

2015

## Multi-agent solutions for task-based resource management under multiple constraints in disaster environments

Xing Su  
*University of Wollongong*

Follow this and additional works at: <https://ro.uow.edu.au/theses>

### University of Wollongong

#### Copyright Warning

You may print or download ONE copy of this document for the purpose of your own research or study. The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site.

You are reminded of the following: This work is copyright. Apart from any use permitted under the Copyright Act 1968, no part of this work may be reproduced by any process, nor may any other exclusive right be exercised, without the permission of the author. Copyright owners are entitled to take legal action against persons who infringe their copyright. A reproduction of material that is protected by copyright may be a copyright infringement. A court may impose penalties and award damages in relation to offences and infringements relating to copyright material.

Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.

Unless otherwise indicated, the views expressed in this thesis are those of the author and do not necessarily represent the views of the University of Wollongong.

### Recommended Citation

Su, Xing, Multi-agent solutions for task-based resource management under multiple constraints in disaster environments, Doctor of Philosophy thesis, School of Computing and Information Technology, University of Wollongong, 2015. <https://ro.uow.edu.au/theses/4496>

# Multi-agent Solutions for Task-based Resource Management under Multiple Constraints in Disaster Environments

A thesis submitted in fulfillment of the  
requirements for the award of the degree

**Doctor of Philosophy**

from

UNIVERSITY OF WOLLONGONG

by

**Xing Su**

School of Computing and Information Technology  
October 2015

© Copyright 2015

by

Xing Su

All Rights Reserved

*Dedicated to*  
*My parents and friends*

# Declaration

This is to certify that the work reported in this thesis was done by the author, unless specified otherwise, and that no part of it has been submitted in a thesis to any other university or similar institution.

---

Xing Su  
October 19, 2015

# Abstract

---

In the last twenty years, disasters throughout the world have become important social and political concerns. In disaster environments, heterogeneous resources with limited energy and capabilities need to be allocated to tasks under multiple constraints. Hence, how to effectively and efficiently manage such resources to complete as many tasks as possible is an important and challenging research topic. This thesis aims at (1) investigating challenging issues of resource management in disaster environments and (2) developing multi-agent approaches to achieve efficient task-based resource management in disaster environments. In the thesis, four multi-agent approaches are proposed and developed to achieve effective and efficient resource management in disaster environments, which are

- a weighted task allocation approach is proposed to enable agents to collect information and make decisions for task allocation in disaster environments under multiple constraints by considering different urgent degree of tasks;
- a dynamic task allocation approach is proposed to enable agents to collect information and form groups for dynamic task allocation in disaster environments under multiple constraints by considering heterogeneity of agents;
- a wireless mobile robot (WR) search and deployment approach is proposed, which can enable WRs with limited energy and sensing capabilities to efficiently collect information and establish ad hoc networks by covering as many tasks in disaster environments as possible; and
- two mathematical programming-based WR deployment approaches are proposed, which can enable a limited number of WRs to establish ad hoc networks by covering the maximum important locations in disaster environments.

# Acknowledgements

---

Studying abroad is a tedious and interesting experience. It would be hard for me to finish my research without the help from many people.

I am indebted to appreciate my supervisors, Prof. Minjie Zhang and Dr. Quan Bai. Their constant commitment and guidance are instrumental in the completion of this thesis, and in making my study a fulfilling experience. I am grateful to Prof. Minjie Zhang for her kind help, encouragement and patient proofreading my thesis and research papers. I am also delighted by Prof. Minjie Zhang's continuous enthusiasm for my everyday life. I would also like to appreciate Dr. Quan Bai's support for my studies. Furthermore, I would like to thank the School of Computing and Information Technology for its technical support and financial support for conference attendance through the years. I also appreciate the University of Wollongong for providing me scholarships of the University Postgraduate Award and International Postgraduate Tuition Award.

My thanks are extended to my colleagues at University of Wollongong, Dr. Fenghui Ren, Dr. Dayong Ye, Dr. Chao Yu, Dr. Ahmed Moustafa, Yan Kong and Dien Tuan Le, who often discussed with me in the laboratory and such discussions have enriched my knowledge. My thanks also go to my friends at Wollongong, especially my roommates, Xiaojun Zhang and Song Liu. They make my life in Wollongong so memorable.

I would like to express my deepest gratitude to my parents, Tieguang Su and Yang Liu, who always make their financial support, encouragement, understanding and love to me. Without their help, this thesis would not be finished. I have dedicated this thesis to my parents and friends for their patience and understanding.

Finally, thanks to all the anonymous reviewers of my research papers, and all my other dear friends and relatives who have supported me.

# Publications

---

The following is a list of my research papers, which have been already published (accepted) or currently are under review, during my PhD study that ends with the completion of this thesis.

## **Refereed Journal Article:**

1. **Xing Su**, Minjie Zhang and Quan Bai, Dynamic Task Allocation for Heterogeneous Agents in Disaster Environments under Time, Space and Communication Constraints. *The Computer Journal*, DOI: 10.1093/comjnl/bxu157, 2015.

## **Scholarly Book Chapter:**

1. **Xing Su**, Minjie Zhang and Quan Bai, Decentralised Task Allocation under Space, Time and Communication Constraints in Disaster Domains. In *Smart Modelling and Simulation for Complex Systems: Practice and Theory, Studies in Computational Intelligence*, pages 41-56, Springer, 2013.

## **Refereed Conference Papers:**

1. **Xing Su**, Minjie Zhang, Quan Bai and Dayong Ye, A Dynamic Coordination Approach for Task Allocation in Disaster Environments under Spatial and Communication Constraints. In *Proceedings of AAAI Workshop on Multi-agent Interaction without Prior Coordination (MIPC 2014)*, Quebec City, Canada, 2014.
2. **Xing Su**, Minjie Zhang and Quan Bai, Task-Based Wireless Mobile Agents Search and Deployment for Ad Hoc Network Establishment in Disaster Environments. In *PRI-CAI2014: Trends in Artificial Intelligence*, pages 1023-1035, Gold Coast, Australia, 2014. (Best paper award)
3. **Xing Su**, Minjie Zhang and Quan Bai, A Wireless Mobile Robots Deployment Approach for Maximising the Coverage of Important Locations in Disaster Rescues. *The 2015 International Conference on Intelligent Agent Technology*, accepted in July 2015 and will appear.



**Submissions:**

1. **Xing Su**, Minjie Zhang and Quan Bai, Coordination for Dynamic Weighted Task Allocation in Disaster Environments with Time, Space and Communication Constraints. *The Computer Journal*, submitted in April 2015.
2. **Xing Su**, Minjie Zhang and Quan Bai, An Innovative Approach for Ad hoc Network Establishment in Disaster Environments by the deployment of Wireless Mobile Agents. *IEEE Transactions on Mobile Computing*, submitted in June 2015.

# Contents

---

<b>Abstract</b>	<b>v</b>
<b>Acknowledgements</b>	<b>vi</b>
<b>Publications</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	2
1.1.1 Special features of disaster environments . . . . .	2
1.1.2 Multi-agent approaches for resource management . . . . .	3
1.1.3 Task-based resource management in disaster environments . . . . .	5
1.2 Challenging Issues for Task-based Resource Management in Disaster Environments . . . . .	6
1.3 The Motivation and Objectives of The Thesis . . . . .	8
1.4 Contributions of The Thesis . . . . .	9
1.5 The Structure of The Thesis . . . . .	10
<b>2 Literature Review</b>	<b>12</b>
2.1 Information Collection in Disaster Environments . . . . .	12
2.1.1 Searching in disaster environments . . . . .	12
2.1.2 Information sharing in disaster environments . . . . .	14
2.2 Task Allocation in Disaster Environments . . . . .	15
2.2.1 Classification of task allocation approaches . . . . .	15
2.2.2 Task allocation approaches for disaster environments . . . . .	17
2.2.3 Testbeds for task allocation in disaster environments . . . . .	24
2.3 Ad hoc Network Establishment in Disaster Environments . . . . .	25
2.3.1 Classification of ad hoc network establishment approaches . . . . .	25

2.3.2	Current approaches for ad hoc network establishment in disaster environments . . . . .	27
2.4	Summary . . . . .	31
<b>3</b>	<b>A Weighted Task Allocation Approach</b>	<b>33</b>
3.1	Problem Description and Definition . . . . .	33
3.2	The Basic Principle of the Coordinated Task Allocation Approach . . . . .	35
3.2.1	Token generation . . . . .	37
3.2.2	Indirected neighbours group formation . . . . .	37
3.2.3	The token passing mechanism . . . . .	40
3.2.4	Weighted task allocation . . . . .	41
3.2.5	Solution return . . . . .	44
3.3	Experiments and Analysis . . . . .	45
3.3.1	Experiment 1: Test of group formation . . . . .	46
3.3.2	Experiment 2: Test of task allocation . . . . .	48
3.3.3	Experiment 3: Test of the impact of urgent degrees of tasks . . . . .	50
3.4	Summary . . . . .	53
<b>4</b>	<b>A Dynamic Task Allocation Approach</b>	<b>54</b>
4.1	Problem Description and Definitions . . . . .	54
4.2	The Principle of the Dynamic Task Allocation Approach . . . . .	57
4.2.1	The information collection mechanism . . . . .	59
4.2.2	The group task allocation mechanism . . . . .	60
4.2.3	The group coordination mechanism . . . . .	65
4.3	Experiments and Analysis . . . . .	70
4.3.1	Experiment 1: test of information collection . . . . .	70
4.3.2	Experiment 2: test of task allocation . . . . .	73
4.3.3	Findings from experiments . . . . .	78
4.4	Summary . . . . .	79
<b>5</b>	<b>A Wireless Mobile Robots Search and Deployment Approach</b>	<b>80</b>
5.1	Definitions . . . . .	80
5.2	The Principle of the Proposed Approach . . . . .	83
5.3	Technical Design of the Search Process . . . . .	85
5.3.1	The search strategy module . . . . .	86
5.3.2	The path creation module . . . . .	87

5.3.3	The movement checking module . . . . .	87
5.4	Technical Design of the Deployment Process . . . . .	88
5.4.1	The deployment location finding module . . . . .	88
5.4.2	The deployment location updating module . . . . .	96
5.5	Experiments and analysis . . . . .	97
5.5.1	Experiment settings . . . . .	97
5.5.2	The results of Experiment 1 . . . . .	99
5.5.3	The results of Experiment 2 . . . . .	100
5.5.4	The results of Experiment 3 . . . . .	102
5.5.5	Findings from experiments . . . . .	103
5.6	Summary . . . . .	104
<b>6</b>	<b>Two Wireless Mobile Robots Deployment Approaches</b>	<b>105</b>
6.1	Problem Description and Definitions . . . . .	105
6.1.1	The NP-hardness and non-linearity of the MILCP . . . . .	106
6.2	Non-greedy Algorithm-based Approaches vs. Greedy Algorithm-based Approaches . . . . .	108
6.3	The Linear Programming-based Approach for the MILCP . . . . .	110
6.3.1	The finding of potential deployment locations . . . . .	110
6.3.2	The construction of the covering relationship matrix . . . . .	112
6.3.3	The LP formulation of the MWRDP . . . . .	113
6.3.4	The proposed LP-based approach . . . . .	113
6.3.5	The computational complexity of the LP-based approach . . . . .	115
6.4	The QP-based approach for the MILCP . . . . .	116
6.4.1	The construction of the duplicated covering matrix . . . . .	116
6.4.2	The quadratic programming formulation of the MILCP . . . . .	118
6.4.3	The computational complexity of the QP formulation . . . . .	119
6.4.4	The extension of the QP-based approach for the MILCP . . . . .	119
6.5	Experiments and Analysis . . . . .	121
6.5.1	The experimental settings . . . . .	121
6.5.2	The results and analysis of Experiment 1 . . . . .	122
6.5.3	The results and analysis of Experiment 2 . . . . .	123
6.5.4	Findings from experiments . . . . .	124
6.6	Summary . . . . .	125

<b>7 Conclusion and Future Work</b>	<b>126</b>
7.1 Contributions of the Thesis . . . . .	126
7.2 Future Work . . . . .	128
<b>Bibliography</b>	<b>131</b>

# List of Tables

---

2.1	The comparison of different categories of task allocation and ad hoc network establishment approaches . . . . .	31
3.1	The settings of Experiment 1 . . . . .	46
3.2	The settings of Experiment 2 . . . . .	48
3.3	The settings of Experiment 3 . . . . .	51
4.1	The setting of Experiment 1 . . . . .	70
4.2	The settings of Experiment 2 . . . . .	73
5.1	The settings of three experiments . . . . .	98
6.1	The experimental settings . . . . .	122

# List of Figures

---

3.1	The five steps of the proposed approach in one loop . . . . .	36
3.2	The difference between INGF and DNGF . . . . .	38
3.3	The example of token passing . . . . .	41
3.4	The information of tasks and status of agents . . . . .	43
3.5	The results of Experiment 1 . . . . .	47
3.6	The results of Experiment 2 . . . . .	49
3.7	Sum workload . . . . .	52
3.8	Sum weighted workload . . . . .	52
4.1	The communication relationships among agents . . . . .	56
4.2	The basic process of the dynamic task allocation approach . . . . .	57
4.3	Locations of tasks before employing the mean-shift algorithm . . . . .	61
4.4	Groups of tasks after employing the mean-shift algorithm . . . . .	62
4.5	The two-layer group coordination mechanism . . . . .	67
4.6	The results of Experiment 1 . . . . .	71
4.7	The results of Experiment 2 . . . . .	75
4.8	The time spent on computation for task allocation . . . . .	76
4.9	The amount of information exchange for task allocation . . . . .	77
5.1	The guidance locations of tasks . . . . .	82
5.2	The principle of the proposed approach . . . . .	83
5.3	An example of DLF-TP based on Proposition 5.1 . . . . .	89
5.4	An example of DLF-TP based on Proposition 5.2 . . . . .	90
5.5	The coverage locations of a task . . . . .	94
5.6	The results of Experiment 1 . . . . .	99
5.7	The coverage of tasks in Experiment 2 . . . . .	100
5.8	The coverage of the environment (i.e., locations) in Experiment 2 . . . . .	101
5.9	The objective values of ad hoc networks in Experiment 2 . . . . .	102

5.10	The results of Experiment 3 . . . . .	103
6.1	An example of WAs deployment . . . . .	108
6.2	The WR deployment between two kinds of approaches . . . . .	109
6.3	The maximum difference of the IL coverage between WRs deployed by two kinds of approaches . . . . .	109
6.4	The potential deployment locations found based on Proposition 6.3 . . . . .	111
6.5	The distribution of ILs in the environment . . . . .	121
6.6	The maximum number of ILs covered by WRs . . . . .	122
6.7	The number of computations used by LP and QP . . . . .	124



# Chapter 1

---

## Introduction

Nowadays, disasters throughout the world have become important social and political concerns. In a disaster environment, many tasks need to be completed by different resources (i.e., human resources, mechanical equipments, robots, etc.). Hence, resource management in disaster environments has become an important research issue. Different from resource management in other environments, there are a number of special features in disaster environments, which need to be considered during resource management, such as communication constraints and obstacles of environments, time and space constraints of tasks, limited and heterogeneous capabilities of resources, etc. How to properly handle these constraints is a key to achieve efficient resource management in disaster environments.

In the last twenty years, Multi-Agent Systems (MASs) have attracted much attention from researchers in computer science, information technology, engineering, as well as other disciplines, due to their abilities of autonomous learning [146], [13], [39], [1], independent decision making [12], [152], [60], collaborative problem solving [67], [68], [3], [6], as well as automatic self-adaption. These abilities are particularly attractive for the applications of homogenous or heterogeneous autonomous agents in open and dynamic environments [91], [54], [118]. Multi-agent technologies have been successfully applied to different domains, such as grid computing [41], [139], [22], [153], pervasive computing [120], [104], [129], E-markets [30], [66], [118], etc.

