znaju da nemaju komšija sa boljom cenom, a i



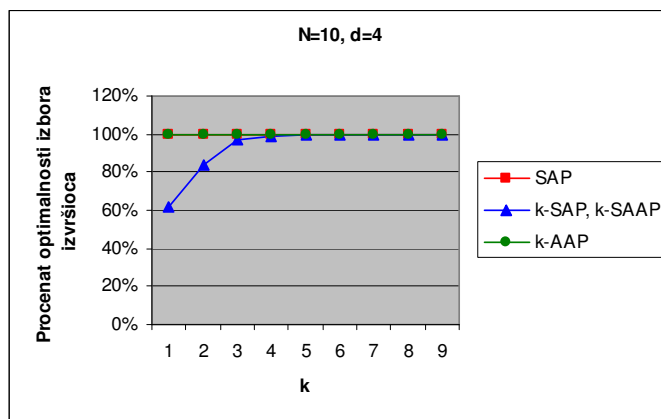
Slika 6.5.1 Ilustracija 2-AAP algoritma

sami imaju lošiju cenu od A1 tako da oni lokalno odlučuju da ne šalju dalje informaciju, a ne šalju ni odgovor nazad. Izvršilac A4 zna da među svojim prvim komšijama (to su A7, A8 i A9) ima neki koji je bolji i šalje im poruku sa do sada najboljom cenom. Pre slanja odgovora sa svojom licitacijom, svaki izvršilac procenjuje da li među njegovim prvim i drugim komšijama možda ima neki koji je bolji od do sada ukupno najbolje cene. Ukoliko nema boljeg (kao što je slučaj u scenariju na Slici 6.5.1) oni koji imaju lošiju cenu od do sada dobijene lokalno odlučuju da ne šalju odgovore (to su A7 i A8), a oni koji imaju bolju cenu odmah šalju odgovore nazad (to je A9). Ukoliko među drugim komšijama ima boljih čekaju na odgovor od svojih prvih komšija pa tek onda oni šalju odgovor nazad prema kolektoru. Nakon dobijanja svih odgovora kolektor odlučuje koji je najbolji i njemu šalje poruku o dodeli posla (to je A9 u scenariju na Slici 6.5.1).

Za ovako definisan aukcijski agregacioni algoritam, slično kao i kod prostog aukcijskog agregacionog algoritma, imamo fazu širenja (informaciona faza) i fazu sažimanja (aukcijaska faza), ali sa najčešće znatno manjim ili u najgorem slučaju istim

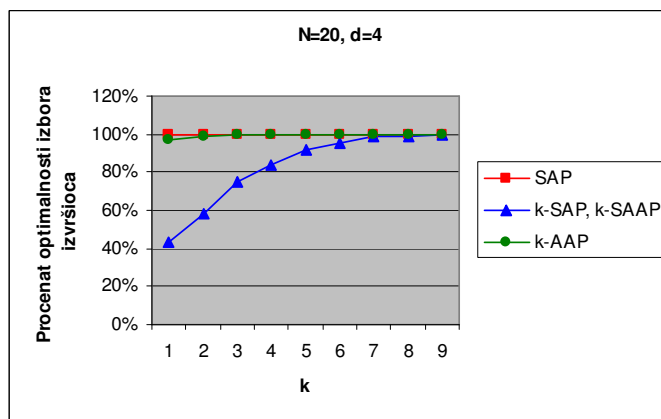
komunikacionim opterećenjem. Značajno smanjenje komunikacionog opterećenja se dobija kao posledica distribuiranja odlučivanja među lokalnim izvršiocima i to je pokazano analizom rezultata u nastavku.

Za male i retke mreže  $k$ -AAP algoritam postiže 100% optimalnost pronalaženja najboljeg već za  $k = 1$ , dok  $k$ -SAP i  $k$ -SAAP to postižu tek za  $k = 5$  što je pokazano na Slici 6.5.2.



Slika 6.5.2  $k$ -AAP daje 100% optimalnost izbora izvršioca za male i retke mreže

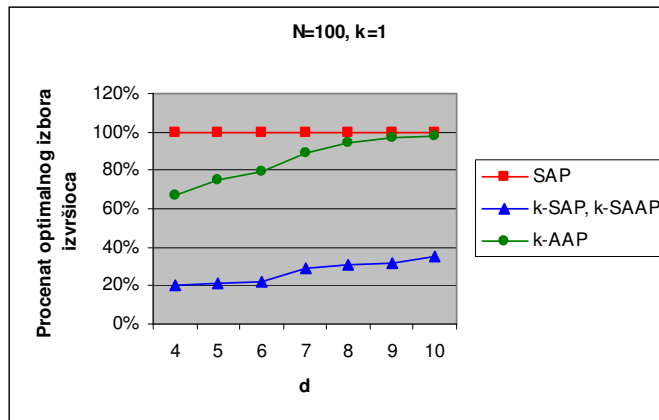
Na Slici 6.5.3 su dati rezultati za scenario sa  $N = 20$  izvršilaca i retku mrežu ( $d = 4$ ). Vidi se da se 100% optimalnost pronalaženja najboljeg izvršioca dobija sa  $k$ -AAP algoritmom za  $k = 3$ , za razliku od  $k$ -SAP i  $k$ -SAAP gde je za to potrebno  $k = 9$ .



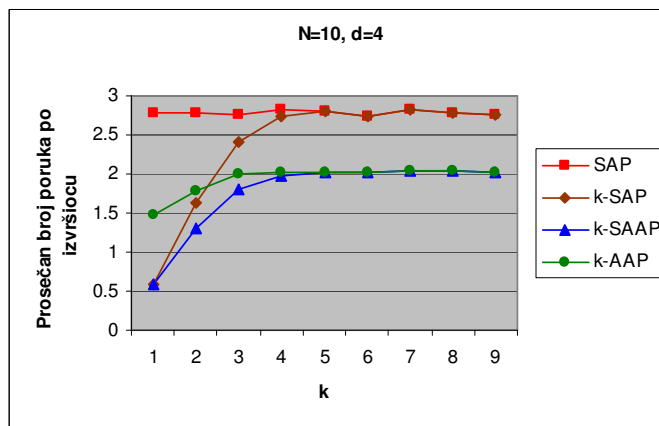
Slika 6.5.3 Ilustracija poboljšanja optimalnosti izbora izvršioca koja se dobija sa  $k$ -AAP algoritmom

Uticaj gustine mreže na procenat optimalnosti izbora izvršioca je prikazan na Slici 6.5.4 za scenario sa  $N = 100$  izvršilaca i za najmanje  $k = 1$ . Sa grafika se vidi da se

korišćenjem  $k$ -AAP algoritma može postići 100% optimalnost izbora u slučaju gustih mreža ( $d = 10$ ) čak i za najmanje  $k = 1$ . Ujedno se vidi sa grafika da je po pitanju optimalnosti izbora izvršioca u ovakvim scenarijima  $k$ -AAP više nego trostruko bolji od  $k$ -SAP/ $k$ -SAAP algoritama.



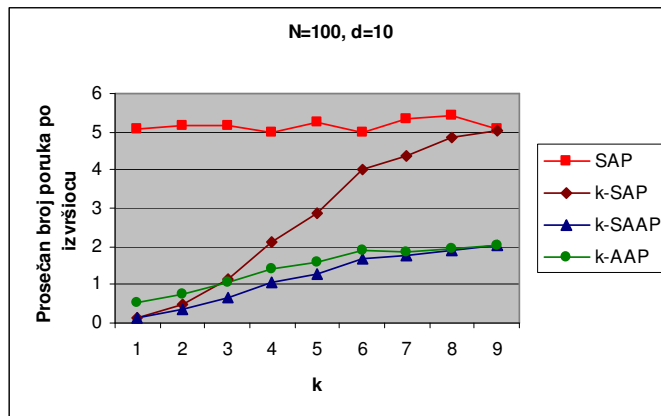
Slika 6.5.4 Uticaj gustine mreže na procenat optimalnosti izbora izvršioca



Slika 6.5.5 Komunikacioni troškovi  $k$ -AAP za male i retke mreže

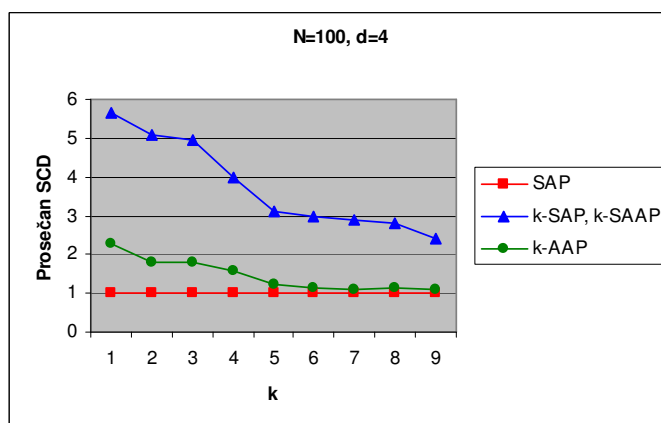
Prethodno prikazani rezultati pokazuju značaj  $k$ -AAP algoritma u optimalnosti pronalaženja najboljeg izvršioca. Ono što je takođe važno je koliki su komunikacioni troškovi. Na Slici 6.5.5, prikazani su rezultati za male i retke mreže za  $N = 10$  izvršilaca i za gustinu mreže  $d = 4$ . Poređen je prosečan broj poruka po izvršiocu za SAP,  $k$ -SAP,  $k$ -SAAP i  $k$ -AAP u funkciji parametra  $k$ . Može se zaključiti da po pitanju komunikacionog opterećenja  $k$ -AAP vrlo brzo dostiže nivo koji ima  $k$ -SAAP (oko dve poruke po izvršiocu u proseku) uz mnogo bolju optimalnost izbora izvršioca (videti Sliku 6.5.1).

Sličnu situaciju imamo i za velike i guste mreže. Prosečan broj poruka za scenario za  $N = 100$  izvršilaca i gustinu mreže  $d = 10$  prikazan je na Slici 6.5.6.



Slika 6.5.6  $k$ -AAP ima dobre osobine malog komunikacionog opterećenja i za velike i guste mreže

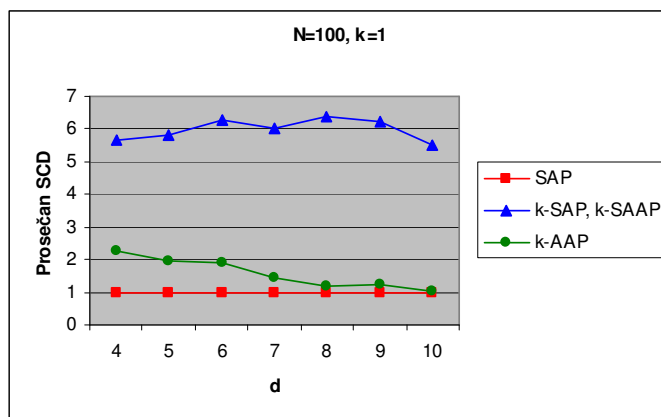
Ono što je posebno dobra osobina koju možemo izvući na bazi rezultata je da pored povećanja optimalnosti odlučivanja o izboru najboljeg izvršioca, je i to što slično  $k$ -SAAP algoritmu i  $k$ -AAP ima komunikaciono opterećenje koje je u pojedinim scenarijima skoro konstantno i može poslužiti u sistemima gde je neophodno ravnomerno raspoređivanje utroška energije. Još jedno zapažanje je da prosečan broj poruka po izvršiocu za ovaj algoritam nikada nije prelazio dve poruke po izvršiocu. Ovo je takođe korisna informacija prilikom projektovanja pojedinih sistema.



Slika 6.5.7 Prosečan SCD u funkciji od  $k$

Treća mera koja je korišćena prilikom procene kvaliteta nekog algoritma je prosečan SCD definisan jednačinom (5.2.1) u prethodnom poglavlju. Što je ova mera bliža jedinici, to je neki algoritam bolji jer optimalnije određuje izvršioca. Na Slici 6.5.7 prikazan je prosečan SCD u zavisnosti od veličine  $k$  za slučaj velikih, a retkih mreža što predstavlja najsloženiji scenario. Vidi se da  $k$ -AAP algoritam dostiže jedinični prosečan SCD tek pri  $k = 9$ . Ovde je jasno da  $k$ -AAP nije algoritam koji bi garantovao jedinični SCD tj. koji bi uvek optimalno pronalazio izvršioca, ali to i nije oblast u kojoj bi se ovaj algoritam primenjivao.

Na Slici 6.5.8 je prikazan uticaj gustine mreže na prosečan SCD za velike mreže, ali za najteži slučaj – najmanji  $k = 1$ . Ovde se vidi da  $k$ -AAP algoritam dostiže jediničan SCD tek u gustim mrežama što je i očekivano jer je tada mnogo lakše pronaći optimalnog izvršioca obzirom da je dosta njih dovoljno blizu.



Slika 6.5.8 Prosečan SCD u funkciji gustine mreže  $d$

Na kraju ovog poglavlja možemo zaključiti da je lokalizovani aukcijski agregacioni algoritam ( $k$ -AAP) pokazao znatno bolje osobine u smislu povećanja optimalnosti izbora izvršioca, kao i smanjenja komunikacionog opterećenja u poređenju sa ostalim algoritmima (SAP,  $k$ -SAP i  $k$ -SAAP) na bazi uvođenja prethodnog topološkog znanja o mreži koje se dobija u toku kofigurisanja mreže i koje se kasnije održava u toku rada razmenom poruka.

# 7 Primena aukcija za poboljšanje iMesh algoritma za pronalaženje izvršioca

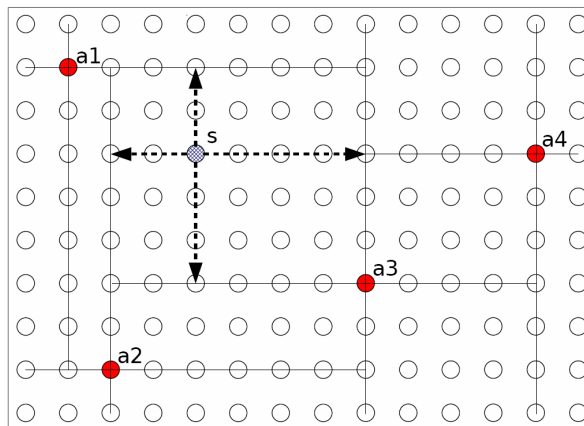
U ovom poglavlju je prikazana primena lokalizovanog aukcijskog agregacionog algoritma ( $k$ -SAAP) radi poboljšanja jednog od postojećih algoritama za otkrivanje lokacije izvršioca koji bazira na informacionoj mreži – iMesh, objavljen u [LSS09]. Odabran je  $k$ -SAAP jer on ima najbolje karakteristike za ovu primenu. U [LSS09] dat je algoritam otkrivanja izvršioca za pravilne strukture mreža – rešetkaste strukture. Za proizvoljne strukture mreža istraživanja su u toku, a algoritam, njegove performanse i deo rezultata prikazani su u [LM12].

## 7.1 iMesh algoritam

Osnovni *iMesh* algoritam [LSS09] predstavlja poboljšanje algoritma za izbor izvršioca na bazi kvoruma [SLJ08] i predstavlja uopštenje GCLP algoritma [TV04]. Kod kvorum algoritma mreža je podeljena na kolone i redove, a poruke o izvršiocima šalju se duž svih redova i kolona u mreži. Osnovno poboljšanje ovog algoritma je primena pravila stopiranja (eng. *blocking rule*) koje prekida dalju propagaciju poruke ukoliko bi se ušlo u zonu gde se već ima informacija o bližem izvršiocu. Na ovaj način je znatno smanjen broj poruka koji je potreban za konstrukciju informacione mreže. Osnovni *iMesh* se koristi u mrežama sa pravilnom strukturom (npr. rešetkastom).

Mrežni model koji se koristi za *iMesh* algoritam je nešto drugačiji nego do sada i sastoji se od skupa sa  $m$  senzora  $S = \{S_1, \dots, S_m\}$  i skupa sa  $n$  aktuatora  $A = \{A_1, \dots, A_n\}$ . Pri tome je broj  $n$  znatno manji od broja  $m$ , barem za red veličine. Komunikacioni radijus senzora i aktuatora je isti, tj.  $r_S = r_A = r$ . Mrežni graf je definisan kao  $G = (V, E)$ , gde je  $V = S \cup A$  skup čvorova grafa koji uključuju i senzore i aktuatore, i postoji grana

grafa  $e$  između čvorova  $u$  i  $v$ ,  $e = (uv) \in E$ , koja je element skupa grana grafa  $E$ , ako i samo ako je Euklidsko rastojanje među njima manje ili jednako komunikacionom radijusu, tj.  $|uv| \leq r$ . Ovde takođe važe i sve pretpostavke koje se koriste u simulacijama postavljene u odeljku 1.8.



Slika 7.1.1 Ilustracija *iMesh* algoritma

Scenario sa 136 senzora i 4 aktuatora prikazan je na Slici 7.1.1. PUNE linije predstavljaju informacionu mrežu koja se formira na bazi pozicija izvršilaca po sledećem principu. Svaki izvršilac šalje poruku sa informacijom o svojoj poziciji u sva četiri geografska pravca sve dok se ne zadovolji neki od sledećih uslova koji prekidaju dalju propagaciju poruke u tom pravcu:

- 1) dostignut je ivični red ili kolona
- 2) čvor kojem je trenutno poslata informacija već ima informaciju o nekom bližem izvršiocu (pravilo blokiranja)

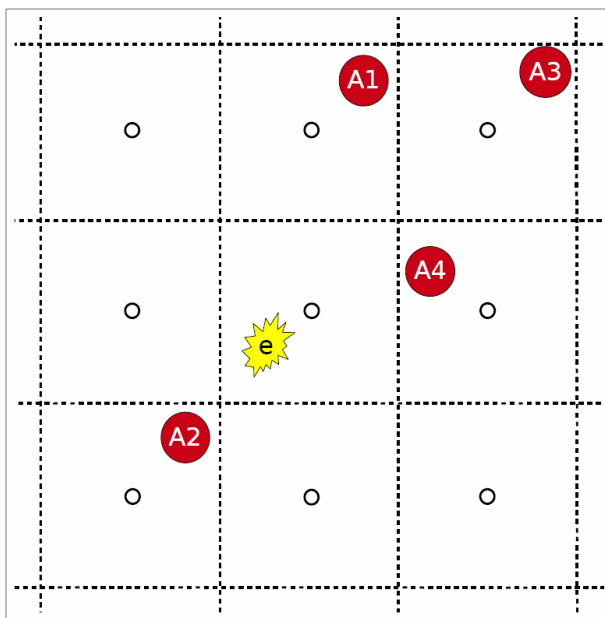
Kao rezultat formiranja informacione mreže ukupno 58 senzora ima informaciju o bar jednom izvršiocu. Pored toga interesantno je uočiti da ne postoji ni jedan senzor koji u jednom ili dva hopa ne može saznati lokaciju nekog od izvršilaca.

Ilustracija načina otkrivanja lokacije izvršioca je data na Slici 7.1.1 isprekidanim linijama gde senzor  $s$  šalje poruku u sva četiri pravca u cilju potrage za nekim od izvršilaca (eng. *cross lookup*). Kada se linija potrage preseče sa nekom od linija informacione mreže, taj čvor šalje nazad poruku o lokaciji izvršioca za koju on zna. Kada dobije sve povratne poruke, senzor  $s$  može da odluči koji je izvršilac najpogodniji da



reaguje na događaj i potom mu šalje informaciju o tome (koristeći npr. neki geografski protokol rutiranja). U konkretnom scenariju, senzor  $s$  će moći da odluči i da izabere stvarno najboljeg izvršioca jer ima informaciju o svim izvršiocima u mreži.

U prethodno opisanom mrežnom modelu su i senzori i aktuatori bili sa jednakim komunikacionim radijusom i formirali su graf sa rešetkastom strukturom. Kao korak dalje u pravcu proizvoljne mrežne strukture ćemo uzeti osnovni mrežni model aktuatora koji se koristi u ovoj disertaciji i koji je postavljen u odeljku 1.8, koji dozvoljava proizvoljne pozicije aktuatorima uz uslov da su svi u povezanom komunikacionom grafu. Pri tome su senzori i dalje imaju prostornu 2D rešetkastu strukturu i imaju informaciju o pozicijama aktuatora u njihovoj ćeliji. Za ovakav scenario, u određenim situacijama algoritam na bazi informacione mreže *iMesh*, ne pronalazi najbližeg izvršioca. To je ilustrovano na Slici 7.1.2. Ovde senzor koji detektuje događaj  $e$ , već ima informaciju o aktuatorima A4 i A1 i odlučuje da je A4 najbliži jer nema informaciju o aktuatoru A2 koji je stvarno najbliži. Inače uspešnost pronalaska najbližeg izvršioca je za *iMesh* algoritam pokazana da je oko 95% u proseku [LSS09]. Ovde se javlja i osnovni motiv za primenu aukcija radi poboljšanja pronalaženja najbližeg izvršioca uz malo dodatno komunikaciono opterećenje.



Slika 7.1.2 Scenario u kojem se ne pronalazi najbliži izvršilac

## 7.2 Poboljšanje korišćenjem $k$ -SAAP algoritma

U prethodnom odeljku je opisana jedna od situacija u kojima se ne pronalazi najbliži izvršilac korišćenjem iMesh algoritma što predstavlja osnovni motiv za primenu aukcija radi poboljšanja procenta pronalaženja najbližeg izvršioca.

Od prikazanih aukcijskih algoritama, ovde koristimo lokalizovani aukcijski agregacioni algoritam ( $k$ -SAAP) zbog njegove osobine da ima najmanje dodatno komunikaciono opterećenje od svih algoritama (videti Slike 6.5.5 i 6.5.6 u prethodnom poglavlju). Tačne granice dodatnog prosečnog komunikacionog opterećenja su iskazane teoremama u prethodnom poglavlju.