Multi-agent approaches have become very important solutions to solve many challenging issues for resource management in disaster environments. For example, Shehory and Kraus proposed a multi-agent based coalition formation approach [130]. Since their approach considers the situations of complex tasks and small agents in disaster environments, it has better performance on coalition formation in disaster environments than many non-agent based approaches. Shehory et al. proposed a multi-agent based task allocation approach [131], which considers the communication constraints of disaster environments and employs decentralised task allocation algorithms. It is proved that Shehory et al.'s approach has better performance on task allocation than theoretical (i.e., non-agent based) approaches in disaster

environments under communication constraints. Scerri et al. proposed a multi-agent based task allocation approach [125], which considers the large amount of heterogeneous agents in disaster environments and employs token-based task allocation algorithms. Scerri et al.'s approach has better performance than other non-agent based approaches on task allocation in disaster environments with a large amount of heterogeneous agents.

The *main purposes* of this thesis are to

- investigate the challenging issues of resource management in disaster environments; and
- develop multi-agent approaches to achieve efficient task-based resource management in disaster environments.

The reminder of this chapter is organized as follows. Section 1.1 introduces some background knowledge about this thesis. Section 1.2 introduces challenging issues of resource management in disaster environments. Section 1.3 introduces the motivation and objectives of the thesis. Section 1.4 introduces contributions of the thesis. Section 1.5 gives the structure of the thesis.

## 1.1 Background

In this section, some background knowledge about this thesis is introduced. Subsection 1.1.1 focuses on introducing the special features of disaster environments. Subsection 1.1.2 mainly introduces applications of multi-agent approaches for resource management.

### 1.1.1 Special features of disaster environments

In the last twenty years, disasters, such as the Hurricane Katrina [142], Indian Ocean Earthquake [23], 2008 Sichuan Earthquake [10], etc. have attracted much attention from researchers all over the world. After disasters happened, many tasks, such as saving survivors in debris, extinguishing fire of buildings need to be completed by different resources, such as rescuer teams, mechanical devices, robots, etc. In disaster environments, a number of special features of environments and tasks need to be considered during resource management.

*Major constraints* in disaster environments:

- **Unknown and complexity of the environment**

An environment can be changed greatly by a disaster. Even if we have knowledge

about the environment before the disaster, it is hard to know or predict the circumstances in the environment after the disaster [97]. In addition, ubiquitous obstacles in a disaster environment could block the path and occupy some locations of tasks [103].

- **Communication constraints**

Due to the destruction of local communication infrastructures, the communication for resource management mainly relies on portable wireless communication devices. The wireless communication technologies have two main drawbacks [72], [112], which are communication capacity and range constraints. The communication capacity constraint indicates that the amount of information that can be transferred between two devices is limited. The communication range constraint indicates that the distance between two communicable devices is limited.

- **Open and dynamic environments**

In disaster environments, resources can continuously enter and leave an environment and tasks can be continuously discovered and finished in the environment [133], [21].

The *characteristics of tasks* include:

- **Time constraints**

In disaster environments, tasks include saving survivors in debris, extinguishing fire of buildings, etc. In such circumstances, each task should have a hard deadline (i.e., the time point until which a survivor is still alive or the building is still standing) and a task is worthy to be completed before its deadline [113], [37], [48].

- **Space constraints**

In disaster environments, many tasks are stationary and need to be performed on the sites so that if a resource is requested by a task, it first needs to move to the location of the task, which will consume a certain amount of time [79], [9], [113].

- **Different urgent degrees of tasks**

In disaster environments, different tasks may have different urgent degrees [111], [76]. The tasks with higher urgent degrees need to be completed first, while tasks with lower urgent degrees should be disregarded if resources are not sufficient.

### 1.1.2 Multi-agent approaches for resource management

Multi-agent approaches for resource management have been studied from late last century. Multi-agent resource management aims to effectively and efficiently manage different kinds

of resources in different areas [58], [65], [144], [98], such as market environments [19], multiprocessor environments [87], robotic applications [45], disasters [76], etc. In different applications, the objectives of resources management are different.

- **Resource management in market environments**

In market environments, [59], [71], [24], sellers and buyers are resource managers, which trend to maximise their profits through trading their resources (i.e., products and money). Therefore, each resource manager in a market environment is self-interest and the relationships between the same kind of resource managers (i.e., all sellers or all buyers) are competitive.

- **Resource management in multiprocessor environments**

In multiprocessor environments [86], [114], [53], a controller centrally manages processors in a system, which aims to make the system to work properly and efficiently.

- **Resource management in robotic applications**

In robotic applications [16], [36], [85], robots, such as mobile robots, unmanned aerial vehicles (UAVs), etc can only carry limited resources (i.e., such as gasoline, battery, etc). Therefore, in robotic applications, efficient management of limited resources of robots is the main objective to extend the working time of robots.

- **Resource management in disaster environments**

In disaster environments, all different resources are heterogeneous and autonomous agents, which form a Multi-Agent System and cooperatively perform tasks. The main objective of resource management in disaster environments is to maximise the number of tasks completed on time (i.e., before their deadlines) in the environments [26], [27], [33]. However, due to the constraints in the environments and the characteristics of tasks (see Subsection 1.1.1), the following aspects need to be considered in the design of multi-agent approaches for such applications.

- **Local views**

In most disaster environments, due to the space and communication constraints as well as unknown and complexity of environments, it is hard for an agent to have global knowledge about the environments. Therefore, agents make decisions for resource management in disaster environments only based on their local views [76], [126].

- **Heterogeneity**

In disaster environments, agents in MASs are heterogeneous, which means that

different agents have different capabilities and work efficiencies to perform different tasks [79], [7]. When allocating heterogeneous agents to tasks, the capabilities of agents need to be considered. For example, first aiders are good at treating injured people and fire fighters are good at extinguishing fires. It is very inefficient to assign tasks of extinguishing fire to first aiders or tasks of treating injured people to fire fighters.

– **Limited resources**

Due to blocked roads and distributed transport facilities of an environment, at the beginning, only a limited number of resources can enter the environment to participate in the disaster rescue [138], [93], which might be much less than the number of tasks in the environment. Therefore, it is hard or even impossible for the limited number of resources to complete all tasks in a disaster environment under requested deadlines.

– **Limited energy and capabilities of resources**

In disaster environments, due to the destruction of local infrastructures, agents can only carry limited energy (gasoline, battery, etc) [156], [2]. When their energy is exhausted, agents will lose corresponding capabilities. In addition, even if agents have energy, the corresponding capabilities might also be limited. For example, it is hard for a first aider to treat hundreds of injured people or a fire fighter extinguish fire of a whole building.

### 1.1.3 Task-based resource management in disaster environments

In the last twenty years, many multi-agent approaches were proposed for resource management in disaster environments from different perspectives, which includes routing approaches [70], [95], locating approaches [49], [115], task allocation approaches [112], [101], ad hoc network establishment approaches [55], [18], evacuating approaches [141], [102], information sharing approaches [42], etc. This thesis mainly focuses on the study of task-based resource management.

Nowadays, wireless mobile robots, which are a kind of agents in disaster environments, have played an important role and become a main technique in task-based resource management in disaster environments, due to their collaborative problem solving, low infrastructure dependence, quick adaptability and scalability, etc, which are suitable for them to cooperatively work in disaster environments. However, due to special features of disaster environments, tasks and agents, there are eight main challenging issues to be handled when

designing multi-agent approaches for task-based resource management in disaster environments. These issues are given in the next section.

## 1.2 Challenging Issues for Task-based Resource Management in Disaster Environments

In disaster environments, the main parts for task-based resource management (i.e., task allocation and ad hoc network establishment) include **i) information collection**, **ii) task allocation** and **iii) ad hoc network establishment**.

- **Information collection**

In task-based resources management, if agents want to make good decisions, they need to collect comprehensive information about the environments. However, due to the special features of disaster environments and limited energy and capabilities of agents, it is hard for an agent to collect such information. To efficiently collect information in disaster environments, the following challenging issues must be solved.

- **Challenging issue 1:** Planning the searching path

In disaster environments, agents should be able to dynamically create searching pathes in the unknown and complex env an collect as much information as possible within its energy and capability limitations.

- **Challenging issue 2:** Sharing Information

In multi-agent systems, information sharing among agents is also an important and popular way for information collection. However, due to communication constraints, an agent in a disaster environment can only share limited information with limited number of other agents in the environment.

- **Task allocation**

Based on the collected information (i.e., local views), agents make decisions for task allocation to achieve different objectives. In disaster environments, task allocation approaches trends to maximise the number of tasks that can be completed by agents on time. To achieve such the objective, agents have to handle a number of challenging issues caused by special features of environments and tasks as well as their different and limited capabilities.

- **Challenging issue 3:** Handling multiple constraints in open and dynamic environments

During task allocation, agents should take their limited energy and capabilities, openness and dynamics of environments as well as time and space constraints of tasks into account to ensure that the allocated tasks can be completed.

- **Challenging issue 4:** Distinguishing different urgent degrees of tasks

During task allocation, tasks should be classified according to their urgent degrees. Tasks with higher urgent degrees need to be completed first, while tasks with lower urgent degrees should be delayed or disregarded, if agents are not sufficient in disaster environments.

- **Challenging issue 5:** Matching different capabilities of agents with tasks

During task allocation, the requirements of tasks should match the capabilities of agents, such as allocating first aiders to treat injured people and fire fighters to extinguish fires.

- **Ad hoc network establishment**

Based on the collected information (i.e., local views), wireless mobile robots (WRs) have to find suitable deployment locations to establish ad hoc networks in disaster environments. To increase the work efficiency of the established ad hoc network, the following challenging issues should be considered.

- **Challenging issue 6:** Maximising the coverage of important locations and the environment

In a disaster environment, the distribution of important locations (ILs) (i.e., locations of tasks, victims, etc.) is uneven. In order for most of ILs can be reached by suitable rescue resources (i.e., rescuer teams, mechanical devices, robots, etc.), the established ad hoc network should be able to cover the maximum number of ILs in the environment. In addition, due to the unpredictable mobility of rescue resources, the established ad hoc network should be able to cover the maximum areas of disaster environments so as to increase the opportunities to guide these resources.

- **Challenging issue 7:** Enabling the communication of agents in ad hoc networks

In an established ad hoc network, agents with limited sensing and communication range should be communicable. By doing so, the information about ILs and rescue resources can be shared among agents within the entire network. In addition, the work efficiencies of the ad hoc networks can be improved.

- **Challenging issue 8:** Maximising the effectiveness of limited number of agents

At the beginning of disaster rescues, due to the blocking of roads, only a limited

number of agents can enter disaster environments to establish ad hoc networks. In such situation, the number of agents is much less than the number of ILs in the disaster environments. Therefore, how to deploy the limited number of agents to maximise their effectiveness and efficiency is an important and challenging issue.

## 1.3 The Motivation and Objectives of The Thesis

The *motivation* of this thesis is to solve challenging issues listed in Section 1.2 by developing multi-agent approaches to achieve efficient task-based resource management in disaster environments. The *objectives* of the thesis are listed as follows.

- **Objective 1:** Effective and efficient path planning  
Developing path planning mechanisms to help agents with limited energy and searching capabilities to efficiently search in unknown environments by avoiding ubiquitous obstacles.
- **Objective 2:** Information sharing under communication constraints  
Developing information sharing mechanisms to help agents with limited communication capabilities to efficiently share information and communicate with each other in environments under communication constraints.
- **Objective 3:** Dynamic task allocation by considering multiple constraints  
Developing dynamic task allocation mechanisms to allocate suitable agents to tasks by considering limited energy and capabilities of agents, openness and dynamics of environments as well as time and space constraints of tasks.
- **Objective 4:** Task allocation by considering different urgent degrees of tasks  
Developing task allocation mechanisms to allocate suitable agents to tasks by considering different urgent degrees of tasks.
- **Objective 5:** Task allocation by considering different capabilities of agents  
Developing task allocation mechanisms to allocate suitable agents to tasks by taking different requirements of tasks and capabilities of agents into account.
- **Objective 6:** Ad hoc network establishment with the maximum coverage  
Developing wireless mobile robot (WR) deployment mechanisms for ad hoc network establishment, which should be able to maximise ILs and areas of environments covered by the ad hoc network established by WRs.



- **Objective 7:** Ad hoc network establishment with the communication of agents  
Developing WR deployment mechanisms for ad hoc network establishment, which should enable WRs in the established ad hoc network to communicate with each other.
- **Objective 8:** Ad hoc network establishment with the limited number of agents  
Developing WR deployment mechanisms for ad hoc network establishment, which should be able to maximise the utility of the ad hoc network established by the limited number of WRs.

## 1.4 Contributions of The Thesis

The *major contributions* of the thesis are listed as follows.

- **A coordinated approach is proposed for weighted task allocation in disaster environments (to achieve Objectives 2, 3 and 4)**

The proposed coordinated task allocation approach includes three mechanisms, which are 1) a group formation mechanism is employed by agents to form temporary groups and select a coordinator for each group by considering limited communication capabilities of agents in disaster environments (to achieve **Objective 2**); 2) a token passing mechanism is employed by group members to pass information for task allocation to coordinators of groups by considering communication constraints in disaster environments (to achieve **Objective 2**); and 3) a utility calculation mechanism is employed by coordinators to create suitable task allocation solutions for their groups by considering multiple constraints (to achieve **Objectives 3**) as well as different urgent degrees of tasks in disaster environments (to achieve **Objectives 4**).

- **A dynamic task allocation approach is proposed for heterogeneous agents by group formation (to achieve Objectives 2, 3 and 5)**

The dynamic task allocation approach includes three mechanisms, which are 1) an information collection mechanism is employed by agents to prune their connections in communication networks and select network leaders by considering limited communication capabilities of agents in disaster environments (to achieve **Objective 2**); 2) a group task allocation mechanism is employed by network leaders to allocate tasks and agents of their networks to isolated groups by considering space and communication constraints (to achieve **Objective 3**) and different capabilities of agents in disaster

environments (to achieve **Objective 5**); and 3) a group coordination mechanism is employed by groups to periodically coordinate at assembly points by considering dynamics and communication constraints of disaster environments (to achieve **Objectives 3**).

- **A wireless mobile robot (WR) search and deployment approach is proposed for ad hoc network establishment in disaster environments (to achieve Objectives 1, 2, 6 and 7)**

The proposed approach consists of a search process and a deployment process. The search process enables WRs with limited energy and (sensing and communication) capabilities to efficiently collect information in unknown and complex disaster environments (to achieve **Objective 1**); The deployment process finds suitable deployment locations for WRs to establish ad hoc networks, which can cover as many important locations (ILs) and areas in disaster environments (to achieve **Objective 6**) and enable the communication of deployed WRs in the established networks (to achieve **Objective 7**) by considering limited sensing and communication capabilities of WRs (to achieve **Objective 2**);

- **Two mathematical programming-based WR deployment approaches are proposed for disaster rescues (to achieve Objectives 2, 6 and 8)**

The two approaches are a linear programming (LP)-based WR deployment approach and a quadratic programming (QP)-based WR deployment approach. The LP-based approach can find suitable deployment locations for a limited number of WRs (to achieve **Objective 8**) to cover the maximum ILs and areas (to achieve **Objective 6**) by considering multiple constraints of disaster environments and limited sensing and communication capabilities of WRs (to achieve **Objective 2**); The QP-based approach can create the same deployment locations for the WRs as the LP-based approach with less computational complexity.