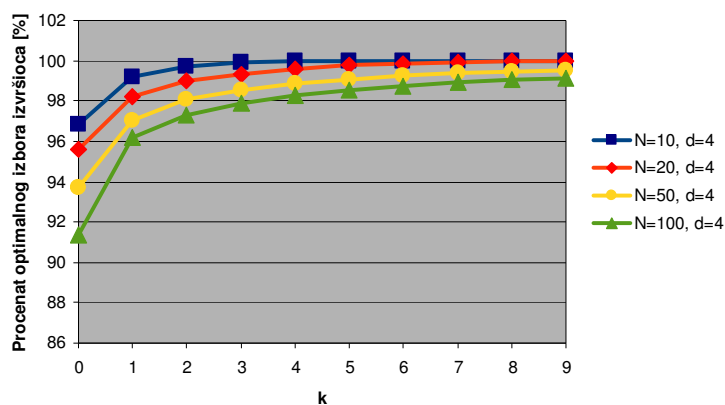
Za istraživanje poboljšanja procenta pronalaženja najbližeg izvršioca koristićemo dva scenarija. U oba scenarija će senzori biti raspoređeni tako da formiraju rešetkastu strukturu kao i u [LSS09]. Razlika u scenarijima, a samim tim i u mrežnim modelima, će biti u aktuatorima. Za razliku od [LSS09] gde su i aktuatori bili elementi rešetkaste mrežne strukture sa istim komunikacionim dometom kao i senzori, ovde u prvom scenariju aktuatori formiraju povezanu mrežu po UDG modelu koristeći MIN-DPA algoritam [OSY08] koji razmešta aktuatore na pseudo-slučajan raspoređujući prosečan broj susednih izvršilaca po uniformnoj raspodeli. U drugom scenariju aktuatori imaju proizvoljnu prostornu raspodelu odabranu na slučajan način. Kompletan simulacioni algoritam dat je u dodatku D3.

Dvodimenzionalni prostor koji je ovde posmatran je dimenzija  $1000m \times 1000m$ , sa  $32 \times 32$  senzora raspoređenih po pravilnoj dvodimenzionalnoj rešetkastoj strukturi. Broj izvršilaca (aktuatora)  $N$  je uziman iz skupa  $A = \{10, 20, 50, 100\}$ . U drugom scenariju se komunikacioni radijus aktuatora bira iz skupa  $R = \{50, 100, 200, 400\}$ . Simulacije su vršene tako da je uzeto po  $1000$  različitih konfiguracija izvršilaca, a pozicija svakih od  $1024$  senzora je uzimana za lokaciju događaja tako da je ukupno vršeno po  $1.024.000$  simulacija za svaki scenario.

Meren je procenat pronalaženja najbližeg (optimalnog) izvršioca (aktuatora) u zavisnosti od broja hopova  $k$  i od gustine mreže  $d$  za prvi scenario, a u drugom scenariju je ovaj procenat meren u zavisnosti od komunikacionog radijusa aktuatora. Kao mera komunikacionog opterećenja i tako dobijenog benefita je meren broj poruka koje

dovode do skraćenja ukupno pređenog puta izvršioca za jedan metar.

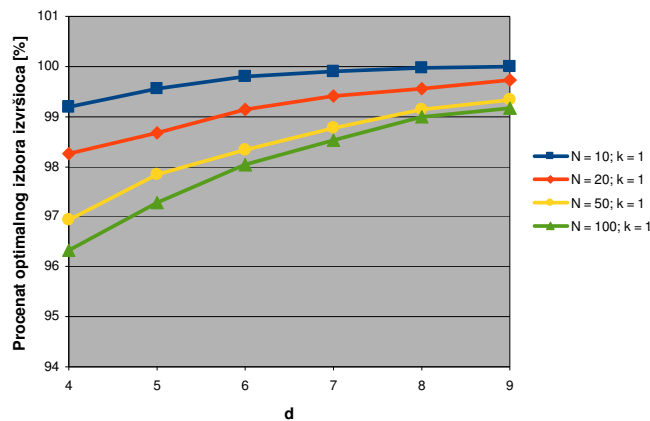
U [LSS09] je pokazano da je prosečan procenat uspešnosti pronalaženja najbližeg izvršioca oko 95%. Mi smo potvrdili ovaj rezultat jer smo dobili da je u prvom scenariju taj procenat 94.3%, a za drugi scenario on iznosi 96.1% u proseku. Ovo se može videti, za slučaj  $k = 0$ , na Slici 7.2.1 što praktično predstavlja ponovljene rezultate dobijene u [LSS09], dakle bez aukcija.



Slika 7.2.1 Efikasnost pronalaženja optimalnog izvršioca za prvi scenario u zavisnosti od broja hopova

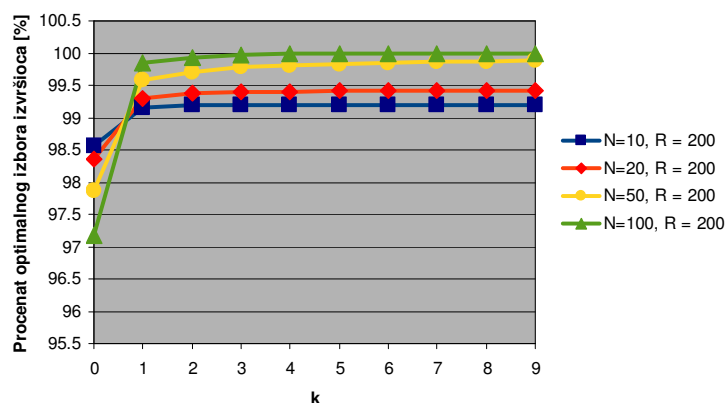
Uvođenjem lokalizovanog aukcijskog agregacionog algoritma, nakon što je izvršilac već pronađen korišćenjem *iMesh* algoritma, daje sledeće rezultate. Za male i retke mreže ( $N = 10$ ,  $d = 4$ , videti Sliku 7.2.1), za  $k \geq 4$ , postizemo 100% efikasnost pronalaženja optimalnog izvršioca. Ovakvu efikasnost moguće je postići čak i za  $k = 1$  u za slučaju gustih mreža ( $N = 10$ ,  $d \geq 8$ , videti Sliku 7.2.2) što je očekivan rezultat jer je tada graf mreže veoma blizak kompletnom grafu gde je skoro svakom čvoru mreže bilo koji drugi čvor prvi komšija. Na Slici 7.2.2 je može se uočiti i uticaj povećanja gustine mreže i zaključiti da, bez obzira na broj izvršilaca, što je mreža gušća veća je i efikasnost pronalaženja optimalnog izvršioca.

Na Slici 7.2.1 se može primetiti i da je najveće poboljšanje efikasnosti pronalaženja optimalnog izvršioca za  $k = 1$ . Analizom ukupnih rezultata se izvodi zaključak da je dobijeno poboljšanje od 4.7% što daje ukupnu prosečnu efikasnost pronalaženja optimalnog izvršioca od 99.1%.



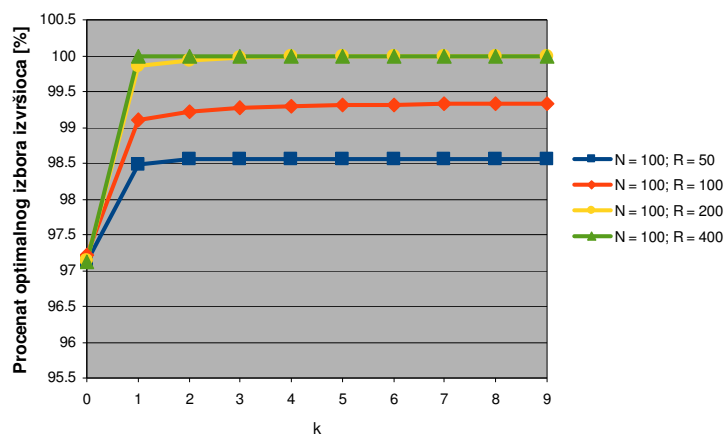
Slika 7.2.2 Efikasnost pronalaženja optimalnog izvršioca za prvi scenario u zavisnosti od gustine mreže

U slučaju drugog scenarija izvršioci su raspoređeni na slučajan način i nema nikakvog uslova u pogledu povezanosti mreže izvršilaca. Na osnovu rezultata prikazanih na Slici 7.2.3, za  $k = 0$ , može se zaključiti da u slučaju drugog scenarija *iMesh* algoritam ima bolju efikasnost pronalaženja optimalnog izvršioca u poređenju sa prvim scenarijom. Posebno je interesantno uočiti da se za  $k > 1$  ne dobija neko značajnije poboljšanje efikasnosti pronalaženja optimalnog izvršioca. Pored toga, veće mreže odlikuje značajnije povećanje efikasnosti za malo  $k$  nego manje mreže. Ovo je posebno uočljivo za  $k = 1$ , gde za  $N = 10$  imamo povećanje efikasnosti od oko  $0.7\%$ , a za  $N = 100$  to povećanje iznosi preko  $2.5\%$ . Uvođenjem lokalizovanog aukcijskog agregacionog algoritma dobija se prosečno poboljšanje od  $1.6\%$  što daje ukupnu prosečnu efikasnost pronalaženja optimalnog izvršioca od  $99.6\%$  za drugi scenario.



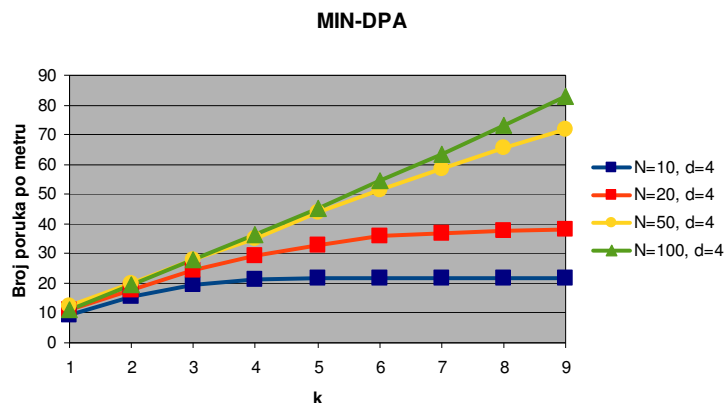
Slika 7.2.3 Efikasnost pronalaženja optimalnog izvršioca za drugi scenario u zavisnosti od broja hopova pri promenama broja izvršilaca

Za drugi scenario smo istraživali i uticaj promene radijusa  $R$  na efikasnost pronalaženja optimalnog izvršioca. Rezultati za sve četiri vrednosti radijusa ( $50$ ,  $100$ ,  $200$  i  $400m$ ) prikazani su na Slici 7.2.4 za slučaj  $N = 100$  izvršilaca. Ovde je takođe najveće poboljšanje za slučaj  $k = 1$  (pri tome za  $R \geq 200$  dostiže efikasnost od  $100\%$ ). Za ostale  $k$ , ne dobija se nikakvo značajno poboljšanje.



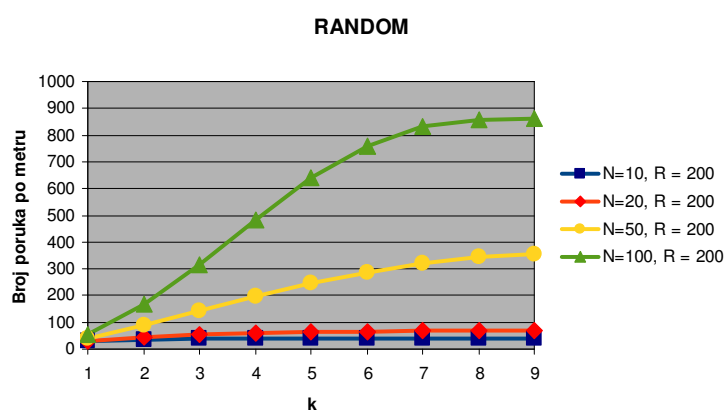
Slika 7.2.4 Efikasnost pronalaženja optimalnog izvršioca za drugi scenario u zavisnosti od broja hopova pri promenama komunikacionog radijusa

Radi istraživanja komunikacionih troškova, merili smo broj poruka potrebnih da se pronađe najbliži aktuator. Da bismo uočili kolika se korist dobija od dodatne komunikacije i aukcija, računali smo broj poruka po metru. Ovim se dobija informacija koliko je potrebno poruka da bi se dobio jedan metar skraćenja ukupno pređenog puta izvršioca. Rezultati za prvi scenario su prikazani na Slici 7.2.5. Može se uočiti da sa povećanjem  $k$ , za manje mreže, ne povećava značajno broj poruka po metru, čak postaje konstantno. Ovo se može objasniti činjenicom da za takve mreže značajnije povećanje  $k$  ne donosi nikakvo poboljšanje (povećanje uštede u pređenom putu, a time i uštede energije) jer je najbolji izvršilac već pronađen za manje vrednosti  $k$  što smo videli i na Slici 7.2.1.



Slika 7.2.5 Prosečan broj potrebnih poruka za jedan metar skraćenja pređenog puta izvršilaca za prvi scenario

Broj poruka po metru skraćenja ukupno pređenog puta za drugi scenario prikazan je na Slici 7.2.6. Ono što razlikuje ove rezultate u odnosu na prvi scenario je značajno povećanje broja poruka po metru sa povećanjem broja izvršilaca, višestruko više nego u slučaju prvog scenarija. Ovo se može objasniti činjenicom da MIN-DPA algoritam u prvom scenariju formira mrežu izvršilaca sa kontrolisanim prosečnim brojem komšija i to na uniforman način, a drugom scenariju su izvršiooci postavljeni na potpuno proizvoljan način i samim tim nije kontrolisan prosečan broj komšija što dovodi do značajnog povećanja broja poruka potrebnih da se pronade bliži izvršilac, pogotovo za veće mreže.



Slika 7.2.6 Prosečan broj potrebnih poruka za jedan metar skraćenja pređenog puta izvršilaca za drugi scenario

Može se zaključiti da je najveće poboljšanje efikasnosti pronalaska boljeg izvršioca od onog koji je već pronađen putem osnovnog *iMesh* algoritma, moguće dobiti za male vrednosti  $k$ . Ovo ujedno znači i da je tada najmanje dodatno komunikaciono opterećenje što je veoma povoljno i potvrđuje opravdanost uvođenja lokalizovanog aukcijskog agregacionog algoritma radi poboljšanja osnovnog *iMesh* algoritma.

## 8 Zaključak i budući rad

Polazeći od prostog aukcijskog algoritma koji su uveli Melodia *et al.* u [MPGA05] i [MPGA07], i sličnog algoritma pod imenom *Murdoch* koji su u robotici uveli Gerkey i Mataric [GM02], u ovoj disertaciji je urađeno nekoliko poboljšanja ovog algoritma, predloženo je nekoliko novih aukcijskih agregacionih algoritama za izbor izvršioca u bežičnim senzorskim i aktuatorskim mrežama i data je primena jednog algoritma za poboljšanje postojećeg algoritma za otkrivanje izvršioca na bazi informacione mreže (*iMesh* algoritam [LSS09]).

Prosti aukcijski algoritam koji su uveli Melodia *et al.*, kao i *Murdoch* algoritam, je korišćen u kompletnim mrežama gde svi izvršioci mogu međusobno da komuniciraju *direktno*. Ova pretpostavka nema smisla za veće mreže ili ukoliko izvršioci imaju mali komunikacioni domet. Prvo poboljšanje koje smo uveli je primena prostog aukcijskog algoritma u multi-hop povezanim BSAM po UDG modelu i algoritam smo nazvali SAP. Ovaj algoritam odlikuje 100 procentna efikasnost u pronalaženju najbližeg izvršioca jer svi izvršioci učestvuju u aukciji na bazi koje se određuje koji je najbolji, tj. najbliži jer je to bio, kao i u većini drugih radova, kriterijum pri izboru izvršioca. Međutim, SAP algoritam odlikuju veoma veliki komunikacioni troškovi. Za situacije kada nije neophodno *uvek* pronaći najbližeg izvršioca, predložili smo lokalizovanu varijantu SAP algoritma – *k-SAP*. Ovaj algoritam ima značajno manje komunikaciono opterećenje, u nekim slučajevima čak i do 200 puta manje. Osim za male i guste mreže i za veće vrednosti *k*, za koje i *k-SAP* može da ima 100 procentnu efikasnost u pronalaženju najbližeg izvršioca, ona se kretala u granicama od 20-70%.

Ispitivan je uticaj gustine mreže. Uočeno je da SAP algoritam ima tendenciju blagog smanjenja komunikacionih troškova sa povećanjem gustine mreže, što se objašnjava činjenicom da je više izvršioca međusobno bliže u gustim mrežama pa je ukupno potrebno manje poruka da se dođe do odluke o izboru. Suprotno, *k-SAP* algoritam pokazuje tendenciju blagog povećanja komunikacionih troškova sa



povećanjem gustine mreže. Razlog se može pronaći u činjenici da je  $k$ -SAP lokalizovan algoritam gde ne učestvuju svi izvršioci u aukciji, pa se sa povećanjem gustine mreže za dato  $k$  povećava i broj učesnika na aukciji, a samim tim i broj poruka koje se razmenjuju, dakle i komunikacioni troškovi rastu.

Radi poboljšanja efikasnosti izbora najbližeg izvršioca, kao poboljšanje  $k$ -SAP algoritma uveden je dodatak u vidu tzv. pohlepnog (eng. *greedy*) algoritma. Nakon što je  $k$ -SAP algoritam završen i odluka o (potencijalno) najboljem izvršiocu doneta, taj izvršilac pokreće novu aukciju na bazi  $1$ -SAP algoritma sa ciljem da potraži eventualno boljeg izvršioca među susednim izvršiocima koji su unutar njegovog komunikacionog radijusa. Ukoliko nema boljeg izvršioca, tada ne dolazi do preraspodele posla i on ostaje dodeljen prvobitnom izvršiocu. U slučaju da je pronađen bolji izvršilac, tada se posao njemu prosleđuje i on ponovo inicira  $1$ -SAP algoritam. Algoritam (označen sa  $k$ -SAPG) se ponavlja sve dok ima boljih izvršilaca među susedima unutar komunikacionog dometa onog koji inicira  $1$ -SAP aukciju. Rezultati pokazuju da uvođenje ovog proširenja poboljšava efikasnost pronalaženja najbližeg izvršioca za 2,78 puta u proseku. Za male, guste ili mreže sa velikim  $k$  moguća je, kao i kod  $k$ -SAP algoritma, 100% efikasnost, a za ostale se kretala od 47% do 98%. Prosečno komunikaciono opterećenje je oko 0.4 poruke po izvršiocu (stvarne vrednosti su se kretale od 0.07 do 1.01). Pri tome, najbolji rezultat dobija se za velike mreže (ali ne i guste) uz male vrednosti  $k$ . Posebno dobra karakteristika  $k$ -SAPG algoritma je što u slučaju većih mreža ( $N > 10$ ) ima približno konstantno dodatno komunikaciono opterećenje u odnosu na  $k$ -SAP algoritam.

Nakon ovih, u osnovi poboljšanja prostog aukcijskog algoritma, predloženo je pet novih aukcijskih agregacionih algoritama. Svi ovi algoritmi koriste na određeni način tehniku agregacije sa ciljem smanjenja komunikacionih troškova. Prvi algoritam je prosti aukcijski agregacioni algoritam (u oznaci SAAP). On polazi od SAP algoritma i u informacionoj fazi se isto ponaša, a ušteda je ostvarena u aukcijskoj fazi. Uveden je princip roditelj-dete gde svaki čvor ima samo jednog roditelja i svaki čvor licitira samo prema svojim roditeljima, a roditelji čekaju odgovore od sve dece i tek tada šalju licitaciju svom roditelju. Na ovaj način je odlučivanje distribuirano po mreži jer svaki roditelj lokalno odlučuje o najboljem izvršiocu. Obzirom da kod SAAP algoritma svi izvršioci učestvuju u aukciji, najbližeg izvršioca je uvek moguće naći. Pri tome su

komunikacioni troškovi takvi da se SAP i SAAP ponašaju skoro identično za male i guste mreže i tu SAAP algoritam ne donosi neko poboljšanje, a unosi kašnjenje u odlučivanju pa u tim slučajevima nije pogodan za korišćenje. Međutim, za slučaj velikih i retkih mreža prosečan broj potrebnih poruka po izvršiocu manji za čak do šest puta u odnosu na SAP.

Lokalizovana varijanta SAAP algoritma ( $k$ -SAAP) se uvodi ako je potrebno dodatno smanjenje komunikacionih troškova, a da pri tome nije problem ukoliko nemamo 100% efikasnost izbora najbližeg izvršioca. Za  $k$ -SAAP algoritam je dokazana teorema da je broj poruka po izvršiocu uvek manji ili jednak dva kao i teorema da je taj broj uvek veći ili jednak  $2k/N$ . Ova osobina omogućava da se unapred izračunaju komunikacioni troškovi, što je veoma bitno za projektovanje mreža i određivanje životnog veka mreže. Pored toga ova osobina omogućava da se  $k$ -SAAP koristi kao dodatni algoritam na neki već postojeći u cilju njegovog poboljšanja uz poznate dodatne komunikacione troškove.

Treći aukcijski agregacioni algoritam predstavlja proširenje  $k$ -SAAP algoritma na bazi pohlepnog algoritma (u oznaci  $k$ -SAAPG) sa ciljem poboljšanja efikasnosti odlučivanja. Poboljšanja koja se dobijaju su praktično ista kao u slučaju  $k$ -SAPG algoritma.