## 1.5 The Structure of The Thesis

The rest of the thesis is organised as follows.

**Chapter 2** reviews current approaches on task allocation and ad hoc network establishment in disaster environments.

**Chapter 3** introduces a coordinated approach for dynamic weighted task allocation in disaster environments under time, space and communication constraints.

**Chapter 4** proposes a task allocation approach for heterogeneous agents in disaster environments under space and communication constraints.

**Chapter 5** introduces a task-based WR search and deployment approach for ad hoc network establishment in disaster environments.

**Chapter 6** introduces two mathematical programming-based WR deployment approaches for the ad hoc network establishment.

**Chapter 7** concludes the thesis and outlines future directions of this research.

# Chapter 2

---

## Literature Review

The main purpose of this thesis is to study and solve challenging issues of task-based resource management (which were identified in Section 1.2) in disaster environments. As introduced in Subsection 1.2, these challenging issues mainly exist during **i) information collection**, **ii) task allocation** and **iii) ad hoc network establishment**. In this chapter, current approaches in these three parts are reviewed in detail. In particular, Section 2.1 introduces approaches related to information collection in disaster environments; Section 2.2 introduces approaches related to task allocation in disaster environments; and Section 2.3 introduces approaches related to ad hoc network establishment in disaster environments.

### 2.1 Information Collection in Disaster Environments

In order for agents to make optimal decisions on task-based resource management, the main objective of information collection is to enable agents with limited energy and capabilities to collect as much information in disaster environments as possible. As introduced in Subsection 1.2, in disaster environments, the information collected by agents comes from two aspects, which are the information collected from searching and from other agents. In this section, approaches related to the two aspects of information collection are introduced in detail in the following two subsections.

#### 2.1.1 Searching in disaster environments

After disasters happened, agents with limited energy and capabilities have to search rescue tasks (i.e., locations of victims, fires, etc) in disaster environments. In many disasters, due to the scale of disaster environments and limited energy and capabilities of agents, it is hard or even impossible for agents to search all locations of the environment. Hence, how to efficient search in disaster environments has become an important and challenging issue for agents.

A good search approach should include two components: 1) an efficient search strategy

and 2) a path planning algorithm. Search strategies enable agents to find as many tasks in disaster environments as possible without prior information about the environments. Path planning mechanisms enable agents to find the shortest pathes to move to locations of environments by avoiding obstacles.

- **The search strategies**

Reich and Sklar [117] developed a search and rescue approach for robots (i.e., agents). They assumed that there are sufficient robots in a disaster environment, so robots search tasks only based on the blind search strategy. However, at the early stage of the disaster response, only a limited number of robots can enter the environment to participate search and rescue. Hence, the blind search strategy is not an effective and efficient strategy for searching in disaster environments. Koenig et al. [78] introduced the greedy mapping method for robots to search in unknown environments. Based on the greedy mapping method, robots always move to the closest locations that they have not visited yet. According to the analysis of Koenig et al., the greedy mapping method can effectively and efficiently reduce travelling distances of robots in disaster environments. Sariel and Akin [124] proposed a search strategy for autonomous robots. In Sariel and Akin's search strategy, a robot always trends to search the fastest unsearched locations in a disaster environment. By doing so, if a robot has enough energy, it can search all locations of the disaster environment. However, due to limited energy and capabilities of robots, it is not applicable to employ Sariel and Akin's search strategy to search in large-scale disaster environments.

- **The path planning methods**

The A\* search algorithm [107], [31], [150] is a common and popular path planning algorithm in disaster environments, which employs the grid map and heuristics to find the shortest pathes for robots by avoiding obstacles. However, the A\* search algorithm has two main drawbacks, which limit its application in disaster environments. First, the path planning in the A\* search algorithm is based on the grid map of environments, which is hard to be applied in unknown disaster environments. Second, the path planning in the A\* search algorithm is based on heuristics so that the computational complexity of the A\* search algorithm increases exponentially when the size of planning areas increases. Nilsson [106] introduced the visibility graph method for path planning in complex environments. The visibility graph method [147], [64] can find shortest pathes and avoid obstacles for robots through connecting locations of robots and visible vertices of obstacles so that a path created by the visibility graph method is always combined by several straight sub-pathes. However, the visibility graph method

has also two drawbacks to limit its applications in disaster environments. First, the path planning in the visibility graph method is based on the information of obstacles in a disaster environment. In addition, the computational complexity of the visibility graph method is proportional to the number of obstacles in the area. Latombe [88], introduced the Voronoi diagram based method for path planning. The Voronoi diagram [74], [5] is a popular tool for plane partitions. Based on the Voronoi diagram, if obstacles are modeled as points, pathes in a disaster environment can be found, which can keep the maximum distances with all obstacles in the environment. The same as the the visibility graph method, the creation of the Voronoi diagram requires the information of obstacles in disaster environment. In addition, the Voronoi diagram path planning method can only find the safest pathes rather that the shortest pathes for robots in disaster environments.

### 2.1.2 Information sharing in disaster environments

Information sharing is also an important and popular way for information collection in MASs. In disaster environments, due to the destruction of local communication infrastructures, an agent has to share information with other agents under communication capacity and range constraints. The communication capacity constraint limits the amount of information that agents can be transmitted at a time. The communication range constraint limits the distance between two communicable agents. Under the communication range constraint, all agents in a disaster environment naturally form a communication network based on their locations and communication distances. In disaster environments, the information sharing aims to enable agents to effectively and efficiently exchange information for task allocation in a decentralised manner through the communication network. In order to achieve this purpose, many decentralised information sharing approaches have been proposed from different perspectives.

Theocharopoulou et al. [140] proposed a task allocation approach for agents in a large-scale network. In their approach, some agents in the network are selected as gateways, which are responsible for the information transmission for a group of agents through maintaining routing indices of them. Based on gateways and routing indices, information can be efficiently and accurately transmitted among agents in a static and large-scale communication network. However, due to the dynamics of disaster environments, communication networks continuously change. Since gateways cannot always be able to communicate with agents of their groups, Theocharopoulou et al.'s approach does not work well in highly dynamic disaster environments.

Wang et al. [145] proposed a distributed task allocation approach for robots search and rescue. In their approach, all information related to task allocation, such as resource allocation, information of tasks and information of agents, etc, are represented by tokens, which can be exchanged among agents through the communication network. Based on Wang et al.'s approach, the amount of information transferred among agents in an environment can be significantly reduced, which has proved that their approach is suitable for agents to share information under communication constraints.

Farinelli et al. [38] developed a max-sum-based approach for task allocation. In Farinelli et al.'s approach, agents are assumed to be cooperative so as to maximise the benefit of all agents in an environment. Rather than exchanging tokens for information sharing, agents in Farinelli et al.'s approach exchange the objective function through the communication network. In the objective function, many factors correspond to different agents so that an agent can collect information of other agents by the values of their factors in the objective function. Based on Farinelli et al.'s approach, the communication requested for task allocation among agents can also be reduced.

## 2.2 Task Allocation in Disaster Environments

Based on the collected information, agents have to make decisions on task allocation. Task allocation is a classic and popular research topic, which can be applied to many areas, such as manufactories, vehicles, CPU, etc., [143], [47], [149], [89], [108], [82], [11]. Task allocation in disaster environments refers to assigning a number of available agents to a number of tasks, during which, special features of disaster environments, tasks and agents should be considered (i.e., see Section 1.1). In this section, approaches related to task allocation in disaster environments are introduced in detail. In particular, the classification of task allocation approaches are introduced at first. Then, task allocation approaches for disaster environments are introduced. Finally, some testbeds that can evaluate the performance of task allocation approaches in disaster environments are introduced.

### 2.2.1 Classification of task allocation approaches

The task allocation approaches aim to create suitable assignments between tasks and agents in disaster environments. Current task allocation approaches were classified by different researchers from different perspectives.

Ramchurn et. al [113] classified task allocation approaches into centralised, decentralised and hybrid approaches, based on who creates task allocation solutions.

- **Centralised task allocation approaches**

In a centralised task allocation approach, there is a central controller in charge of creating task allocation solutions for all agents in a disaster environment. In order to create suitable task allocation solutions, the central controller should be able to freely communicate with all agents in the environment so as to timely collect information from them.

- **Decentralised task allocation approaches**

In decentralised task allocation approaches, there is no central controller so that agents in a disaster environment have to collect formation and create solutions for task allocation by themselves. Different from centralised task allocation approaches, decentralised task allocation approaches do not highly depend on the communication among agents, which makes these approaches more suitable for agents to work under communication constraints.

- **Hybrid task allocation approaches**

In hybrid task allocation approaches, agents in a disaster environment form a number of coalitions. In each coalition, there is a controller in charge of creating task allocation solutions for coalition members (i.e., agents). The architecture of the controller and agents in a coalition is centralised, while the architecture of coalitions in the environment is decentralised. Therefore, the hybrid task allocation approaches could have features of both centralised and decentralised task allocation approaches.

Gerkey and Mataric [45] proposed a taxonomy for task allocation approaches based on three criteria.

- **Single agent vs. Multiple agents**

Single agent: task allocation approaches can only assign one agent to perform one task.

Multiple agents: task allocation approaches can assign multiple agents to perform one task.

- **Single task vs. Multiple tasks**

Single task: In task allocation approaches, one agent can only perform one task at a time.

Multiple tasks: In task allocation approaches, one agent can perform multiple tasks at the same time.



- **Instantaneous vs. Time-extended**

Instantaneous: task allocation approaches assign agents to tasks at one time.

Time-extended: task allocation approaches continuously assign available agents to unfinished tasks until there is no task in the environment.

### 2.2.2 Task allocation approaches for disaster environments

From the last century, disasters have become important social and political concerns. It was found that suitable task allocation approaches can improve work efficiencies of agents so as to enable them to complete more tasks in disaster environments. In addition, during task allocation, special features of disaster environments, features of tasks and features of agents should be considered. In this subsection, several famous task allocation approaches for disaster environments are reviewed in detail based on the classification by Ramchurn et al. [113].

#### Centralised task allocation approaches

In centralised task allocation approaches, the central controller is in charge of collecting information and creating solutions for task allocation for all agents. In large-scale disasters, such as the Indian Ocean Earthquake [110], Hurricane Katrina [35], etc, the number of tasks and the number of agents in disaster environments can be large so that how to create optimal solutions for task allocation by considering special features of environments, tasks and agents in disasters is the primary concern of central controllers in this type of approaches.

In order to create optimal solutions for task allocation, some researchers employed the Mixed Integer Linear Programming (MILP) for task allocation in disaster environments. Koes et al. [79] proposed a MILP-based task allocation approach for robots, which considers time and space constraints of tasks as well as different capabilities of agents. In their approach, the time of an agent to perform a task is divided into three stages. Based on accurate information of the three time stages of tasks and information of agents, the MILP formulates task allocation solution in a disaster environment. Koes et al.'s approach can guarantee to find the optimal solution for task allocation.

Ramchurn et al. [113] also developed a MILP-based coalition formation approach for task allocation in disaster environments. Their approach also considers time and space constraints of tasks. Different from Koes et al.'s approach, Ramchurn et al.'s approach improved the work efficiencies of agents through forming coalitions. Ramchurn et al.'s approach also

can find optimal solutions for coalition formation and task allocation in disaster environments. Although MILP-based approaches can guarantee optimal solutions for task allocation in disaster environments, it is proved that the process of creating solutions based on the MILP is NP-hardness (the proof can be found in [113]), which means that a central controller has to spend a huge amount of time to find the optimal solution for task allocation in a disaster environment. As authors mentioned in paper [113], ‘For small scenarios with not more than 4 agents and 7 tasks, the MILP-based algorithm takes more than 2 hours to find the optimal solution.’ In order to simplify the process of creating task allocation solutions, Ramchurn et al. also developed a myopic approach for task allocation in disaster environments in [113]. In their myopic approach, a long-term task allocation problem is divided into a number of short-term task allocation problems. In each short-term task allocation problem, the central controller finds the optimal solution for task allocation based on three heuristic steps so as to achieve three objectives, which are

- maximising the number of completed tasks;
- maximising the working time of agents; and
- minimising the time taken by agents to complete tasks.

Based on the Ramchurn et al.’s myopic approach, the central controller can find near-optimal solutions for task allocation in large-scale disaster environments without sophisticated calculation.

From above introductions, it can be seen that centralised task allocation approaches have following **advantages**.

- **Optimal or near-optimal solutions**

In centralised task allocation approaches, central controllers can create optimal or near-optimal solutions for tasks and agents in disaster environments by considering special features of environments, tasks and agents.

- **Fast information collection**

Since the central controller is responsible for information collection, agents in a disaster environment only need to transmit their collected information to the central controller, which can significantly reduce the time for information collection.

- **Less conflicts**

Since only the central controller is responsible for creating task allocation solutions for all agents in a disaster environment, there are very few conflicts among agents in centralised task allocation approaches.

However, centralised task allocation approaches have following **disadvantages**, which limit their applications in disaster environments.

- **High communication dependence**

Optimal or near-optimal task allocation solutions are created by central controllers based on the comprehensive information timely collected from agents. Therefore, centralised task allocation approaches are vulnerable to communication constraints in disaster environments.

- **High computation complexity**

In centralised task allocation approaches, all information of tasks and agents in a disaster environment is collected by the central controller. In order to create optimal or near-optimal solutions for these tasks and agents, the central controller needs a large amount of time and resources for solution computation.

### **Decentralised task allocation approaches**

In decentralised task allocation approaches, agents have to collect information and make decisions for task allocation by themselves. In order to make suitable decisions on task allocation, agents have to collect comprehensive information in disaster environments. Due to special features of environments and limited energy and capabilities of agents, it is hard for agents to collect such information through searching so that information sharing among agents through the communication network is an important way for information collection in disaster environments. In the last twenty years, many information sharing methods (i.e., see Subsection 2.1.2) were employed by decentralised task allocation approaches in disaster environments so as to enable agents to exchange information for task allocation under communication constraints.

Some researchers employed “token-based” mechanisms to exchange information for task allocation in large-scale disaster environments. Scerri et al. [125] proposed a threshold-based task allocation approach for disaster rescues. In Scerri et al.’s approach, tasks are represented by tokens and can be exchanged among agents through the communication network. When an agent receives a task token, the agent has to make the decision on performing the task of the token or passing the token to another agent based on the threshold of the task in the token. Based on the token mechanism, the communication for task allocation among agents in large-scale disaster environments can be greatly reduced. However, since in Scerri et al.’s approach, the task can only be performed by the agent that holds its token, it is hard to handle complex tasks (i.e., tasks cannot be completed by single agent) or interdependent tasks (i.e.,

a task can only be performed when another task are completed) in disaster environments.

Beside “token-based” mechanisms, some researchers employed the max-sum algorithm for agents to share information for task allocation under communication constraints. Ramchurn et al. [112] proposed a max-sum-based task allocation approach for RoboCup Rescue. In Ramchurn et al.’s approach, the objective function of all agents in a disaster environment is exchanged among agents through the communication network. In the objective function, there are a number of factors corresponding to decisions on task allocation of agents. When an agent receives the objective function, the agent can collect information for task allocation of other agents by their factors and maximise the objective function (i.e., the benefit of all agents) through adjusting its corresponding factors. Ramchurn et al.’s approach effectively and efficiently combines the information sharing and task allocation together. Based on Ramchurn et al.’s approach, agents in a disaster environment can cooperatively find the task allocation solution that can maximise the objective function. However, since the max-sum algorithm requires a certain amount of time for agents to exchange and adjust the objective function, Ramchurn et al.’s approach is not applicable in highly dynamic disaster environments.

All above decentralised task allocation approaches were proposed for cooperative agents, where agents in a disaster environment aim to maximise the benefit of all agents in the environment. In recent years, a number of decentralised task allocation approaches were proposed for non-cooperative and competitive agents in disaster environments.