Četvrti algoritam (aukcijski agregacioni algoritam - AAP) i peti algoritam koji je lokalizovana verzija AAP –  $k$ -AAP, uvode suštinski drugačiji koncept. Do sada ni jedan algoritam nije podrazumevao da čvorovi u mreži imaju bilo kakvo prethodno topološko znanje o drugim čvorovima u mreži. AAP algoritam to uvodi i time omogućava dalju lokalizaciju odlučivanja uz povećanje efikasnosti i smanjenje komunikacionih troškova. Rezultati pokazuju da za male i retke mreže  $k$ -AAP postiže 100% efikasnost pronalaženja najbližeg izvršioca već za  $k = 1$  (kada su komunikacioni troškovi najmanji), za razliku od  $k$ -SAP i  $k$ -SAAP koji to postižu tek za  $k = 5$ . Sličnu prednost ima i kada su veće mreže u pitanju. Pored toga u slučaju gustih mreža, moguće je postići 100% efikasnost izbora čak i za  $k = 1$ . Po pitanju optimalnosti izbora izvršioca za ovakve scenarije  $k$ -AAP je više nego trostruko bolji o  $k$ -SAP/ $k$ -SAAP algoritama. Pri tome komunikacioni troškovi su tek neznatno veći od  $k$ -SAAP.

Primena aukcija radi poboljšanja postojećeg algoritma za izbor izvršioca na bazi informacione mreže (*iMesh* algoritam [LSS09]) je pokazala da je moguće povećati

efikasnost izbora sa 95%, koliko osnovni *iMesh* omogućava, na 99.3% u proseku. Od svih aukcijskih algoritama, korišćen je  $k$ -SAAP algoritam jer on ima najbolja svojstva sa stanovišta dodatnog komunikacionog opterećenja, kao i unapred poznato dodatno komunikaciono opterećenje, a ne zahteva nikakvo topološko predznanje o mreži. Najveće poboljšanje efikasnosti pronalaska boljeg izvršioca od onog koji je već pronađen putem osnovnog *iMesh* algoritma, moguće je dobiti za male vrednosti  $k$  (za  $k = 1$  je najveće poboljšanje). Ovo ujedno znači i da je tada najmanje dodatno komunikaciono opterećenje što je veoma povoljno sa aspekta malog povećanja komunikacionih troškova koje  $k$ -SAAP algoritam unosi.

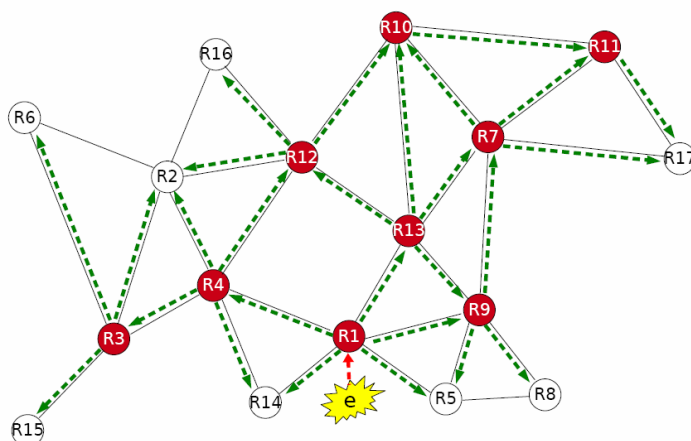
U toku rada na ovoj disertaciji pojavilo se nekoliko ideja od kojih su neke istražene u okviru master-diplomskih radova. U [J10] je istraživana mogućnost primene lokalizovanih aukcija pri odlučivanju u bežičnim senzorskim mrežama, korišćenjem realnih čvorova tipa IRIS kompanije Crossbow (sada Memsic) [M1]. Zbog zatvorenog sistema programskog koda na čvorovima, uočene su znatne poteškoće pri implementaciji više od jednog ili dva hopa. Rezultate ovih istraživanja objavili smo u [JM10] i [MJ11]. Trenutno je u razmatranju primena drugačijeg operativnog sistema za ove čvorove koji bi prevazišao ove probleme.

U diplomskom-master radu [G10] istraživana je mogućnost primene aukcijskih agregacionih protokola baziranih na agentima u multi-hop bežičnim robotskim mrežama. Ovde je ideja bila da se mreža izvršilaca (robota) podeli na ćelije i da se u svakoj ćeliji centralno pozicionira po jedan statičan čvor – agent koji će imati informacije o svim događanjima unutar ćelije za koju je nadležan i koji može da komunicira sa susednim agentima. Time je odlučivanje o izboru izvršioca prebačeno na agente, a izvršioci imaju samo zadatak javljanja agentima u svojim ćelijama i izvršenja dodeljenog zadatka. Pokazano je da se na ovaj način dodatno drastično smanjuje komunikaciono opterećenje [GM11]. Trenutno je u toku istraživanje primene *iMesh* algoritma na agentima za odabir izvršilaca.

Pokrenuta je zajednička istraživačka saradnja sa francuskom istraživačkom grupom sa Univerziteta u Strazburu. Dobili smo pristup jednoj od najvećih bežičnih senzorskih mreža (Senslab [S1]) kojoj je moguće pristupiti putem Interneta i gde ćemo moći da praktično proverimo neke od algoritama.

Jedno od poboljšanja u daljem radu je u situacijama gde je potrebno sa

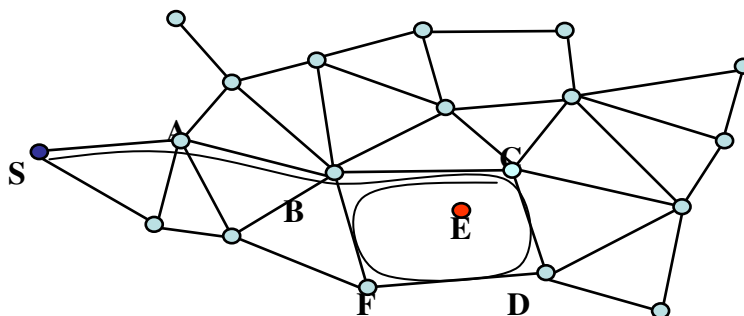
potpunom sigurnošću utvrditi koji je izvršilac najbolji. Tada je neophodno konsultovanje svih izvršilaca u mreži. Jedan od načina da se to uradi, a da se pri tome imaju manji komunikacioni troškovi od prostog plavljenja mreže je upotreba podskupa izvršilaca koji su odabrani na takav način da se može preko njih doći do bilo kog drugog izvršioca. Ovaj podskup se naziva kičma mreže (eng. *backbone*) [SSW05] i ilustrovan je crvenim krugovima na Slici 8.1.



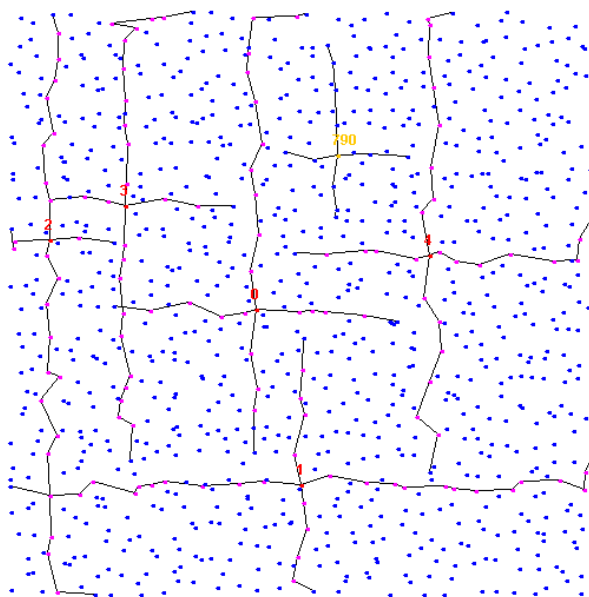
Slika 8.1 Ilustracija odabira određenog podskupa izvršilaca koji čine 'kičmu' mreže

U ovoj disertaciji su samo SAP i SAAP algoritmi obezbeđivali da se najbliži izvršilac uvek može naći, ali uz visoke komunikacione troškove. Za slučajeve kada je obavezno pronaći najbližeg izvršioca, ali uz manje komunikacione troškove, moguće je koristiti RFTA algoritam (*Routing with Face Traversal Auctions*) čiju ideju ovde samo predložimo, a buduća istraživanja treba da pokažu kakva svojstva ima. RFTA algoritam bi bazirao na GFG algoritmu (*Greedy Face Greedy*) koji je prvi put objavljen u [BSMU99], primenjen na problem čuvanja podataka u [RK+02] u GHT algoritmu koji je objašnjen u pregledu literature, a tačno objašnjenje je dato u [FS06]. Na Slici 8.2 se vidi objašnjenje na primeru mreže koja je predstavljena Gabrielovim grafom po UDG modelu. Rutiranje kreće od čvora  $S$  ka čvoru  $E$ , koji nije deo mreže, korišćenjem GFG rutiranja. Kao posledica se detektuje petlja  $BCDFB$ . Ovo je moguće iskoristiti za slučaj problema izbora izvršioca. Ukoliko je  $E$  u stvari događaj, koji senzori otkrivaju, a ostali čvorovi su izvršioци. Tada neka se na neki način informacija o događaju dojadi nekom od izvršilaca, on postaje kolektor i započinje aukciju. Neka je kolektor izvršilac  $B$ , tada

on pokreće aukciju, a čvorovi  $C$ ,  $D$  i  $F$  mu odgovaraju sa svojom cenom da reaguju na događaj. Nakon obilaska petlje, kolektor  $B$  odlučuje koji je najbliži i obaveštava ga. Ovaj mehanizam, iako nije jednostavan, trebalo bi da ima znatno manje komunikacione troškove nego SAP i SAAP i da obezbedi pronalaženje najbližeg u svim slučajevima.



Slika 8.2 Ilustracija RFTA algoritma za Gabrielov graf po UDG modelu



comcom

Slika 8.3 iMesh struktura za scenario sa 5 izvršilaca sa proizvoljnom ali uniformnom strukturom i gustom mrežom

U ovoj disertaciji korišćen je lokalizovani aukcijski agregacioni algoritam za poboljšanje efikasnosti pronalaženja optimalnog izvršioca nakon što je neki od izvršilaca pronađen upotrebom *iMesh* algoritma. Ovo je pokazano u poglavlju 7. Kao

dalje proširenje ovog rada je istraživanje scenarija koji su još bliži realnoj situaciji kada su i senzori i aktuatori postavljeni na potpuno proizvoljan način. Na putu ka tome postoji još nekoliko jednostavnijih scenarija. Prvi od njih je kada imamo senzore i aktuatore raspoređene u nepravilnoj rešetkastoj formaciji (eng. *gridy*). Zatim kada su i senzori i aktuatori raspoređeni proizvoljno, ali uniformno po nadgledanom prostoru kao što je prikazano na Slici 8.3. Na kraju imamo scenario sa potpuno proizvoljnim položajima senzora i aktuatora, kao i postojanje delova prostora koji nisu pokriveni sensorima što daje potpuno proizvoljnu topologiju. Ova istraživanja su u toku.

U ovoj disertaciji je kao kriterijum optimalnog izvršioca korišćeno udaljenje između događaja i određenog izvršioca i optimalan, tj. najbolji izvršilac je smatran onaj koji je najbliži događaju. Međutim moguće je koristiti i druge metrike umesto distance. Na primer, moguće je koristiti metriku koja maksimizuje rezidualnu energiju izvršilaca. Druga mogućnost je balansiranje energije među izvršiocima. Dalje moguće je kombinovati nekoliko metrika itd. Ovo su sve neki od pravaca daljih istraživanja na bazi ove disertacije.

# Literatura

- [AB06] N. Atay, B. Bayazit, "Mixed-Integer Linear Programming Solution to Multi-Robot Task Allocation Problem," Technical report WUCSE-2006-54, Washington University in Sent Louis, 2006.
- [AGB10] Kemal Akkaya, Ismail Guneydas, Ali Bicak, „Autonomous actor positioning in wireless sensor and actor networks using stable-matching“, Int. J. Parallel, Emergent and Distributed Systems, 25, 6, pp. 439 – 464, 2010.
- [AK04] I. F. Akyildiz i I.H. Kasimoglu, “Wireless sensor and actor networks: research challenges”, Ad hoc networks, Vol.2, Issue 4, Oct. 2004, pp. 351-367
- [AL09] K. Al-Yafi, H. Lee, "Centralized versus market-based approaches to mobile task allocation problem: State-of-the-art," Proc. of the European and Mediterranean Conference on Information Systems, Izmir , 13-14 July 2009.
- [AS02] I. Aydin and C.-C. Shen, “Facilitating match-making service in ad hoc and sensor networks using pseudo quorum,” Proc. on IEEE ICCCN, 2002.
- [ASSC02] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, „Wireless sensor networks: a survey“, Computer Networks, 38(4), pp. 393–422, 2002.
- [AV10] I. F. Akyildiz, M. C. Vuran, *Wireless Sensor Networks*, Wiley, 2010.
- [B90] D. P. Bertsekas, ”The Auction Algorithm for Assignment and Other Network Flow Problems: A Tutorial”, Interfaces 20(4), 1990, pp. 133–149.
- [BDM09] R. Burkard, M. Dell'Amico, S. Martello, “Linear Sum Assignment Problem”, in: Assignment Problems, SIAM Monographs on Discrete Mathematics and Applications, 2009, pp.73-142.
- [BMSU99] P. Bose, P. Morin, I. Stojmenovic, J. Urrutia, „Routing with guaranteed delivery in ad hoc wireless networks“, Proc. of 3rd ACM Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications DIAL M99, pp. 48-55, 1999.
- [BM+05] W. Burgard, M. Moors, C. Stachniss, F.E. Schneider, “Coordinated multi-robot exploration”, IEEE Transactions on Robotics, Volume 21, Issue 3, June 2005, pp. 376 - 386
- [BY08] G. Bekey and J. Yuh, “The Status of Robotics”, IEEE Robotics & Automation Magazine, Vol. 15, Issue 1, 2008, pp. 80-86
- [CCXS10] X. Cao, J. Chen, Y. Xiao, Y. Sun, "Building-Environment Control With Wireless Sensor and Actuator Networks: Centralized Versus Distributed", IEEE Transactions on Industrial Electronics, vol.57, no.11, pp. 3596-3605, 2010.
- [CO+11] H. Chung, S. Oh, D.H. Shim, S.S. Sastry, "Toward Robotic Sensor Webs: Algorithms, Systems, and Experiments", Proceedings of the IEEE, vol.99, no.9, pp.1562-1586, 2011.
- [CV10] B. Coltin, M. Veloso, "Mobile robot task allocation in hybrid wireless sensor networks," Proc. on IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2932 – 2937, 2010.
- [CW92] D.G. Cattrysse and L.N. Van Wassenhove, “A survey of algorithms for the generalized assignment problem”, European Journal of Operational Research, Volume 60, Issue 3, 1992, pp. 260-272
- [DA10] F. Dressler, O.B. Akan, „A survey on bio-inspired networking“, Computer Networks, Volume 54, Issue 6, 2010, pp. 881–900.

- [DJM02] G. Dudek, M. Jenkin, E. Milios, "A Taxonomy of Multirobot Systems", T. Balch and L. E. Parker (Eds.), *Robot Teams: From Diversity to Polymorphism*, A. K. Peters, Wellesley, MA, 2002, pp. 3–22
- [DJMW96] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, "A taxonomy for multi-agent robotics", *Autonomous Robots*, vol. 3, 1996, pp. 375-397
- [DP10] W. Dargie, C. Poellabauer, *Fundamentals of Wireless Sensor Networks: Theory and Practice*, Wiley, 2010.
- [DZKS06] M.B. Dias, R. Zlot, N. Kalra, A. Stentz, "Market-Based Multirobot Coordination: A Survey and Analysis," *Proceedings of the IEEE*, Vol. 94, No. 7, pp. 1257-1270, 2006.
- [FGG06] Q. Fang, J. Gao, and L.J. Guibas, "Landmark-Based Information Storage and Retrieval in Sensor Networks", *Proc. IEEE INFOCOM*, pp. 286-297, 2006.
- [FGNW06] S. Funke, L. J. Guibas, A. Nguyen, and Y. Wang, "Distance sensitive routing and information brokerage in sensor networks", *Proc. on IEEE DCOSS*, 2006.
- [FIN04] A. Farinelli, L. Iocchi, i D. Nardi, "Multirobot systems: a classification focused on coordination", *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, Vol. 34, Issue 5, Oct. 2004, pp. 2015-2028
- [FINZ06] A. Farinelli, L. Iocchi, D. Nardi, V.A. Ziparo, "Assignment of Dynamically Perceived Tasks by Token Passing in Multirobot Systems", *Proceedings of the IEEE*, Vol. 94, Issue 7, July 2006, pp. 1271-1288
- [FS06] H. Frey, I. Stojmenovic, „On Delivery Guarantees of Face and Combined Greedy-Face Routing Algorithms in Ad Hoc and Sensor Networks“, *Proc. 12th ACM Annual Int. Conference on Mobile Computing and Networking MOBICOM*, pp. 390-401, 2006
- [G10] B. Gašparović, "Aukcijski agregacioni protokoli bazirani na agentima u multi-hop bežičnim robotskim mrežama", diplomski-master rad, Fakultet tehničkih nauka, Novi Sad, 2010.
- [GBH10] V.C. Gungor, L. Bin, G.P. Hancke, „Opportunities and Challenges of Wireless Sensor Networks in Smart Grid“, *IEEE Transactions on Industrial Electronics*, 57(10), 2010, pp. 3557 - 3564
- [GH09] V.C. Gungor, G.P. Hancke, „Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches“, *IEEE Transactions on Industrial Electronics*, 56(10), 2009, pp. 4258 – 4265
- [GM02] B.P. Gerkey, M.J. Mataric, "Sold!: Auction methods for multi-robot coordination," *IEEE Trans. on Robotics and Automation*, Vol. 18, No. 5, pp. 758-768, 2002.
- [GM02a] B. Gerkey, M. Mataric, "A market-based formulation of sensor-actuator network coordination", *Proceedings of the AAAI Spring Symposium on Intelligent Embedded and Distributed Systems*, Palo Alto, California, March 25-27, 2002, pp. 21-26
- [GM04] B.P. Gerkey i M.J. Mataric, "A formal analysis and taxonomy of task allocation in multi-robot systems", *Intl. J. of Robotics Research* 23(9), September 2004, pp. 939-954
- [GM11] B. Gasparovic, I. Mezei, "Auction Aggregation Protocols for Agent-based Task Assignment in Multi-hop Wireless Sensor and Robot Networks", *Proc. on IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM2011)*, 3-7 july 2011., Budapest, Hungary
- [GMVL04] A. Gage, R. Murphy, K. Valavanis, M. Long, "Affective Task Allocation for Distributed Multi-Robot Teams", *Technical report TR2006-26 for Center for Robot-Assisted Search and Rescue (CRASAR)*.
- [GS04] S. Giordano, I. Stojmenovic, "Position based routing algorithms for ad hoc networks: A taxonomy", in: *Ad Hoc Wireless Networking*, X. Cheng, X. Huang and D.Z. Du (eds.), Kluwer, 2004, pp. 103-136.
- [HS+10] J. Hu, L. Shen, Y. Yang, R. Lv, "Design and implementation of wireless sensor and actor network for precision agriculture," *IEEE International Conference on Wireless Communications, Networking and Information Security (WCNIS)*, pp.571-575, 2010.



- [HX11] X. Hu, B. Xu, "Task Allocation Mechanism Based on Genetic Algorithm in Wireless Sensor Networks", in *Applied Informatics and Communication Part 1* (Ed. D. Zeng), Springer, Vol. 224, 2011, pp. 46-58
- [IMRA06] Y. Chevaleyre, P.E. Dunne, U. Endriss, J. Lang, M. Lemaître, N. Maudet, J. Padget, S. Phelps, J.A. Rodríguez-Aguilar, and P. Sousa, "Issues in Multiagent Resource Allocation", *Informatica*, 30, 2006, pp. 3-31
- [J10] N. Janićijević, "Primena lokalizovanih aukcija pri odlučivanju u bežičnim senzorskim mrežama", diplomski-master rad, Fakultet tehničkih nauka, Novi Sad, 2010.
- [JH10] W.S. Jang, W.M. Healy, „Wireless sensor network performance metrics for building applications“, *Energy and Buildings*, Volume 42, Issue 6, pp. 862-868, 2010.
- [JM10] N. Janicijevic, I. Mezei, "Decision making by localized auctions and data fusion in building energy monitoring applications", *Proc. on X Int. Conf. on Systems, Automatic Control and Measurements, SAUM 2010, Nis, Serbia, 10-12 Nov. 2010*, pp. 246-249
- [K55] H.W. Kuhn, "The Hungarian Method for the Assignment Problem", *Naval Research Logistics Quarterly*, 2, 1955, pp. 83-97
- [KCM12] P. Kułakowski, E. Calle, J.L. Marzo, "Performance study of wireless sensor and actuator networks in forest fire scenarios", *International Journal of Communication Systems*, pp. 1099-1131, 2012.
- [KDPT02] K. S. Kwok, B. J. Driessen, C. A. Phillips, and C. A. Tovey, "Analyzing the multiple-target-multiple-agent scenario using optimal assignment algorithms," *J. Intell. Robot. Syst.*, vol. 35, 2002, pp. 111–122
- [KK+05] H. Kose, K. Kaplan, C. Mericli, U. Tatlıdede and L. Akin, "Market-Driven Multi-Agent Collaboration in Robot Soccer Domain", *Cutting Edge Robotics*, Pro Literatur Verlag, Germany / ARS, Austria, 2005, pp. 407-416
- [KPOK10] B. Kaleci, O. Parlaktuna, M. Ozkan, G. Kirlik, "Market-based task allocation by using assignment problem", *Proc. on IEEE International Conference on Systems Man and Cybernetics (SMC)*, Istanbul, Turkey, 2010, pp. 135-141.
- [KRS08] V. Kumar, D. Rus, G.S. Sukhatme, "Networked Robots", *Springer Handbook of Robotics*, S. Bruno; K. Oussama (Eds.), Springer, 2008, pp. 943-958
- [KZ94] C. Kube and H. Zhang, "Stagnation recovery behaviours for collective robots," in *Proc. IEEE/RSJ/GI Int. Conf. Intelligent Robots and Systems (IROS '94)*, 1994, pp. 1883–1890 vol.3.
- [LJ+00] J. Li, J. Jannotti, D.S.J.D. Couto, D.R. Karger, and R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing," *Proc. ACM MobiCom*, pp. 120-130, 2000.
- [LNS10] X. Li, A. Nayak, I. Stojmenovic, „Location Service in Sensor and Mobile Actuator Networks“, in: *Wireless Sensor and Actuator Networks: Algorithms and Protocols for Scalable Coordination and Data Communication*, (Eds. A. Nayak, I. Stojmenovic), Wiley Blackwell, Jan. 2010.
- [LM12] M.Lukic, I.Mezei, „Distributed Distance Sensitive iMesh based Service Discovery in Dense WSAAN“, prihvaćeno za prezentovanje na konferenciji AdHocNow 2012.
- [LSS09] X. Li, N. Santoro, I. Stojmenovic, "Localized Distance-Sensitive Service Discovery in Wireless Sensor and Actor Networks," *IEEE Transactions on Computers*, Vol. 58, No. 9, pp. 1275-1288, 2009.
- [LYNS12] X. Li, J. Yang, A. Nayak, i I. Stojmenovic, „Localized Geographic Routing to a Mobile Sink with Guaranteed Delivery in Sensor Networks“, prihvaćeno za publikovanje u *IEEE Journal on Selected Areas in Communications*, 2012.
- [M1] IRIS mote datasheet, Memsic Inc., [online] <http://www.memsic.com/products/wireless-sensor-networks/wireless-modules.html>
- [MBB09] A. Noor Mian, R. Baldoni, and R. Beraldi, "A Survey of Service Discovery Protocols in Multihop Mobile Ad Hoc Networks," *IEEE Pervasive computing*, Vol. 8 Issue 1, pp.66-74, 2009.