The Contract Net Protocol (CNP) [77] is an effective and efficient tool for task allocation among non-cooperative agents. Sugawara et al. [136] proposed a CNP-based approach for task allocation in multi-agent systems. In their approach, in order to elicit the capabilities of competitive agents in large-scale environments (i.e., disaster environments, market environments, etc), the award strategy of agents can be dynamically adjusted through the CNP. Based on Sugawara et al.’s approach, the performance of agents in a large-scale environment can be improved greatly due to the motivated award strategy. However, in disaster environments, the CNP-based task allocation approaches can only find sub-optimal solutions for task allocation. This is because when an agent sends a request, due to the communication constraints, only some of agents rather than all agents in an environment can receive and reply the request.

Except for the CNP, the game theory was also proved to be able to elicit enthusiasms of non-cooperative agents through competition. Chapman et al. [21] proposed a game theory-based task allocation approach for disaster environments. In their approach, a long-term task allocation problem is divided into a number of time intervals. In each time interval,

the task allocation problem is formulated as a potential game, in which, agents trend to maximise their individual benefits through making decisions on performing certain tasks based on their current states. Through potential games (i.e., all time intervals), individual benefits of all agents in a disaster environment can be maximised. Chapman et al.'s approach is also a myopic approach, which trends to maximise individual benefits of all agents through maximising individual benefits of all agents in the current time interval without considering following time intervals. Therefore, task allocation solutions created by Chapman et al.'s approach are also sub-optimal.

From above introductions, it can be seen that decentralised task allocation approaches have following **advantages**.

- **Low communication dependence**

Many decentralised information sharing mechanisms (i.e., see Subsection 2.1.2) can be employed by decentralised task allocation approaches, which helps agents to efficiently collect information for task allocation under communication constraints.

- **Low computation complexity**

Since in decentralised task allocation approaches, each agent only creates task allocation solutions for itself, during task allocation, an agent just needs to calculate tasks and agents related to it, which greatly reduce the complexity for solution generations.

Comparing with centralised task allocation approaches, decentralised task allocation approaches have following **disadvantages**.

- **Sub-optimal solutions**

Since task allocation solutions are created by agents based on their local views, the optimization of these solutions cannot compare with the solution created by the central controller based on its global view.

- **Time consuming on information collection**

In decentralised task allocation approaches, without central controllers, agents have to spend a large amount of time to collect information for task allocation in disaster environments under communication constraints.

- **Conflicts and inconsistent solutions**

Since different agents might have different local views, agents in decentralised task allocation approaches might not create consistent solutions for task allocation so that some conflicts could exist among their solutions.

### Hybrid task allocation approaches

The architectures of hybrid task allocation approaches are various, which might have features of centralised and decentralised task allocation approaches at the same time. One kind of hybrid task allocation approaches are coalition formation approaches. In coalition formation approaches, agents in a disaster environment form coalitions to achieve centralised task allocation within each coalition and decentralised coordination and adjustment between coalitions.

With the development of task allocation approaches, the complex task problems appeared in disaster environments. A complex task requires a number of agents with different capabilities to complete so that task allocation approaches for single agent cannot handle the complex task allocation problems. There are two directions to handle the complex task allocation problem, which are the task division and the coalition formation.

- **Task division**

Based on task division approaches, complex tasks are first divided into many suitable subtasks, which are suitable to be completed by single agents in the environment. Then, suitable single agents are allocated to subtasks.

- **Coalition formation**

Based on coalition formation approaches, a number of agents first form coalitions based on existing complex tasks. Then, suitable coalitions are allocated to corresponding complex tasks.

In addition, it is found that due to the tight cooperation, the work efficiencies of agents working in coalitions can be improved. For example, if  $E(\cdot)$  is used to represent work efficiencies of agents, the relationship between two agents (i.e.,  $a$  and  $b$ ) working in a coalition and individually can be described as  $E(a + b) \geq E(a) + E(b)$ . Since the last decade, many coalition formation approaches for task allocation have been proposed for disaster environments from different perspectives.

Shehory and Kraus [130] proposed coalition formation algorithms for task allocation. In their approach, Shehory and Kraus proved that through forming coalitions, agents can perform tasks that cannot performed by single agents (i.e., complex tasks) to improve the work efficiencies of agents in coalitions. In their approaches, Shehory and Kraus handled the coalition formation problem in two following situations.

- An agent can only belong to one coalition; and
- An agent can join in multiple coalitions.

Since the Shehory and Kraus's algorithms are any-time algorithms and form coalitions in a decentralised manner, their algorithms can work well in open and dynamic environments, such as disaster environments, market environments, etc.

Shehory et al. [131] proposed a modification method to enable theoretical coalition formation algorithms to be applied to real-world multi-agent systems. In their paper, Shehory et al. pointed out that most theoretical coalition formation algorithms are developed based on the game theory, which are centralised and computation complex. In order to apply these algorithms in decentralised MASs in open and dynamic environments, Shehory et al. proposed a modification method through relaxing a number of common assumptions and limitations in theoretical coalition formation algorithms. They also proved that the modification method can enable theoretical algorithms to create near-optimal solutions on coalition formation for decentralised MASs in open and dynamic environments (i.e., agents in disaster environments) without the sophisticated calculation.

Koes et al. [80] proposed a MILP-based optimization framework to help a team of robots to replan under communication constraints. In Koes et al.'s approach, a team of robots is divided into many fractured subteams (i.e., coalitions), where robots in the same subteam have the similar plan and can communicate with each other. Then, the Constraint Optimization Coordination Architecture (COCOa) is employed to map the team planning problem of each subteam to a MILP problem and find the optimal replanning solution for the subteam. Koes et al.'s framework can enable a team of robots to dynamically replan under communication constraints so as to suit open and dynamic environments.

Ramchurn et al. [113] proposed a task allocation approach through coalition formation for complex tasks. Ramchurn et al.'s approach mainly handles long-term task allocation so that in their approach, agents form coalitions based on the reusability of coalitions. Therefore, based on Ramchurn et al.'s approach, coalitions in a disaster environment do not need to frequently dissolve and reform, which can improve the work efficiencies of agents working in same coalitions due to their tight and long-term cooperations.

From above reviews, it can be seen that coalition formation approaches for task allocation have following **advantages**.

- **Handling complex tasks**

In disaster environments, many complex task requires a number of agents with different capabilities to complete. Through coalition formation, agents can cooperatively perform such tasks in the environments.

- **Improvement of work efficiency**

Due to the tight cooperation among agents in coalitions, the work efficiencies of agents

can be improved greatly during task performing.

However, coalition formation approaches for task allocation also have following **disadvantages**.

- **High computation complexity**

In [130], Shehory and KrausIt has proved that the process of forming effective and efficient coalitions is also NP-hardness.

- **Inflexibility**

Since most of approaches form coalitions for particular complex tasks, it is hard for these coalitions to handle all complex tasks a disaster environment. Therefore, agents have to frequently dissolve and reform coalitions in a dynamic disaster environment.

### 2.2.3 Testbeds for task allocation in disaster environments

In order to comprehensively evaluate task allocation approaches and present real-world situations of disaster environments to researchers, many testbeds for task allocation approaches in disaster environments were proposed in recent years.

Dimakis et al. proposed a simulator built on top of JADE platform [34], [40]. The objective of their simulator is to simulate building evacuation scenarios (i.e., evacuation when building is ablaze) with heterogeneous autonomous agents. Dimakis et al.'s simulator is characterised by the multi-agent paradigm, the achievement on entities' interaction and the distributed decision making of entities. In addition, the Dimakis et al.'s simulator also simulates the disaster spread in the building to reflect the real-world situation in disaster environments. However, since the Dimakis et al.'s simulator mainly focus on disasters in buildings, some special feature in large-scale disaster environments are not considered.

The DARPA coordinators program is a popular testbed for multi-tasks allocation in dynamic environments [81]. In the DARPA coordinators program, the relationships among tasks are represented by a complicated hierarchical network and each agent can only have the local view about its responsible tasks. In addition, tasks in the hierarchical network are interdependent, where a task can be performed, when all its interdependent tasks have been completed. Although the DARPA coordinators program can comprehensively describe the relationships among tasks in disaster environments, it neglects the communication constraints in disaster environments. Therefore, in approaches [100], [133], [9] for the DARPA coordinators program, agents are assumed to be able to freely communicate with each other so as to share information about interdependent tasks and achieve efficient coordination for task allocation.



The RoboCup Rescue (RCR) platform [75] is also a popular testbed for disaster environments simulation, which is established due to the the Great Hanshin earthquake and continuously maintained by Autonomous Learning Agents for Decentralised Data and Information Networks [75]. The RCR platform aims to simulate real-world situations in large-scale disaster environments so that the platform considers heterogeneous agents, such as civilians, fire brigades, ambulances, police agents, etc and different situations such as road blocks, traffic congestion, communication constraints, etc in disaster environments. Due to the comprehensive and realistic simulations, nowadays, the RCR have been widely applied to evaluate performance of task allocation approaches in large-scale disaster environments.

## **2.3 Ad hoc Network Establishment in Disaster Environments**

In disaster environments, due to the destruction of local communication infrastructures, a number of wireless mobile robots (WRs) are usually employed to establish ad hoc networks [32], [69], [122], [94] so as to achieve effective and efficient task-based resource management in such environments. In this section, the approaches related to ad hoc network establishment in disaster environments are introduced in detail. In particular, the classification of ad hoc network establishment approaches are introduced at first. Then, ad hoc network establishment approaches for disaster environments are reviewed.

### **2.3.1 Classification of ad hoc network establishment approaches**

Since the last decade, with the development of wireless technologies, ad hoc networks established by WRs have played an important role in task-based resource management in disaster environments, due to their low infrastructure dependence [105], low expenses [154], quick deployment [8], quick adaptability [127] and scalability [63]. Current ad hoc network establishment approaches [28], [56], [15] were classified by researchers from different perspectives.

Tang et al. [139] classified ad hoc network establishment approaches into unconnected and connected ad hoc network establishment approaches based on whether the deployed WRs are connected with each other.

- **Unconnected ad hoc network establishment approaches**

In unconnected ad hoc network establishment approaches, the main objective of the ad

hoc network establishment is to collect information for task allocation in disaster environments. Therefore, WRs are deployed at the locations that maximise the coverage of important locations (ILs) (i.e., locations of tasks, sensors, agents, etc) in disaster environments. By doing so, the deployed WRs can collect information from their covered sensors in disaster environments and transmit the collected information to base stations and sinks or guide.

- **Connected ad hoc network establishment approaches**

In connected ad hoc network establishment approaches, in order to improve the lifetime and performance of the established ad hoc networks, WRs are deployed at the locations that not only can cover as many ILs in a disaster environment as possible, but also can communicate at least one deployed WR in the environment. By doing so, a deployed WR can save energy for information transmission and share information with other WRs in the established ad hoc networks.

Except for Tang et al.'s classification, current approaches for the ad hoc network establishment can be also classified into non-greedy algorithm-based approaches and greedy algorithm-based approaches based on by what kind of algorithms the ad hoc network are used to establish networks.

- **Greedy algorithm-based approaches**

This kind of approaches establish ad hoc networks based on greedy algorithms and deploy WRs in a disaster environment one by one. New WRs are deployed at the locations that can maximise the number of additional ILs covered by the established network without adjusting deployment locations of the WRs that have already deployed in the network.

- **Non-greedy algorithm-based approaches**

This kind of approaches establish ad hoc networks based on non-greedy algorithms, mostly the mathematical programming, which deploy all WRs in a disaster environment at a time. Different from greedy algorithm-based approaches, non-greedy algorithm-based approaches can maximise ILs covered by all WRs in the established network rather than the new WRs.

### **2.3.2 Current approaches for ad hoc network establishment in disaster environments**

From the last century, disasters have become important social and political concerns. In disaster environments, many tasks need to be performed by agents. In such environments, WRs are usually employed to establish ad hoc networks so as to collect information and guide agents for task allocation in disaster environments. With the guidance of WRs in a established ad hoc network, the work efficiencies of agents in a disaster environment can be improved. As introduced in Section 1.2, due to special features of disaster environments, agents needs to handle a number of challenging issues to establish an ad hoc network. In this subsection, current approaches for ad hoc network establishment in disaster environments are introduced in detail based on the classification of Tang et al. [139].

#### **Approaches for unconnected ad hoc network establishment**

In approaches for unconnected ad hoc network establishment, the WRs cannot communicate with each other. Most of these approaches assumed that WRs are powerful enough to directly communicate with base stations or sinks. Hence, the main objective of this kind of approaches is to cover the maximum ILs and areas in disaster environments through deploying a limited number of WRs in the environments. To achieve this objective, many this type of approaches were developed in the last decade.

In order to maximise the coverage of areas of the established ad hoc networks in disaster environments, Heo et al. [61] proposed a sensor deployment approach for the sensor network establishment. In their approach, sensors trend to maximise their area coverage through minimising the overlaps between their sensing ranges. In order to achieve this objective, a self-spread approach, inspired by the equilibrium of molecules, is employed by Heo et al.'s approach. The strength of interaction forces between two sensors in a network is calculated from the distance between the two sensors. The final locations of sensors in Heo et al.'s approach are their balance points of interaction forces of their surrounding sensors. Based on the Heo et al.'s approach, sensors can autonomously and evenly distributed in a disaster environment and cover as many areas in the environment as possible.

Bilbao et al. [14] proposed a hybrid heuristics for the dynamic relay deployment problem in disaster environments. In their approach, relays are assumed to be powerful enough to directly communicate with satellites, with the assistant of which, collected information can be transmitted to the base station out of disaster environments. In order to dynamically deploy relays in disaster environments, the main objectives of Bilbao et al.'s approach include

- minimising the number of relays deployed in a disaster area;
- maximising the coverage of areas of deployed relays; and
- minimising the cost of the relay deployment.

In order to achieve the above three objectives, Bilbao et al.'s approach employs heuristics including a harmony search-based global search mechanism and a K-means-based local search mechanism. From simulations, it is found that Bilbao et al.'s approach can maximise relays' coverage areas in disaster environments.

In order to maximise the coverage of ILs of the established ad hoc networks in disaster environments, Guo et al. [56] proposed three dynamic relay deployment approaches for the establishment of wireless network in disaster environments. The same as Bilbao et al.'s approach, relays in Guo et al.'s approach also directly communicate with satellites to transmit collected information to base stations or sinks. Hence, the main objective of their approaches is to maximise first responders (i.e., ILs) covered by relays in the established network. In order to achieve this objective, Guo et al.'s approaches employed two-vertex square covering algorithm, the circle covering algorithm and the binary integer programming algorithm. The first two algorithms are based on greedy algorithms to find near-optimal deployment locations for relays in a short time. The third algorithm is based on mathematical programming, which can find optimal deployment locations for relays with sophisticated calculations.

From above introductions, it can be seen that unconnected ad hoc networks have following **advantages**.

- **The maximum coverage of ILs and areas**

Since the communication of WRs is not considered in approaches for unconnected ad hoc network establishment, the coverage of ILs and areas of unconnected ad hoc networks are big so that the established networks can cover the maximum ILs and areas in disaster environments.

- **The minimum number of WRs**

For the same reason, approaches to establish unconnected ad hoc networks can deployed the minimum WRs to achieve the maximum coverage of ILs and areas in disaster environments.

However, unconnected ad hoc networks have following **disadvantages**, which limit their applications in disaster environments.

- **Low work efficiency**

Since WRs in this kind of ad hoc networks cannot communicate with each other, each

WR needs to perform jobs (i.e., collecting and transmitting information, guiding agents for task allocation, etc) by itself. Without cooperation of agents, the work efficiencies of WRs in the established ad hoc networks are low.

- **Short lifetime**

In [119], it was proved that the energy consumption of WRs for information transmission is proportional to the square of their transmission distance. Since WRs in the unconnected ad hoc networks have to directly transmit information to based stations or sinks by themselves, the lifetimes of unconnected ad hoc networks are short.

### **Approaches for connected ad hoc network establishment**

In approaches for connected ad hoc network establishment, the WRs in the established ad hoc network are communicable so as to extend the lifetime and improve the work efficiencies of WRs in the established ad hoc network. Based on this consideration, the objectives of the established ad hoc networks aim to

- enable WRs to communicate with each other; and
- cover as many ILs and areas in an environment as possible.

To achieve these objectives, many this type of approaches were developed in the last decade.

Reich et al. [116] proposed a robot-sensor deployment approach for establishing robot-sensor networks for search and rescue. In their approach, a very large number of sensor robots (i.e., WRs) guide a smaller number of mobile robots (i.e., first responders) to perform tasks (i.e., ILs). In order to increase the opportunity to guide mobile robots to perform tasks, the objectives of sensor robots are to cover discovered tasks in a disaster environment and communicate with each other in the established robot-sensor networks. In order to achieve these objectives, Reich et al.'s approach deploys sensor robots based on a number of hierarchical actions. Although Reich et al.'s approach can find deployment locations for sensor robots in disaster environments, it cannot find the optimal deployment locations for the robots due to the simpleness of hierarchical actions.

Srinivas et al. [134] proposed a mobility node placement approach for constructing and maintaining backbones in Wireless Sensor Networks (WSNs). In [134], Srinivas et al. first introduced some backbone nodes (i.e., WRs) in networks, which can be placed to suitable locations to form backbones of WSNs, to improve the performance and extend the lifetime of WSNs. In their approach, Srinivas et al. aim to select the minimum number of backbone nodes and place them at suitable locations so as to enable these nodes to cover all the other

nodes in WSNs and connect with each other. In order to achieve their objectives, authors divided the backbone placement problem into two sub-problems. The first sub-problem aims to place the minimum number of backbone nodes to cover all nodes in WSNs, handled by distributed approximation algorithms. The second sub-problem aims to place the minimum number of relays to connect all backbone nodes in WSNs, handled by a discretization approach. Based on the simulation and evaluation, authors found that their approach could use less backbone nodes and relays to form backbones in WSNs than other current approaches.

Shao et al. [128] proposed an approach for ad hoc network establishment based on aero-crafts. Different from ad hoc networks established by robots on the ground, the establishment of the aerial network is rapid and reliable, and is not affected by blocks or obstacles in disaster environments. In Shao et al.'s approach, aero-crafts worked as relays in traditional ad hoc networks to cover the maximum areas in an environment. In addition, aero-crafts in the disaster environment also form a network to cooperatively transmit the information collected from the ground to the base stations (out of disaster environments) with the assistant of satellites. Shao et al.'s approach showed the advantages of aero-crafts in disaster rescuers and directed a potential direction of future disaster rescues.

The connected ad hoc networks have following **advantages**.

- **High work efficiency**

Since WRs in connected ad hoc networks can communicate with each other, the information about covered ILs and agents can be shared among WRs and efficient task allocation can be achieved within the entire networks.

- **Long lifetime**

In [119], Rogers et al. has proved that the energy consumption of WRs for information transmission is proportional to the square of their transmission distance. WRs in connected ad hoc networks can transmit information with assistant of other WRs in the networks, which can reduce transmission distances of WRs, so that the lifetime of WRs is increased.

Compare with unconnected ad hoc networks, connected ad hoc networks have following **disadvantages**.

- **Low coverage of ILs and areas**

Since WRs in connected ad hoc networks can communicate with each other, each WR must sacrifice some coverage areas as overlaps so as to communicate with other WRs in the networks.

- **High number of WRs**

For the same reason, if an ad hoc network trend to cover the same number of ILs and areas in an environment, the number of WRs required by connected ad hoc networks is much more than that required by unconnected ad hoc networks.

## 2.4 Summary

The key contributions and critics of different categories of approaches introduced in this chapter are summarized in Table 2.1.

Table 2.1: The comparison of different categories of task allocation and ad hoc network establishment approaches

	Categories	Key contributions	Critics
Task allocation approach	Centralised	<ol style="list-style-type: none"> <li>1. Optimal or near optimal solutions</li> <li>2. Fast information collection</li> <li>3. Less conflicts</li> </ol>	<ol style="list-style-type: none"> <li>1. High communication dependence</li> <li>2. High computation complexity</li> </ol>
	Decentralised	<ol style="list-style-type: none"> <li>1. Low communication dependence</li> <li>2. Low computation complexity</li> </ol>	<ol style="list-style-type: none"> <li>1. Sub-optimal solutions</li> <li>2. Time consuming on information collection</li> <li>3. Conflicts and inconsistent solutions</li> </ol>
	Hybrid	<ol style="list-style-type: none"> <li>1. Handling complex tasks</li> <li>2. Improvement of work efficiency</li> </ol>	<ol style="list-style-type: none"> <li>1. High computation complexity</li> <li>2. Inflexibility</li> </ol>
Ad hoc network establishment approach	Unconnected	<ol style="list-style-type: none"> <li>1. The maximum coverage of ILs and areas</li> <li>2. The minimum number of WRs</li> </ol>	<ol style="list-style-type: none"> <li>1. Low work efficiency</li> <li>2. Short lifetime</li> </ol>
	Connected	<ol style="list-style-type: none"> <li>1. High work efficiency</li> <li>2. Long lifetime</li> </ol>	<ol style="list-style-type: none"> <li>1. Low coverage of ILs and areas</li> <li>2. High number of WRs</li> </ol>

In this chapter, current approaches for task-based resource management were reviewed and analysed comprehensively. In particular, first, approaches related to information collection in disaster environments were reviewed in Section 2.1, where searching and information sharing approaches for agents in disaster environments are reviewed in detail. Then, approaches related to task allocation in disaster environments were reviewed in Section 2.2, where the classification of task allocation approaches, task allocation approaches and testbeds for task allocation approaches in disaster environments were given, respectively. Finally, approaches related to ad hoc network establishment were reviewed in Section 2.3,

---

where the classification of ad hoc network establishment approaches and ad hoc network establishment approaches for disaster environments were given in detail.



# Chapter 3

---

## A Weighted Task Allocation Approach

Dynamic task allocation based on available resources is a very challenging research issue in disaster environments with time, space and communication constraints. In addition, the space and communication constraints and the dynamic features of disaster environments make an extra difficulty to achieve efficient task allocation through centralised coordination approaches, which require a central controller to have global knowledge of the environments. To this end, a coordinated task allocation approach for weighted tasks is proposed in this chapter. The proposed approach includes three mechanisms. In particular,

- a group formation mechanism is developed to help agents to form groups and select coordinators by considering communication constraints of disaster environments (to solve **Challenging Issue 2**, identified in Section 1.2);
- a token passing mechanism is employed to help coordinations to collect information for task allocation from agents by considering limited communication capabilities of agents (to solve **Challenging Issue 2**, identified in Section 1.2); and
- a utility calculation mechanism is developed to help coordinators to create solutions for task allocation by considering multiple constraints of disaster environments and urgent degrees of tasks (to solve **Challenging Issues 3 and 4**, identified in Section 1.2);

### 3.1 Problem Description and Definition

In general, agent-based task allocation involves to model the coordinating problem of a set of agents during the task allocation process. The set of agents contains  $M$  agents, which can be described as  $\{A_1, A_2, A_3, \dots, A_M\}$ , where  $A_i$  represents the  $i^{th}$  agent and  $1 \leq i \leq M$ . Each agent can discover its nearby tasks and index these tasks as  $T_{ij}$ , where  $T_{ij}$  represents

the  $j^{th}$  task discovered by  $A_i$ . The following definitions are given to describe the coordinated task allocation problem in detail.

**Definition 3.1.** An Agent ( $A_i$ ) can be defined as a six-tuple  $A_i = \langle ANo, Uti_i, Loc_i, MSp_i, Comm_i, ASta_i \rangle$ , where  $ANo$  is the ID of  $A_i$ ;  $Uti_i$  is the work efficiency of  $A_i$ , which represents how many units of workload that  $A_i$  can perform per time unit;  $Loc_i$  is the current location of  $A_i$ ;  $MSp_i$  is the moving speed of  $A_i$ , which represents how many units of distance that  $A_i$  can move per time unit;  $Com_i$  is the communication range of  $A_i$ , which represents the maximum units of distance that  $A_i$  can directly communicate with; and  $ASta_i$  is the status of  $A_i$ , which can be either ‘available’ or ‘working’.

In order to distinguish different urgent degrees of workloads among tasks, the variable  $Emg_{ij}$  is proposed. By taking  $Emg_{ij}$  into account, the definition of a task is given as follows.

**Definition 3.2.** A Task ( $T_{ij}$ ) can be defined as a six-tuple  $T_{ij} = \langle TNo, DL_{ij}, WL_{ij}, Loc_{ij}, Emg_{ij}, TSta_{ij} \rangle$ , where  $TNo$  is the ID (generated by the agent which discovered the task) of  $T_{ij}$ ;  $DL_{ij}$  is the deadline of  $T_{ij}$  and  $DL_{ij} \in [0, \infty]$ ;  $WL_{ij}$  is the workload of  $T_{ij}$ , which represents how many units of workload must be done to complete  $T_{ij}$ ;  $Loc_{ij}$  is the location of  $T_{ij}$ ;  $Emg_{ij}$  is the urgent degree of the workload of  $T_{ij}$  and  $Emg_{ij} \in [1, 10]$ , where 1 and 10 represent the lowest and the highest urgent degrees of workloads of tasks, respectively; and  $TSta_{ij}$  is the status of  $T_{ij}$ , which is either ‘available’, ‘working’, ‘finished’ or ‘expired’.

The proposed approach in this chapter deals with the task allocation problem by considering the urgent degrees of workloads of tasks, called the weighted task allocation problem. The main objective of a weighted task allocation problem is to find an allocation solution  $Alloc^*$  to maximise the sum of weighted workloads of finished tasks, which is described as follows.

$$Alloc^* = \operatorname{argmax} \sum_{Finish(T_{ij})=1} WL_{ij} \cdot Emg_{ij}, \quad (3.1)$$

where  $Finish(T_{ij})$  is a Boolean function under the conditions that if  $T_{ij}$  is finished, i.e.,  $Finish(T_{ij}) = 1$ , otherwise  $Finish(T_{ij}) = 0$ ;  $WL_{ij}$  is the workload of a finished task; and  $Emg_{ij}$  is the corresponding urgent degree of the workload of  $T_{ij}$  (see Definition 3.2).

In the proposed approach, a token describing and passing mechanism is employed to describe information of tasks and agents and help agents to efficiently share such information for task allocation. There are two types of tokens: agent tokens (e.g.  $AToken_i$ ) and task tokens (e.g.  $TToken_{ij}$ ). The definitions of two types of tokens are presented in Definition 3.3 and Definition 3.4, respectively.

**Definition 3.3.** An *Agent Token* ( $AToken_i$ ) is generated by  $A_i$  for itself, which can be defined as a four-tuple  $AToken_i = \langle ANo, Uti_i, Loc_i, MSp_i \rangle$ , where  $ANo$  is the ID of the agent represented by  $AToken_i$ ;  $Uti_i$  is the work efficiency of the agent represented by  $AToken_i$ ;  $Loc_i$  is the current location of the agent represented by  $AToken_i$ ; and  $MSp_i$  is the moving speed of the agent represented by  $AToken_i$ .

**Definition 3.4.** A *Task Token* ( $TToken_{ij}$ ) is generated by  $A_i$  for its  $j^{th}$  discovered task, which can be defined as a five-tuple,  $TToken_{ij} = \langle TNo, DL_{ij}, WL_{ij}, Loc_{ij}, Emg_{ij} \rangle$ , where  $TNo$  is the ID of the task represented by  $TToken_{ij}$ ;  $DL_{ij}$  is the deadline of the task represented by  $TToken_{ij}$ ;  $WL_{ij}$  is the workload of the task represented by  $TToken_{ij}$ ;  $Loc_{ij}$  is the location of the task represented by  $TToken_{ij}$ ; and  $Emg_{ij}$  is the urgent degree of the workload of the task represented by  $TToken_{ij}$ .

Agents can have two different roles, i.e., *coordinators* and *resource providers*. An agent can be a *coordinator* or a *resource provider* or both.

- A *coordinator* is an agent, which is in charge of allocating agents to tasks.
- A *resource provider* is an agent, which is in charge of finishing tasks.

In general disaster environments, the constraint of communication ranges is a common problem. In this situation, an agent can only directly communicate with other agents close to its location within its communication range (i.e.,  $Com_i$ , see Definition 3.1). The direct neighbours of an agent is defined as follows.

**Definition 3.5.** The *direct neighbours* of an agent  $A_i$  are the agents within  $A_i$ 's communication range. (i.e., the Euclidean distances between  $A_i$  and its *direct neighbours* cannot be more than the communication range of  $A_i$  (represented by  $Com_i$ , see Definition 3.1)).

## 3.2 The Basic Principle of the Coordinated Task Allocation Approach

The proposed coordinated task allocation approach consists of five looping steps. 1) token generation, 2) group formation, 3) token passing, 4) task allocation, and 5) solution return. The five steps of the proposed approach is shown by Figure 3.1.

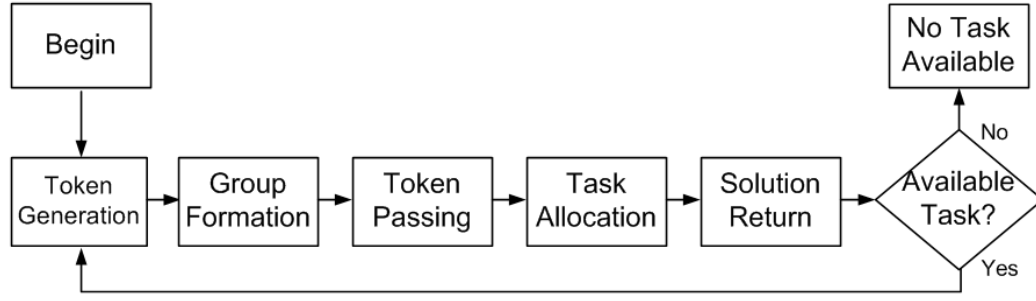


Figure 3.1: The five steps of the proposed approach in one loop

The process of the coordinated task allocation approach is described by Algorithm 3.1.