- [MB+10] M. Marin-Perianu, S. Bosch, R. Marin-Perianu, H. Scholten, i P. Havinga, „Autonomous vehicle coordination with wireless sensor and actuator networks“ ACM Trans. Auton. Adapt. Syst. 5, 4, Article 13, pp. 1-29, 2010.
- [MJ11] I. Mezei, N. Janicijevic, “Decision Making Based on Localized Auctions in Wireless Sensor Networks”, Proc. on the IEEE Eurocon2011 International Conference on a Computer as a Tool, 27-29.04.2011. Lisbon, Portugal, pp. 1-4
- [MLMS12] I. Mezei, M. Lukic, V. Malbasa and I. Stojmenovic, “Auctions and iMesh based Task Assignment in Wireless Sensor and Actuator Networks”, na drugom krugu recenzije u *Computer Communications*, Elsevier.
- [MM10] A. Mosteo, L. Montano, AMI-009-10-TEC: A survey of multi-robot task allocation, Tech. report, 2010, pp. 1-27
- [MM87] R. P. McAfee i J. McMillan, “Auctions and bidding” Journal of Economic Literature, vol. 25, no. 2, pp. 699–738, 1987.
- [MMS09] I. Mezei, V. Malbasa and I. Stojmenovic, “Auction Aggregation Protocols for Wireless Robot-Robot Coordination”, P.M. Ruiz and J.J. Garcia-Luna-Aceves (Eds.): ADHOC-NOW 2009, LNCS 5793, Springer-Verlag, pp. 180 – 193, 2009.
- [MMS10] I. Mezei, V. Malbasa and I. Stojmenovic, “Robot to Robot: Communication Aspects of Coordination in Robot Wireless Networks”, IEEE Robotics and Automation Magazine, Vol. 17, No. 4, pp.63-69, 2010.
- [MMS12] I. Mezei, V. Malbasa and I. Stojmenovic, “Greedy Extension of Localized Auction Based Protocols for Wireless Actuator Task Assignment”, prihvaćeno za publikovanje u časopisu Ad Hoc & Sensor Wireless Networks.
- [MPA06] T. Melodia, D. Pompili and I. F. Akyildiz, “A Communication Architecture for Mobile Wireless Sensor and Actor Networks”, Proceedings of the 3rd IEEE SECON 2006, pp. 109 – 118, 2006.
- [MPA10] T. Melodia, D. Pompili, I.F. Akyildiz, "Handling Mobility in Wireless Sensor and Actor Networks", IEEE Transactions on Mobile Computing, vol.9, no.2, pp.160-173, 2010.
- [MPGA05] T. Melodia, D. Pompili, V.C. Gungor, I.F. Akyildiz, „A Distributed Coordination Framework for Wireless Sensor and Actor Networks“, In Proc. of the 6th ACM international conference Mobihoc 2005, pp. 99-110, 2005.
- [MPGA07] T. Melodia, D. Pompili, V.C. Gungor, I.F. Akyildiz, “Communication and Coordination in Wireless Sensor and Actor Networks”, IEEE Transactions on Mobile Computing, Volume 6, Issue 10, Oct. 2007, pp. 1116 – 1129
- [MZKP08] N. Michael, M. Zavlanos, V. Kumar, G.J. Pappas, "Distributed multi-robot task assignment and formation control," IEEE International Conference on Robotics and Automation (ICRA 2008), pp. 128-133, 2008.
- [N1] Network Robot Forum, <http://www.scot.or.jp/nrf/English/>
- [N2] IEEE tehnički komitet za umreženu robotiku, <http://tab.ieee-ras.org/committeeinfo.php?tcid=15>
- [NG06] M. Nanjanath, M. Gini, "Auctions for task allocation to robots," Proceedings of the 9th International Conference on Intelligent Autonomous Systems, IOS Press, Tokyo, pp. 550-557, 2006.
- [NS10] A. Nayak, I. Stojmenovic, *Wireless Sensor and Actuator Networks: Algorithms and Protocols for Scalable Coordination and Data Communication*, Wiley Blackwell, Jan. 2010, ISBN 978-0-470-17082-3
- [OSY08] F. Atay Onat, I. Stojmenovic, H. Yanikomeroglu, “Generating Random Graphs for the Simulation of Wireless Ad Hoc, Actuator, Sensor, and Internet Networks,” Pervasive and Mobile Computing, Elsevier, Volume 4, Issue 5, pp. 597–615, 2008.
- [P07] D.W. Pentico, “Assignment problems: A golden anniversary survey”, European Journal of Operational Research, Volume 176, 2007, pp. 774–793
- [P08] L.E. Parker, “Multiple Mobile Robot Systems”, Springer Handbook of Robotics, S. Bruno; K. Oussama (Eds.), Springer, 2008, pp. 921-941

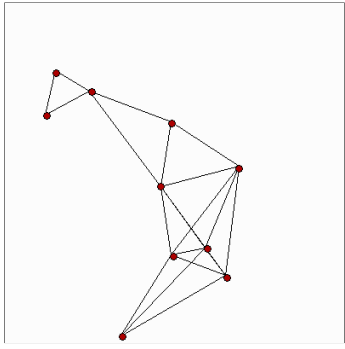
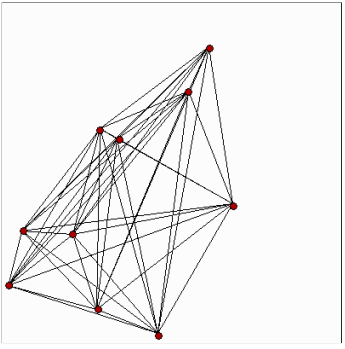
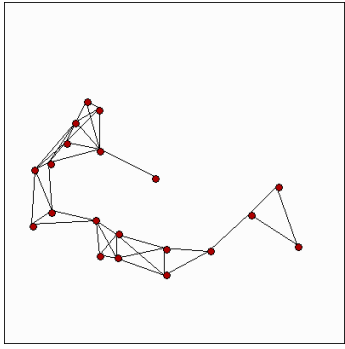
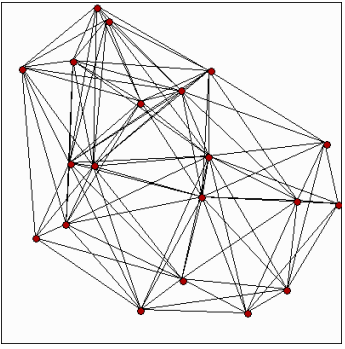
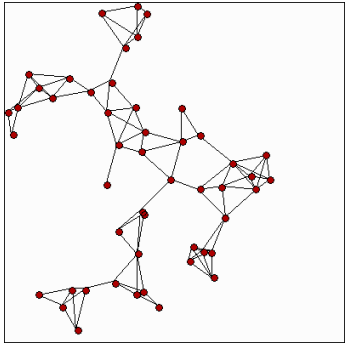
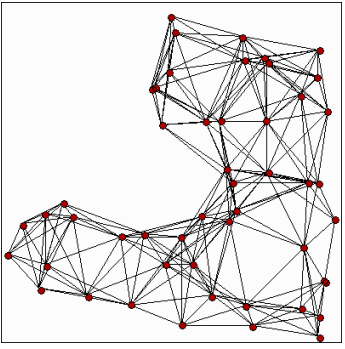
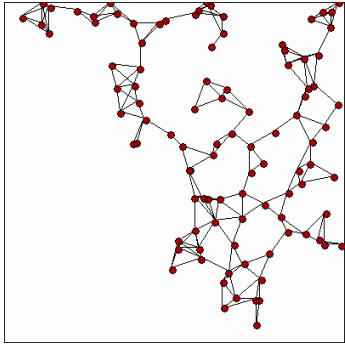
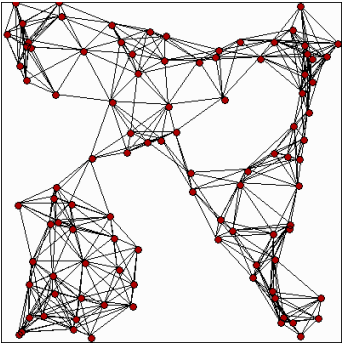
- [P98] L. Parker, "Alliance: An architecture for fault tolerant multirobot cooperation," *IEEE Trans. Robot. Automat.*, vol. 14, no. 2, 1998, pp. 220–240
- [PKL07] J. Partan, J. Kurose, B. N. Levine, „A survey of practical issues in underwater networks“, *SIGMOBILE Mob. Comput. Commun. Rev.*, 11(4), 2007, pp. 23-33.
- [PRK11] S. Parsons, J.A. Rodriguez-Aguilar, M. Klein, „Auctions and bidding: A guide for computer scientists“, *ACM Computing Surveys*, 43(2), pp. A10:1-59, 2011.
- [RG+09] L. Ruiz-Garcia, L. Lunadei, P. Barreiro, J. I. Robla, „A Review of Wireless Sensor Technologies and Applications in Agriculture and Food Industry: State of the Art and Current Trends“, *Sensors*, 2009, pp. 4728-4750.
- [RK+02] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, S. Shenker "GHT: A Geographic Hash Table for Data-Centric Storage," *Proc. Int'l Workshop on Wireless Sensor Networks and Applications (WSNA)*, pp. 78-87, 2002.
- [RV09] E. Ruiz-Ibarra, L. Villasenor-Gonzalez, "Cooperation Mechanism Taxonomy for Wireless Sensor and Actor Networks," *Ad Hoc & Sensor Wireless Networks, An International Journal (ISSN 1551-9899)*, Volume 7, Number 1-2, pp. 91-113, 2009.
- [S05] I. Stojmenović (Ed.), *Handbook of Wireless Sensor Networks: Algorithms and Architectures*, Wiley, 2005.
- [S1] Senslab- eksperimentalna platforma bežičnih senzorskih mreža sa pristupom putem Interneta povezanih i lociranih u pet gradova Francuske, [online] [www.senslab.info](http://www.senslab.info)
- [S81] R.G. Smith, "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver," *IEEE Transactions on Computers*, Vol. C-29, N.12, pp. 1104-1113, 1981.
- [SDM06] K. Selvaradjou, M. Dhanaraj, C. SivaRam Murthy, "Energy Efficient Assignment of Events in Wireless Sensor and Mobile Actor Networks," *Proc. on 14th IEEE International Conference on Networks*, Volume 2, pp. 1-6, 2006.
- [SLJ08] I. Stojmenovic, D. Liu and X. Jia, "A scalable quorum-based location service in ad hoc and sensor networks", *Int. J. Communication Networks and Distributed Systems*, Vol. 1, No. 1, 2008, pp. 71–94
- [SM07] K. Shah, Y. Meng, "Communication-Efficient Dynamic Task Scheduling for Heterogeneous Multi-Robot Systems," *Proc. International Symposium on Computational Intelligence in Robotics and Automation*, pp. 230-235, 2007.
- [SM10] K. Selvaradjou and C. Siva Ram Murthy, "Maximizing Lifetime of Mobile Actors in Wireless Sensor and Actor Networks," *Ad Hoc & Sensor Wireless Networks*, Vol. 9, No. 3-4, pp.179-202, 2010.
- [SMZ07] K. Sohrawy, D. Minoli, T. Znati, *Wireless Sensor Networks: Technology, Protocols, and Applications*, Wiley, 2007.
- [SSG07] P.B. Sujit, A. Sinha and D. Ghose, "Team, Game, and Negotiation based Intelligent Autonomous UAV Task Allocation for Wide Area Applications", *Innovations in Intelligent Machines (1)*, 2007, pp. 39-75
- [SSW05] D. Simplot-Ryl, I. Stojmenovic, and J. Wu, "Energy efficient backbone construction, broadcasting, and area coverage in sensor networks", in: *Handbook of Sensor Networks: Algorithms and Architectures (I. Stojmenovic, ed.)*, Wiley, 2005, pp. 343-379.
- [SW02] P. Stone, M. Veloso, "A Survey on Multiagent and Multirobot Systems", T. Balch and L. E. Parker (Eds.), *Robot Teams: From Diversity to Polymorphism*, A. K. Peters, Wellesley, MA, 2002, pp. 37–92
- [SW04] I. Stojmenovic and J. Wu, "Broadcasting and activity scheduling in ad hoc networks", in: *Mobile Ad Hoc Networking (S. Basagni, M. Conti, S. Giordano and I. Stojmenovic, eds.)*, IEEE/Wiley, 2004, pp. 205-229.
- [SZG09] R. Sarkar, X. Zhu, and Jie Gao. „Double rulings for information brokerage in sensor networks“ *IEEE/ACM Trans. Netw.* 17, 6, 2009, pp. 1902-1915.