---

**Algorithm 3.1:** The process of coordinated task allocation approach
 

---

- 1 Available agents collect information for task allocation in a disaster environment
  - 2 **while** *There are available tasks in the environment* **do**
  - 3     **for** *each available agent  $A_i$ , where  $ASta_i = \text{'available'}$*  **do**
  - 4          $A_i$  generates  $AToken_i$  and  $TToken_{ij}$ .
  - 5     **end**
  - 6     All available agents form groups and select coordinators based on the group formation mechanism.
  - 7     **for** *each formed group* **do**
  - 8         available agents pass their  $AToken_i$  and  $TToken_{ij}$  to the coordinator  $C$  based on the token passing mechanism.
  - 9          $C$  creates task allocation solutions based on the utility calculation mechanism.
  - 10         $C$  returns task allocation solutions to their group members.
  - 11        The group are dismissed and available agents begin to work on their allocated tasks.
  - 12     **end**
  - 13 **end**
- 

Algorithm 3.1 is explained as follows. After collected information in a disaster environment, available agents (i.e.,  $A_i$ , where  $ASta_i = \text{'available'}$ , see Definition 3.1) will generate task tokens (i.e.,  $TToken_{ij}$ , see Definition 3.4) for available tasks (i.e.,  $T_{ij}$ , where  $TSta_{ij} = \text{'available'}$ , see Definition 3.2) in the environment and agent tokens (i.e.,  $AToken_i$ , see Definition 3.4) for themselves (Lines 1 to 5). Then, available agents form temporary groups and select a coordinator (i.e.,  $C_k$ ) for each group based on the group formation mechanism (Line 6). Based on the token passing mechanism, available agents in each group pass

their agent tokens and task tokens to the coordinator  $C$  of their group (Line 7 to 8). Based on the utility calculation mechanism, the coordinator  $C$  creates the task allocation solution for its group members (Line 9). Then, the coordinator returns the task allocation solution to their group members (Line 10). After one loop of the five steps, groups formed in group formation step will be dismissed and agents begin to work on their allocated tasks according to the allocation solution (Line 11). When some agents finish their allocated tasks and there are still available tasks in the environment, the above five steps will be repeated again according to the information of available tasks and agents at that time, so the dynamic task allocation will be achieved. The technical design for five steps are introduced in following subsections.

### 3.2.1 Token generation

In this step, each available agent  $A_i$  (i.e.,  $ASta_i$  = ‘available’, see Definition 3.1) checks its surrounding area and generates a task token,  $TToken_{ij} = \langle T_{ij}, DL_{ij}, WL_{ij}, Loc_{ij}, Emg_{ij} \rangle$  (see Definition 3.4) for each available task  $T_{ij}$  (i.e.,  $TSta_{ij}$  = ‘available’, see Definition 3.2) that  $A_i$  discovered. After task checking,  $A_i$  also generates an agent token,  $AToken_i = \langle A_i, Uti_i, Loc_i, MSp_i \rangle$  (see Definition 3.3) for itself.

### 3.2.2 Indirected neighbours group formation

Due to space and communication constraints in disaster environments, each agent can only directly communicate with its direct neighbours (see Definition 3.5). In order to get much information for task allocation in such disaster environments, agents could form groups to share information with their surrounding neighbours for task allocation. Under this consideration, the group formation mechanism should help agents to connect as many other agents in the environment as possible. Many group formation mechanisms [43], [44], [46], [113] have been developed in multi-agent research under different considerations and very few of them consider the communication constraints, especially the constraints of communication ranges. These group formation mechanisms form groups through connecting only direct neighbours, we call this kind of mechanisms as Direct Neighbours Group Formation (DNGF) mechanisms in this thesis. In order to collect as much information for task allocation as possible under space and communication constraints in disaster environments, we develop a group formation mechanism in the proposed approach to help agents to form groups through connecting as many agents as possible including direct and indirect neighbours within communication ranges of agents in a decentralised manner. The formed group is a tree structure, in which the agent at the root node is chosen to be the coordinator of the group. To distinguish

with DNGF mechanisms, the group formation mechanism proposed in this chapter is called Indirect Neighbours Group Formation (INGF) mechanism. The difference between DNGF and INGF can be described in Figure 3.2.

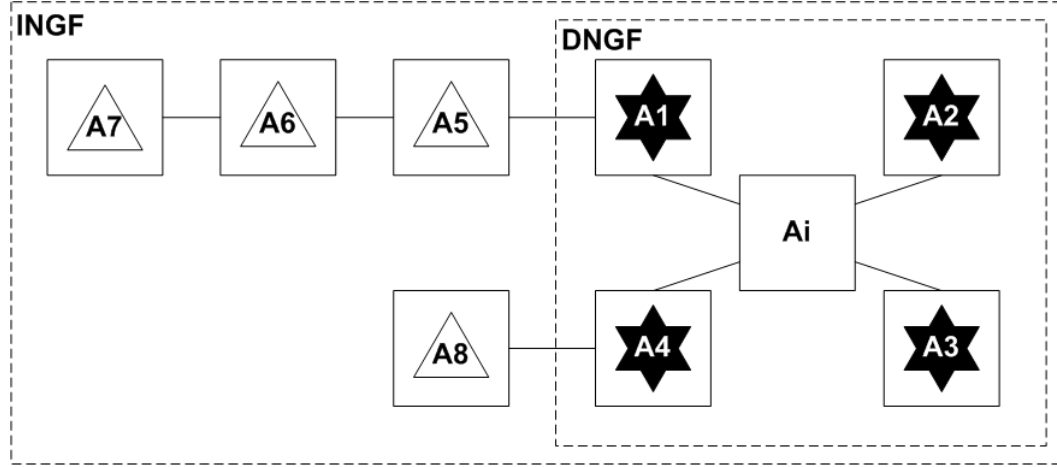


Figure 3.2: The difference between INGF and DNGF

In Figure 3.2, the rectangles represent the agents and solid lines between agents indicate that the two agents are direct neighbours of each other (see Definition 3.5). In the DNGF mechanism, an agent  $A_i$  can only form a group with its direct neighbours i.e.,  $A_1$ ,  $A_2$ ,  $A_3$  and  $A_4$  (black stars), while in INGF mechanism,  $A_i$  can add four indirect neighbours i.e.,  $A_5$ ,  $A_6$ ,  $A_7$  and  $A_8$  (white triangles) to its group. The more agents in a group, the more information can be collected by the coordinator of the group and the more efficient task allocation can be achieved.

In the INGF mechanism, three ‘neighbour related’ variables are defined for each agent, which are the parent agent ( $PA$ ), the coordinator ( $C$ ) and the number of direct neighbours of the coordinator ( $NNC$ ). For example, for an agent  $A_i$ , the three ‘neighbour related’ variables can be denoted as  $A_i.PA$ ,  $A_i.C$  and  $A_i.NNC$ , respectively. The proposed group formation mechanism is described by Algorithm 3.2.

**Algorithm 3.2:** INGF mechanism

---

```

1 for each Agent  $A_i$  do
2    $A_i.PA \leftarrow A_i$ ;  $A_i.C \leftarrow A_i$ ;  $A_i.NNC \leftarrow$  the number of direct neighbours of  $A_i$ 
3 end
4 for each Agent  $A_i$  do
5   Get  $A_u$  from  $A_i$  and  $A_i$ 's direct neighbours, where  $A_u.NNC$  is the maximum
6   if  $A_u.NNC > A_i.NNC$  then
7      $A_i.PA \leftarrow A_u$ ;  $A_i.C \leftarrow A_u.C$ ;  $A_i.NNC \leftarrow A_u.NNC$ 
8     for each direct neighbours of  $A_i$ :  $A_l$  do
9       if  $A_l.C \neq A_i.C$  then
10         $A_l.PA \leftarrow A_i$ ;  $A_l.C \leftarrow A_i.C$ ;  $A_l.NNC \leftarrow A_i.NNC$ 
11      end
12    end
13  end
14 end

```

---

At the beginning of Algorithm 3.2, the three ‘neighbour related’ variables of each agent (e.g.  $A_i$ ) are initialised as follows:  $A_i.PA$  is set to  $A_i$ ,  $A_i.C$  is set to  $A_i$  and  $A_i.NNC$  is set to the number of direct neighbours of  $A_i$  (since  $A_i$  is the coordinator of itself) (Lines 1 to 2). Then, each agent (e.g.  $A_i$ ) repeats the following three steps.

**Step 1:**  $A_i$  finds an agent (e.g.  $A_u$ ), which has the highest value of the variable  $NNC$  (i.e.,  $A_u.NNC$ ) from its direct neighbours (including  $A_i$  itself) (Line 5).

**Step 2:** If  $A_u.NNC$  has a higher value than the value of  $A_i.NNC$ , the three ‘neighbour related’ variables of  $A_i$  (i.e.,  $A_i.PA$ ,  $A_i.C$  and  $A_i.NNC$ ) are updated as follows:  $A_i.PA$  is set to  $A_u$ ,  $A_i.C$  is set to  $A_u.C$  and  $A_i.NNC$  is set to  $A_u.NNC$  (Lines 6 to 7).

**Step 3:** If the three ‘neighbour related’ variables of  $A_i$  are updated in Step 2, each direct neighbour of  $A_i$  (e.g.  $A_l$ ) compares its coordinator (i.e.,  $A_l.C$ ) with the coordinator of  $A_i$  (i.e.,  $A_i.C$ ). If a different coordinator has been found (i.e.,  $A_l.C \neq A_i.C$ ), the three ‘neighbour related’ variables of  $A_l$  (i.e.,  $A_l.PA$ ,  $A_l.C$  and  $A_l.NNC$ ) are updated as follows:  $A_l.PA$  is set to  $A_i$ ,  $A_l.C$  is set to  $A_i.C$  and  $A_l.NNC$  is set to  $A_i.NNC$  (Lines 8 to 10). Through Step 3, the values of three ‘neighbour related’ variables of  $A_i$  can be passed to its child agent  $A_l$  and the agent with the highest number of direct neighbours in a group can be the root node of the tree structure and

is chosen as the coordinator of the group.

The above three steps (Lines 4 to 10) will be repeated by each agent until no further updating for three ‘neighbour related’ variables of any agent, which means that each agent in the communication network has found its only way to pass message to the coordinator of the network.

A group formed by the INGF mechanism can connect the maximum number of direct and indirect neighbours of the coordinator. Therefore, if there are two or more groups formed according to the proposed group formation mechanism in a disaster environment, these groups are completely isolated and cannot communicate with each other according to current communication situations. The following sub-sections describe the steps of task allocation in each isolated group.

### 3.2.3 The token passing mechanism

Since in a group, most members (agents) are indirect neighbours of the coordinator, which means that these neighbours cannot directly communicate with the coordinator, a token passing mechanism is employed to help the coordinator to collect information from its group members [92], [84]. The token passing mechanism begins from the agents who do not have child agents and ends at the coordinator. By employing the the token passing mechanism, each agent first receives agent and task tokens (generated in the token generation step) from its child agents. Different agents in the same group can generate task tokens for the same task. For example,  $A_l$  and  $A_u$  are child agents of  $A_i$  and they can generate task tokens,  $TToken_{lj} = \langle T_{lj}, DL_{lj}, WL_{lj}, Loc_{lj}, Emg_{lj} \rangle$  and  $TToken_{uj} = \langle T_{uj}, DL_{uj}, WL_{uj}, Loc_{uj}, Emg_{uj} \rangle$ , respectively, for the same task. After  $A_i$  received the task tokens from  $A_l$  and  $A_u$ ,  $A_i$  found that except the first element (i.e.,  $ANo$ , see Definition3.4), the values of other variable of two task tokens (i.e.,  $DL_{lj} = DL_{uj}$ ,  $WL_{lj} = WL_{uj}$ ,  $Loc_{lj} = Loc_{uj}$ ,  $Emg_{lj} = Emg_{uj}$ , see Definition3.4) are exactly same. In this situation,  $A_i$  only keeps one task token to represent the task and abandons the other (i.e.,  $TToken_{lj}$  created by  $A_l$  is abandoned and only  $TToken_{uj}$  created by  $A_u$  is kept). After that,  $A_i$  passes the agent tokens, the remaining task tokens, and the agent token of itself (i.e.,  $AToken_i$ ) to its parent agent. Finally, all of task and agent tokens generated by the group members are passed to the coordinator of the group. Figure 3.3 shows an example of token passing.



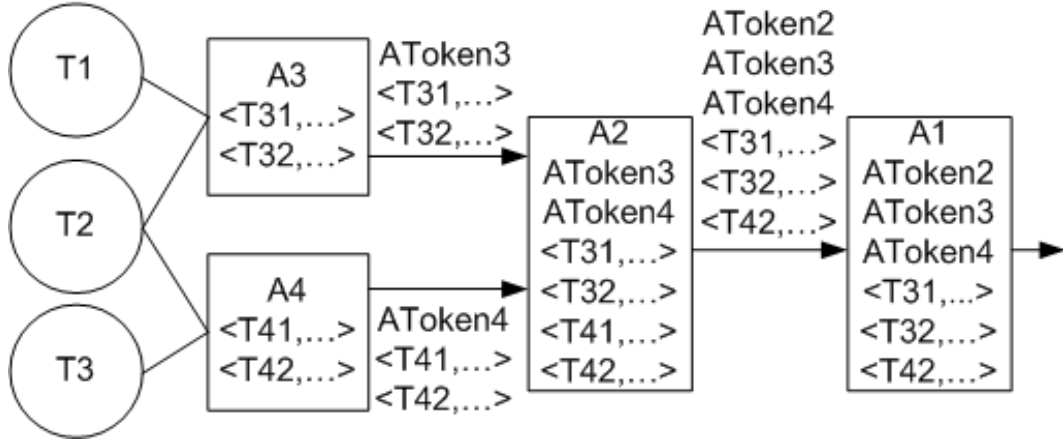


Figure 3.3: The example of token passing

There are four agents (i.e.,  $A_1$  to  $A_4$  represented by rectangles), where  $A_1$  is the parent agent of  $A_2$  ( $A_2.PA = A_1$ ),  $A_2$  is the parent agent of  $A_3$  and  $A_4$  ( $A_3.PA = A_2$  and  $A_4.PA = A_2$ ) and three tasks (i.e.,  $T_1$  to  $T_3$  represented by circles), where  $T_1$  and  $T_2$  can be discovered by  $A_3$  and  $T_2$  and  $T_3$  can be discovered by  $A_4$ . After the token generation step,  $A_3$  generates two task tokens ' $\langle T_{31}, \dots \rangle$ ' and ' $\langle T_{32}, \dots \rangle$ ', while  $A_4$  also generates two task tokens ' $\langle T_{41}, \dots \rangle$ ' and ' $\langle T_{42}, \dots \rangle$ '. Then, the task tokens and the agent tokens of  $A_3$  and  $A_4$  are passed to their parent agent  $A_2$ . After checking the variables of received task tokens,  $A_2$  discovers that task tokens ' $\langle T_{32}, \dots \rangle$ ' and ' $\langle T_{41}, \dots \rangle$ ' are generated for the same task by two different agents. Then,  $A_2$  abandons the task token ' $\langle T_{41}, \dots \rangle$ ' and keeps the task token ' $\langle T_{32}, \dots \rangle$ '. After that,  $A_2$  passes the agent tokens received from  $A_3$  and  $A_4$  ( $AToken_3$ ,  $AToken_4$ ), task tokens received from  $A_3$  and  $A_4$  ( $\langle T_{31}, \dots \rangle$  and  $\langle T_{42}, \dots \rangle$ ), the remaining task token ( $\langle T_{32}, \dots \rangle$ ), and the agent tokens of itself ( $AToken_2$ ) to its parent agent  $A_1$ .

### 3.2.4 Weighted task allocation

This step is the core of the proposed approach. In this step, the coordinator trends to find the most suitable allocation solution for its group according to the information of tasks and agents collected in the token passing step. In the proposed approach, an allocation solution  $Aloc_a$  is a set of allocations (e.g.  $Aloc_a = \{A_1 \rightarrow T_{12}, A_2 \rightarrow T_{41}, \dots, A_i \rightarrow T_{ij}\}$ ). An allocation is a map from an agent to a task (i.e.,  $A_i \rightarrow T_{ij}$ ), which represents that the agent is allocated to the task. If an agent is not allocated to any task,  $T_{ij}$  in the allocation equals to  $\emptyset$ . Since finding the most suitable allocation solution is an NP-hard problem (the proof process can be found in [113]), the proposed approach employs a utility calculation mechanism to help the coordinator to find the most suitable allocation solution. The process of finding the

most suitable allocation solution consists of two sub-steps: **1)** the elimination of useless allocation solutions; and **2)** the utilities calculation of allocation solutions.