- [TLJK05] C. Tovey, M.G. Lagoudakis, S. Jain, S. Koenig, "The Generation of Bidding Rules for Auction-based Robot Coordination," *Multi-Robot Systems: From Swarms to Intelligent Automata*, Vol. III, Springer, pp. 3–14, 2005.
- [TV04] J.B. Tchakarov and N.H. Vaidya, "Efficient Content Location in Wireless Ad Hoc Networks," *Proc. IEEE Int'l Conf. Mobile Data Management (MDM)*, pp. 74-85, 2004.
- [VDMC08] R. Verdone, D. Dardari, G. Mazzini, A. Conti, *Wireless Sensor and Actuator Networks: Technologies, Analysis and Design*, Elsevier Academic Press, 2008.
- [W1] Agregacija paketa, [http://en.wikipedia.org/wiki/Packet\\_aggregation](http://en.wikipedia.org/wiki/Packet_aggregation)
- [W11] Z.D. Wu, "Modelling and analysis of strategies in the design of WSN coordination systems", *Journal of Parallel and Distributed Computing*, Volume 71, Issue 7, pp. 1055-1064, 2011.
- [W96] E. Wolfstetter, "Auctions: An introduction", *Journal of Economic Surveys* 10(4), 1996, pp. 367-420
- [WPT10] Y.C. Wang, W.C. Peng, Y.C. Tseng, Energy-balanced dispatch of mobile sensors in a hybrid wireless sensor networks, *IEEE Transactions on Parallel and Distributed Systems*, 21, 12, pp. 1836-185, 2010.
- [WS+09] T. Wark, D. Swain, C. Crossman, P. Valencia, G. Bishop-Hurley, R. Handcock, "Sensor and Actuator Networks: Protecting Environmentally Sensitive Areas," *Pervasive Computing*, IEEE , vol.8, no.1, pp.30-36, 2009.
- [WT+11] C. Wu, G.S. Tewolde, W. Sheng, B. Xu, Y. Wang, "Distributed Multi-Actuator Control for Workload Balancing in Wireless Sensor and Actuator Networks", *IEEE Transactions on Automatic Control*, vol.56, no.10, pp.2462-2467, 2011.
- [WW11] J. H. Walker and M. S. Wilson, „Task allocation for robots using inspiration from hormones“, *Adaptive Behavior*, June 2011, 19: pp. 208-224.
- [WW98] W.E. Walsh, M.P. Wellman, "A Market Protocol for Decentralized Task Allocation," *Third International Conference on Multi Agent Systems (ICMAS'98)*, pp.325, 1998.
- [WY+10] J. Wu, S. Yuan, S. Ji, G. Zhou, Y. Wang, Z. Wang, „Multi-agent system design and evaluation for collaborative wireless sensor network in large structure health monitoring“, *Expert Systems with Applications*, Volume 37, Issue 3, 2010, pp. 2028-2036
- [XZQD06] Z.Y. Xu, G.S. Zhang, J. Qin, and W. Dou, "Single-Actor Selection Algorithms for Wireless Sensor and Actor Networks", *Wireless Algorithms, Systems, and Applications 2006*, LNCS 4138, 2006, pp. 283 – 294
- [Y06] G. Yang, *Body Sensor Networks*, Springer, 2006.
- [YMG08] J. Yick, B. Mukherjee, D. Ghosal, „Wireless sensor network survey“, *Computer Networks*, Volume 52, Issue 12, pp. 2292-2330, 2008.
- [YP05] Y. Yu, V.K. Prasanna, "Energy-Balanced Task Allocation for Collaborative Processing in Wireless Sensor Networks," *Mobile Networks and Applications*, Vol. 10, Issue 1, pp. 115 – 131, 2005.

# Dodaci

## D1

Tabela D.1 Primeri povezanih pseudo slučajnih grafova mreža izvršilaca generisanih na bazi MIN-DPA algoritma

	$d = 4$	$d = 10$
$N = 10$		
$N = 20$		
$N = 50$		
$N = 100$		

## D2

---

---

### Kompletan globalni simulacioni algoritam za SAP, k-SAP, SAAP, k-SAAP, k-AAP, k-SAPG i k-SAAPG algoritme

---

---

```
1: deklaracija, definicija i inicijalizacija svih promenljivih
2: N_niz = {10, 20, 50, 100} // broj izvršilaca
3: for(i=0;i<4;i++) {
4:     N = N_niz[i];
5:     for(d=4;d<11;d++){
6:         for (k = 1;k<10;k++){
7:             inicijalizacija privremenih promenljivih
8:             for(g=0;g<br_grafova;g++){
9:                 kreiranje povezane pseudo-slučajne mreže po MIN-DPA algoritmu
10:                crtanje grafa i čuvanje u fajlu
11:                inicijalizacija privremenih promenljivih
12:                for (l=0;l<br_ponavljanja;l++){
13:                    ubacivanje jednog događaja na slučajan način
14:                    izračunavanje kvadratne distance od čvorova do događaja
15:                    izbor kolektora probabilističkom metodom
16:                    informaciona faza (za SAP, k-SAP, k-SAAP, k-AAP, k-SAPG, k-SAAPG)
17:                    aukcijska faza (za SAP, k-SAP, k-SAAP, k-AAP, k-SAPG, k-SAAPG)
18:                    faza dodele posla (za SAP, k-SAP, k-SAAP, k-AAP, k-SAPG, k-SAAPG)
19:                    sva računanja i pamćenja merenih veličina u log fajlovima
20:                }//od l
21:            }//od g
22:        }//od k
23:    }//od d
24: }//od i
```

---

---

## D3

---

---

Kompletan globalni simulacioni algoritam za nadgradnju iMesh algoritma upotrebom k-SAP i k-SAAP aukcijskih algoritama

---

---

```
1: deklaracija, definicija i inicijalizacija svih promenljivih
2: N_niz = {10, 20, 50, 100} // broj izvršilaca
3: S_niz = {25, 35, 50, 100} // veličina prostora u metrima
4: for(i=0;i<4;i++) {
5:     N = N_niz[i]; a = S_niz [i];
6:     for(d=4;d<11;d++){
7:         for (k = 1;k<10;k++){
8:             inicijalizacija privremenih promenljivih
9:             for(g=0;g<br_grafova;g++){
10:                 kreiranje povezane pseudo-slučajne mreže izvršilaca po MIN-DPA
                    algoritmu za prvi scenario, odnosno postavljanje mreže na slučajan način
                    za drugi scenario
11:                 kreiranje iMesh informacione strukture
12:                 inicijalizacija privremenih promenljivih
13:                 for (l=0;l<br_ponavljanja;l++){
14:                     ubacivanje jednog događaja na slučajan način
15:                     izračunavanje kvadratne distance od čvorova do događaja
16:                     izbor kolektora cross lookup metodom i početak aukcija
17:                     informaciona faza (za k-SAP i k-SAAP)
18:                     aukcijska faza (za k-SAP i k-SAAP)
19:                     faza dodele posla (za k-SAP i k-SAAP)
20:                     sva računanja i pamćenja merenih veličina u log fajlovima
21:                 }//od l
22:             }//od g
23:         }//od k
24:     }//od d
25: }//od i
```

---

---