### 1) The Elimination of Unuseful Allocation Solutions

Based on the combinatorics, the number of allocation solutions for a group including  $M$  agents and  $N$  tasks is  $(N + 1)^M$  (i.e., each agent can be allocated to one of  $N$  tasks or  $\emptyset$ ). Among these allocation solutions, there are many useless allocation solutions, in which some agents cannot finish their allocated tasks on time (before the deadlines of tasks). In order to make the following utility calculation more efficient, these useless allocation solutions should be eliminated first. The process of the elimination of useless allocation solutions can be described as follows. First, the coordinator gets all  $(N + 1)^M$  allocation solutions of its group. Second, for each task  $T_{ij}$  in each allocation solution  $Aloc_a$ , the coordinator calculates whether the agents allocated to  $T_{ij}$  can finish it before the deadline of  $T_{ij}$  according to current status of the agents.  $UFin(T_{ij})$  is the unfinish indicator function of  $T_{ij}$ , which is calculated as follows.

$$UFin(T_{ij}) = \begin{cases} 1 & WL_{ij} > \sum_{Aloc_a} (DL_{ij} - CT - \frac{Dis(Loc_{ij}, Loc_i)}{MSp_i}) Uti_i \\ 0 & WL_{ij} \leq \sum_{Aloc_a} (DL_{ij} - CT - \frac{Dis(Loc_{ij}, Loc_i)}{MSp_i}) Uti_i \end{cases}, \quad (3.2)$$

where  $WL_{ij}$  is the workload of  $T_{ij}$ ;  $A_i$  is one of agents allocated to  $T_{ij}$  in  $Aloc_a$ ;  $DL_{ij}$  is the deadline of  $T_{ij}$ ;  $CT$  is the current time;  $Dis(Loc_{ij}, Loc_i)$  is the distance between the location of  $T_{ij}$  and the current location of  $A_i$ ; and  $MSp_i$  is the moving speed of  $A_i$ ;  $Uti_i$  is the work efficiency of  $A_i$ . If  $T_{ij}$  cannot be finished by the allocated agents in  $Aloc_a$ , the function  $UFin(T_{ij}) = 1$ , otherwise  $UFin(T_{ij}) = 0$ .

$$Elim(Aloc_a) = \bigcup_{T_{ij} \in Aloc_a} UFin(T_{ij}) = \begin{cases} 1 & Aloc_a \text{ useless} \\ 0 & Aloc_a \text{ useful} \end{cases}, \quad (3.3)$$

where  $Elim(Aloc_a)$  is the elimination indicator function of  $Aloc_a$ . If any allocated task, say  $T_{ij}$ , cannot be finished in  $Aloc_a$ , the elimination indicator function of  $Aloc_a$  equals to 1 (i.e.,  $UFin(T_{ij}) = 1$ ), which means that the  $Aloc_a$  is useless and should be eliminated. For example, in a disaster environment, there are 2 agents ( $A_1, A_2$ ) and 2 tasks ( $T_{11}, T_{21}$ ). Therefore, there are totally  $(2 + 1)^2 = 9$  different kinds of allocation solutions, which are listed as follows.

$$\begin{aligned} Aloc_1 &= \{A_1 \rightarrow \emptyset, A_2 \rightarrow \emptyset\} \\ Aloc_2 &= \{A_1 \rightarrow \emptyset, A_2 \rightarrow T_{11}\} \\ Aloc_3 &= \{A_1 \rightarrow \emptyset, A_2 \rightarrow T_{21}\} \end{aligned}$$

$$\begin{aligned}
Aloc_4 &= \{A_1 \rightarrow T_{11}, A_2 \rightarrow \emptyset\} \\
Aloc_5 &= \{A_1 \rightarrow T_{11}, A_2 \rightarrow T_{11}\} \\
Aloc_6 &= \{A_1 \rightarrow T_{11}, A_2 \rightarrow T_{21}\} \\
Aloc_7 &= \{A_1 \rightarrow T_{21}, A_2 \rightarrow \emptyset\} \\
Aloc_8 &= \{A_1 \rightarrow T_{21}, A_2 \rightarrow T_{11}\} \\
Aloc_9 &= \{A_1 \rightarrow T_{21}, A_2 \rightarrow T_{21}\}
\end{aligned}$$

The information of  $T_{11}$  and  $T_{21}$  and status of  $A_1$  and  $A_2$  are shown as follow.

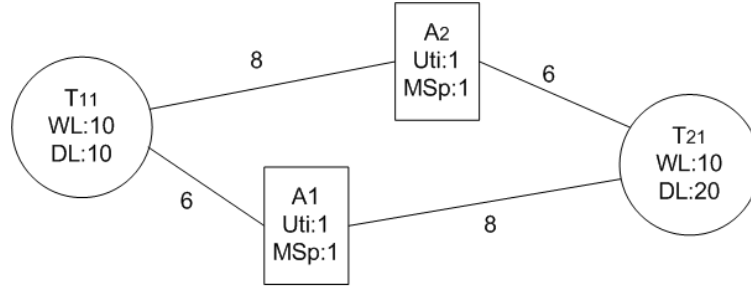


Figure 3.4: The information of tasks and status of agents

From Figure 3.4, it can be seen that both of workloads of  $T_{11}$  and  $T_{21}$  (circles) are 10 workload units and  $T_{11}$  will expire in 10 time units, while  $T_{21}$  will expire in 20 time units. The work efficiency and moving speed of  $A_1$  and  $A_2$  (rectangles) are 1 workload unit and 1 distance unit, respectively. In addition, the distances between tasks and agents are shown by the lines between them. After calculation, we can find that  $UFin(T_{11})$  equals to 1 in any allocations (i.e.,  $10 > (4 + 2)$ ,  $10 > 4$ , and  $10 > 2$ ), which means that even if both of  $A_1$  and  $A_2$  work on  $T_{11}$ ,  $T_{11}$  cannot be finished according to the information of  $T_{11}$  and status of  $A_1$  and  $A_2$ . Therefore, the allocation solutions (i.e.,  $Aloc_2$ ,  $Aloc_4$ ,  $Aloc_5$ ,  $Aloc_6$  and  $Aloc_8$ ) involving allocations of  $A_1 \rightarrow T_{11}$  or  $A_2 \rightarrow T_{11}$  are useless and should be eliminated.

## 2) The Utilities Calculation of Allocation Solutions

In this sub-step, the coordinator calculates the utilities of useful allocation solutions. The utility calculation depends on not only how many tasks that can be completed, but also the benefits that agents can receive after completing their allocated tasks and the costs that agents must spend on the completion of these tasks. Based on the benefits and costs, the utility of an allocation solution  $Aloc_a$  can be calculated by the following formula.

$$Utility(Aloc_a) = Q_{benefit} - Q_{cost}, \quad (3.4)$$

where  $Q_{benefit}$  is the benefits that agents can receive after completing their allocated tasks, which involves: 1) the sum of the weighted (urgent degree) workloads of allocated tasks and 2) the sum of saved time of completed tasks; and  $Q_{cost}$  is the costs that agents must spend for completing tasks, which involves: 1) the sum of traveling time for completing tasks and 2) the sum of working time for completing tasks.

Under above considerations, the  $Q_{benefit}$  of  $Aloc_a$  can be calculated as follows.

$$Q_{benefit} = \sum_{T_{ij} \in ran.Aloc_a} Emg_{ij} \cdot (WL_{ij} + \frac{(DL_{ij} - FT_{ij})WL_{ij}}{DL_{ij}}), \quad (3.5)$$

where  $WL_{ij}$ ,  $Emg_{ij}$  and  $DL_{ij}$  are the workload, the urgent degree of the workload and the deadline of  $T_{ij}$  in  $Aloc_a$ , respectively, and  $FT_{ij}$  is the predicted completing time of  $T_{ij}$  in  $Aloc_a$ .

The  $Q_{cost}$  of  $Aloc_a$  can be calculated as follows.

$$Q_{cost} = \sum_{A_i \in dom.Aloc_a} (FT_{ij} - CT)Uti_i, \quad (3.6)$$

where  $FT_{ij}$  is the predicted completing time of  $T_{ij}$  in  $Aloc_a$ ,  $CT$  is the current time and  $Uti_i$  is the work efficiency of  $A_i$ .

The  $FT_{ij}$  in Equation 3.5 and 3.6 can be calculated as follows:

$$FT_{ij} = \frac{WL_{ij} + CT \cdot \sum_{Aloc_a} Uti_i + \sum_{Aloc_a} \frac{Dis(Loc_{ij}, Loc_i)}{MSp_i} Uti_i}{\sum_{Aloc_a} Uti_i}, \quad (3.7)$$

where  $WL_{ij}$  is the workload of  $T_{ij}$ ;  $CT$  is the current time;  $Uti_i$  is the work efficiency of  $A_i$ , which is allocated to  $T_{ij}$  in  $Aloc_a$ ;  $Dis(Loc_{ij}, Loc_i)$  is the distance between the location of  $T_{ij}$  and the current location of  $A_i$ ; and  $MSp_i$  is the moving speed of  $A_i$ .

The allocation solution (e.g.  $Aloc_a$ ) with the highest value of utility (i.e.,  $Utility(Aloc_a)$ ) is chosen by the coordinator to be the most suitable allocation solution of its group.

### 3.2.5 Solution return

In this step, the coordinator sends the chosen allocation solution (e.g.  $Aloc_a$ ) to its group members. The process of solution return is the inverse process of token passing. Different from token passing, when an agent (e.g.  $A_i$ ) receives the  $Aloc_a$ , it needs to perform following actions based on Algorithm 3.3.

---

**Algorithm 3.3:** The actions performed by  $A_i$ 


---

```

1 for each direct neighbours of  $A_i$ :  $A_u$  do
2    $A_i \xrightarrow{Aloc_a} A_u$ 
3 end
4  $A_i$  is dismissed from its group
5 if  $Aloc_a(A_i) = T_{ij}$  then
6    $A_i$  moves to and works on  $T_{ij}$ ;
7    $ASta_i = \text{'working'}$ 
8 end
9 if  $A_i$  finishes  $T_{ij}$  then
10   $ASta_i = \text{'available'}$ ;  $TSta_{ij} = \text{'finished'}$ 
11 end

```

---

First,  $A_i$  passes the  $Aloc_a$  to its direct neighbours and  $A_i$  is dismissed from its group (Lines 1 to 4). Then, if  $A_i$  is allocated to a task in  $Aloc_a$ ,  $A_i$  changes its status (i.e.,  $ASta_i$ , see Definition 3.1) from ‘available’ to ‘working’ and starts to move to the location of its allocated task (Lines 5 to 6). When  $A_i$  reaches the location of its allocated task, it changes the status of the task (i.e.,  $TSta_{ij}$ , see Definition 3.2) from ‘available’ to ‘working’ (Line 7). After finishing the allocated task, all agents working on the same task change their status from ‘working’ to ‘available’ and the status of the task is also changed from ‘working’ to ‘finished’ (Lines 9 to 11).

### 3.3 Experiments and Analysis

Three experiments are conducted to evaluate the performance of the proposed approach.

**Experiment 1** is to evaluate the performance of the proposed group formation mechanism.

**Experiment 2** is to evaluate the performance of the proposed approach on task allocation in disaster environments.

**Experiment 3** is to evaluate the impact of urgent degrees of workloads of tasks on the proposed approach.

Three experiments are demonstrated and the results are analysed in detail in the following three sub-sections, respectively.

### 3.3.1 Experiment 1: Test of group formation

The purpose of this experiment is to evaluate the performance of the proposed group formation (INGF) mechanism under different communication ranges of agents.

#### Experiment settings

In the proposed approach, before task allocation, the agents need to share information for task allocation through forming groups under space and communication constraints. Therefore, the performance of the proposed group formation mechanism has a great impact on the results of task allocation. In this experiment, the INGF mechanism is compared with the DNGF mechanism proposed by Glington et al. [46]. The performance of the proposed group formation mechanism and the benchmark mechanism is compared under different communication ranges of agents. The settings of Experiment 1 are described in Table 3.1.

Table 3.1: The settings of Experiment 1

Name	Value
Area size	$50 \times 50 \text{ cm}^2$
Number of agents	10
Communication ranges of agents	5, 10, 15, 20, 25, 30, 35, 40 cm

In Experiment 1, the Euclidean space is employed to describe the distance between locations of two agents [99], [83]. Through the calculation, it can be seen that in a Euclidean space, if  $M$  agents with the same communication range  $D$  are randomly arranged in an  $N \times N$  area and the average number of direct neighbours of an agent can be calculated as follow:

$$Poss = (M - 1) \cdot \frac{\pi \cdot D^2}{N^2}. \quad (3.8)$$

Based on Equation 3.8, when communication ranges of agents are 5, 10, 15, 20, 25, 30, 35 and 40, the average numbers of direct neighbours of each agent are 0.28, 1.13, 2.54, 4.52, 7.07, 10.18, 13.85 and 18.10, respectively. These settings can cover most of communication situations of agents in disaster environments from ‘isolated’ to ‘full’ communication. In this experiment, two indicators are employed to represent the performance of group formation mechanisms, which are 1) the *average number of agents* of formed groups and 2) the *maximum number of agents* of the formed group.

#### Experimental results and analysis

The results of Experiment 1 are shown in Figure 3.5.

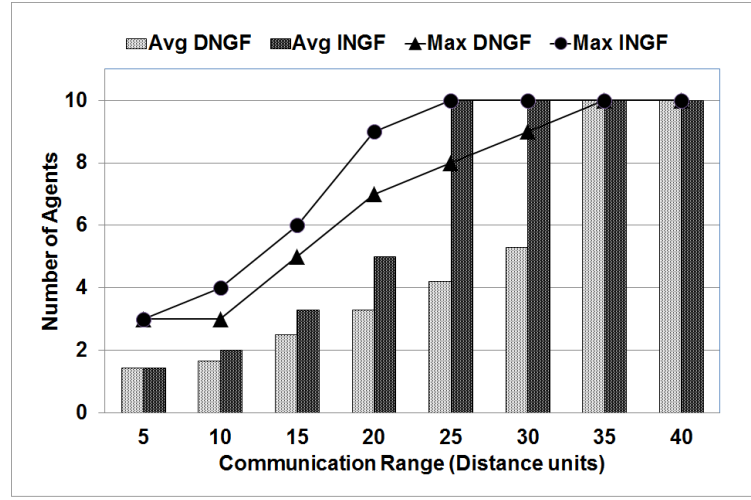


Figure 3.5: The results of Experiment 1

In Figure 3.5, the X-axis is the communication ranges of agents, while the Y-axis is the number of agents. The two indicators (i.e., the *average number of agents* of formed groups and the *maximum number of agents* of the formed group) in this experiment are represented by the bars and lines in Figure 3.5, respectively. From Figure 3.5, it can be seen that both DNGF and INGF mechanisms have the same performance on the two indicators, when the communication ranges ( $D$ ) of agents are only 5. Since the average number of direct neighbours of each agent is only 0.28, we call this communication situation ‘isolated’. In this communication situation, the biggest agent groups formed by both of group formation mechanisms contain only 3 agents and the average number of agents of formed groups through both of group formation mechanisms contain only about 1.5 agents, which means that most of agents in this communication situation cannot form groups with other agents. With the increase of the communication ranges of agents, the two indicators of groups formed through both of group formation mechanisms increase. However, the two indicators of groups formed by INGF mechanism increase shapely. That is because groups formed through the proposed group formation mechanism contain not only the direct neighbours but also the indirect neighbours of group members, while groups formed by DNGF mechanism only contain direct neighbours. Therefore, the groups formed by INGF mechanism can reach the ideal status (one group contains all of 10 agents) when the communication ranges of agents are only 25, while the groups formed by DNGF mechanism cannot reach the ideal status until communication ranges of agents are 35. Therefore, when communication ranges of agents are limited, the groups formed by INGF mechanism contain more agents than that of formed by DNGF mechanism.

### 3.3.2 Experiment 2: Test of task allocation

The purpose of this experiment is to evaluate the performance of the proposed approach on task allocation under different communication ranges of agents.

#### Experiment settings

Experiment 2 focuses on the evaluation of the task allocation performance of the proposed approach. Therefore, most of experimental parameters (i.e., the size of the area, the number of tasks and agents, the deadlines and workloads of tasks, the moving speeds and work efficiencies of agents) are set to simulate real-life disaster environments. The settings of Experiment 2 are described in Table 3.2.

Table 3.2: The settings of Experiment 2

Name	Value
Area size	$50 \times 50 \text{ cm}^2$
Number of tasks	100
Deadlines of tasks	$5 \sim 200$
Workloads of tasks	$10 \sim 50$
Urgent degrees of workloads of tasks	1
Number of agents	10
Work efficiencies of agents	1
Moving speeds of agents	1
Communication ranges of agents	$5 \sim 40 \text{ cm}$ with $5 \text{ cm}$ per step or $50 \text{ cm}$

From Table 3.2, it can be seen that in Experiment 2, 100 tasks are randomly distributed in a  $50 \times 50$  environment. The deadlines and workloads of these tasks are 5 to 200 and 10 to 50, respectively. The urgent degrees of workloads of tasks are set to 1 to remove their impact on this experiment. 10 agents with the same work efficiency (i.e., 1 at a time) and moving speed (i.e., 1 at a time) work in the environment. The benchmark approach of Experiment 2 is the heuristics task allocation approach proposed by Ramchurn et al. [113]. The reason for this choice is because that the experimental settings of this approach are similar to the proposed approach, both of which consider the time and space constraints in disaster environments. The difference between the benchmark approach and the proposed approach is that the benchmark approach uses a centralised method for task allocation without considering the communication constraints, while our approach uses a decentralised manner for task allocation. Therefore, in the experiment, it is assumed that there is no communication



constraints when conducting the benchmark approach in this experiment in order for the coordinator to get the global knowledge of the environment. For the proposed approach, eight different kinds of communication ranges of agents are simulated (i.e., from 5 to 40 with 5 per step). As mentioned in Section 3.2.2, the proposed group formation mechanism can connect the maximum number of direct and indirect neighbours of the coordinator based on the communication ranges of agents so that if two or more groups are formed through the proposed group formation mechanism, these groups are completely isolated. In order to compare the performance of the proposed approach with the benchmark (centralised) approach on task allocation, the sum of finished tasks is used in a disaster environment by all isolated groups of the proposed approach as its overall performance in the environment. Therefore, in this experiment, the number of finished tasks in a disaster environment is employed as the indicator to evaluate the performance of two task allocation approaches.

### Experimental results and analysis

The results of Experiment 2 are shown in Figure 3.6. In Figure 3.6, the X-axis is the communication ranges of agents of the proposed approach. The Y-axis is the number of tasks that have been finished.

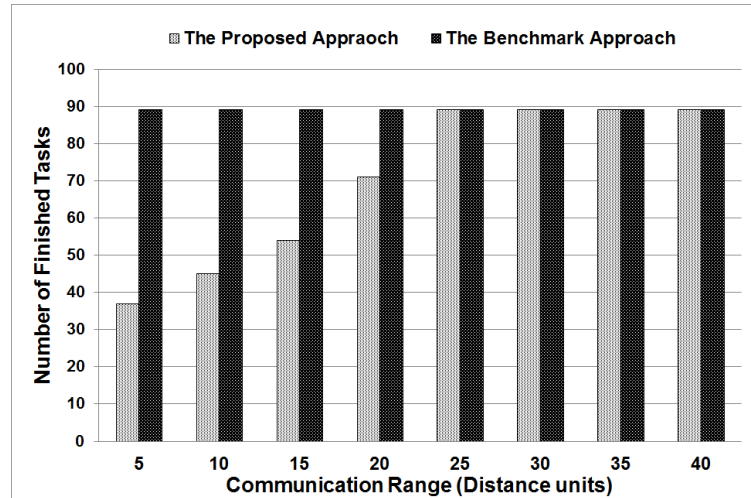


Figure 3.6: The results of Experiment 2

From Figure 3.6, it can be seen that since the communication constraints are not considered when conducting the benchmark approach, the performance of the benchmark approach is constant in this experiment. For the proposed approach, when the communication ranges of agents are very limited (i.e., the communication range is only 5), the performance of the proposed approach on task allocation is about 50% less than that of the benchmark approach. That is because when the communication range is 5, only many small scale isolated groups are formed in the environment. Although the coordinator of each group tends to find the

most suitable allocation solution for its group, the performance of these isolated groups is far away from that of the centralised approach. However, with the increase of communication ranges of agents, the coordinator of each group can connect more and more agents and the performance of the proposed approach is becoming better and better. When the communication ranges of agents are 25 (i.e., agents can communicate with each other in the environment), the performance of the proposed approach is as good as the benchmark approach. That is because all 10 agents in a disaster environment form only one group, so the task allocation result is similar to the benchmark (centralised) approach. Therefore, we can conclude that the benchmark (centralised) approach can achieve the optimal solution than the proposed (decentralised) approach, if the coordinator can have a global view about the environment. However, due to space and communication constraints, it is hard to have such a view in most of disaster environments. For that reason, the applications of the benchmark (centralised) approach in disaster environments are limited. However, the coordinators in the proposed (decentralised) approach can create near optimal solutions under space and communication constraints without the global view of the environment. Therefore, the proposed (decentralised) approach is more suitable to be applied in disaster environments with space and communication constraints than the benchmark (centralised) approach.

### 3.3.3 Experiment 3: Test of the impact of urgent degrees of tasks

This experiment is to evaluate the proposed approach on task allocation under different urgent degrees of workloads of tasks.

#### Experiment settings

Experiment 3 focuses on the evaluation of the impact of urgent degrees of workloads of tasks on the proposed approach. The settings of general disaster environments in Experiment 2 are also employed in this experiment. In order to get rid of the impact of the proposed group formation mechanism on task allocation, the communication ranges of agents are fixed to 50 in this experiment. The settings of this experiment are described in Table 3.3.

Table 3.3: The settings of Experiment 3

Name	Value
Area size	$50 \times 50 \text{ cm}^2$
Number of tasks	100
Deadlines of tasks	$5 \sim 200$
Workloads of tasks	$10 \sim 50$
Urgent degrees of workloads of tasks	5 or $1 \sim 9$
Number of agents	10
Work efficiencies of agents	1
Moving speeds of agents	1
Communication ranges of agents	$50 \text{ cm}$

In Experiment 3, the performances of the proposed approach in two kinds of environments are compared to demonstrate the impact of the urgent degrees of workloads of tasks.

**Environment 1:** the urgent degrees of workloads of all tasks are same and equal to 5.

**Environment 2:** the urgent degrees of workloads of all tasks vary from 1 to 9 with the normal distribution.

The sitting of the first environment is equivalent to the situation when the proposed approach does not consider the urgent degrees of workloads of tasks.

#### Experimental results and analysis

The results of Experiment 3 are shown in Figures 3.7 and 3.8. In Figure 3.7, the X-axis is the consumed time units, while Y-axis is the sum of the workloads of tasks that have been finished. In Figure 3.8, the X-axis is the consumed time units, while Y-axis is the sum of the weighted workloads of tasks that have been finished.

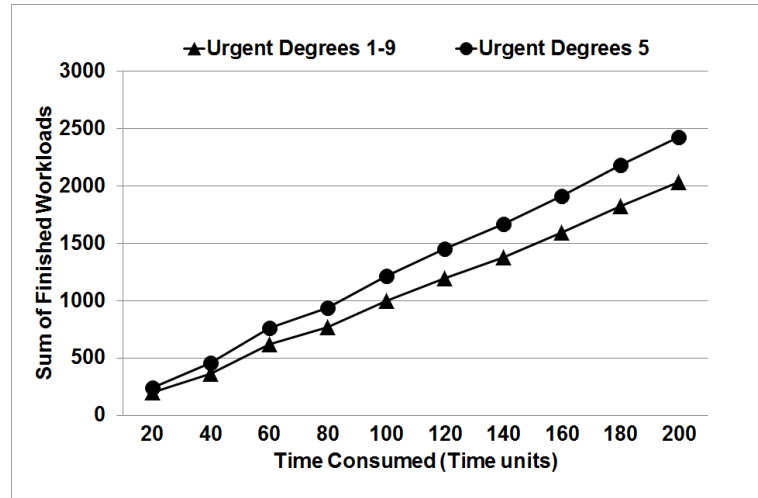


Figure 3.7: Sum workload

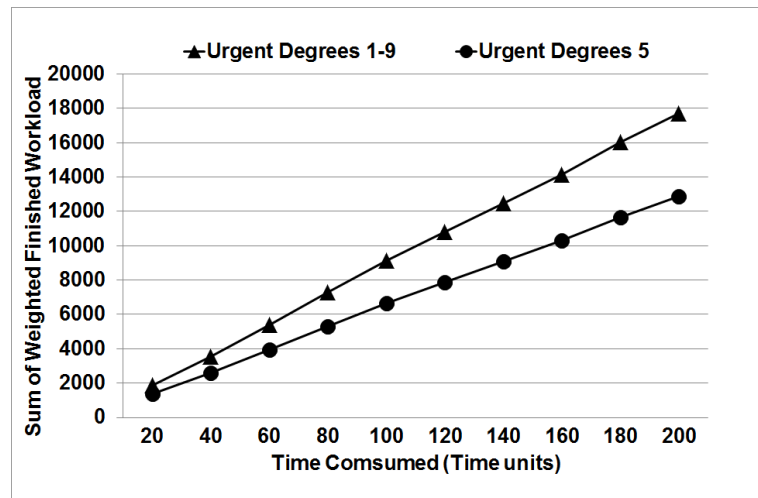


Figure 3.8: Sum weighted workload

From Figures 3.7 and 3.8, it can be seen that the proposed approach can finish more workload, when the urgent degrees of workloads of all tasks are same. This is because when the urgent degrees of workloads of tasks have no difference, the coordinator allocates agents to tasks only based on the workload of each task, which equals to the situation that the workloads of tasks do not have urgent degrees. However, when the urgent degrees of workloads of tasks are different, the proposed approach can finish more weighted workloads than the situation when the urgent degrees of tasks are the same. The reason behind this is that the proposed approach considers the urgent degrees of workloads of tasks in its utility calculation function. Therefore, if we differentiate the urgent degrees of workloads of tasks in an environment, the proposed approach can allocate more urgent tasks before less urgent

tasks. This result is encouraging when allocating urgent tasks in disaster environments.

## 3.4 Summary

From experiments, it can be discovered that

1. by employing the INGF mechanism and token passing mechanism, each agent can connect more neighbouring agents and share more information for task allocation in disaster environments under space and communication constraints (to achieve **Objective 2**, see Section 1.3);
2. by employing the utility calculation mechanism, coordinators can create the task allocation solution as good as the solution created by the centralised task allocation approach under time, space and communication constraints (to achieve **Objective 3**, see Section 1.3); and
3. by employing the utility calculation mechanism, agents can finish more weighted workloads of tasks (to achieve **Objective 4**, see Section 1.3).

In this chapter, a coordinated task allocation approach for weighted tasks in disaster environments with time, space and communication constraints was proposed. First, the problem description and definitions of the coordinated task allocation approach in disaster environments were given. Then, the process and the five steps of the coordinated task allocation approach were introduced in detail. Finally, experiments to evaluate the performance of the coordinated task allocation approach were demonstrated and analysed.

# Chapter 4

---

## A Dynamic Task Allocation Approach

A dynamic task allocation approach is proposed for heterogeneous agents in this chapter. The proposed approach consists of an information collection mechanism, a group task allocation mechanism and a group coordination mechanism.

- The information collection mechanism is developed to help agents to reduce their connections and select network leaders to collect information for task allocation in communication networks by considering communication constraints of disaster environments (to solve **Challenging Issue 2**, identified in Section 1.2);
- The group task allocation mechanism is developed to help network leaders to allocate tasks and agents of their communication network to groups with suitable space ranges by considering space and communication constraints as well as different capabilities of agents (to solve **Challenging Issues 3 and 5**, identified in Section 1.2); and
- The group coordination mechanism is developed to help groups to achieve dynamic coordination in disaster environments by considering communication constraints (to solve **Challenging Issue 3**, identified in Section 1.2).

### 4.1 Problem Description and Definitions

In general, there is a set of agents in task allocation problems, which can be described as  $\{A_1, A_2, \dots, A_i, \dots, A_m\}$ . A task can only be discovered by the agents close to its location and is numbered as  $T_{(i,j)}$ , where  $T_{(i,j)}$  represents the  $j^{th}$  task discovered by  $A_i$ . The terms of an agent and a task were defined in Definitions 3.1 and 3.2 in Chapter 3. To fit the task allocation problem solved in this chapter, they are redefined in this section as follows.

**Definition 4.1.** An *Agent* ( $A_i$ ) can be defined as a four-tuple  $A_i = \langle Uti_i, Loc_i, MSp_i, \vec{Cap}_i \rangle$ , where  $Uti_i$  is the work efficiency of  $A_i$ , which represents how many units of workload that  $A_i$  can perform per time unit;  $Loc_i$  is the current location of  $A_i$ ;  $MSp_i$  is the moving speed

of  $A_i$ , which represents how many units of distance that  $A_i$  can move per time unit; and  $\vec{Cap}_i$  is the capabilities of  $A_i$ , which can be described as a vector  $\vec{Cap}_i = (c_i^1, c_i^2, \dots, c_i^R)$ , where  $c_i^r$  is the indicator of the  $r^{th}$  capability, which indicates that whether  $A_i$  has the  $r^{th}$  capability, if  $A_i$  has the  $r^{th}$  capability,  $c_i^r = 1$ , otherwise,  $c_i^r = 0$ .

**Definition 4.2.** A Task ( $T_{(i,j)}$ ) represents the  $j^{th}$  task discovered by  $A_i$ , which can be defined as a four-tuple  $T_{(i,j)} = \langle DLine_{(i,j)}, WLoad_{(i,j)}, Loc_{(i,j)}, R\vec{Cap}_{(i,j)} \rangle$ , where  $DLine_{(i,j)}$  is the deadline of  $T_{(i,j)}$  (e.g., the time point until which the survivor can remain alive or a building remain standing), where  $DLine_{(i,j)} \in [0, \infty]$ ;  $WLoad_{(i,j)}$  is the workload of  $T_{(i,j)}$ , which represents how many units of workload must be done to complete  $T_{(i,j)}$ ;  $Loc_{(i,j)}$  is the location of  $T_{(i,j)}$ ; and  $R\vec{Cap}_{(i,j)}$  is the capabilities requirement of  $T_{(i,j)}$ , which can be described as a vector  $R\vec{Cap}_{(i,j)} = (c_{(i,j)}^1, c_{(i,j)}^2, \dots, c_{(i,j)}^R)$ , where  $c_{(i,j)}^r$  is the requirement indicator of the  $r^{th}$  capability, which indicates whether  $T_{(i,j)}$  requires the  $r^{th}$  capability to finish, if  $T_{(i,j)}$  requires the  $r^{th}$  capability to finish,  $c_{(i,j)}^r = 1$ , otherwise,  $c_{(i,j)}^r = 0$ .

The objective of task allocation in disaster environments is to maximize the workload of finished tasks, which can be described as follows.

$$Objective = \max \sum_{\forall T_{(i,j)}} Finish(T_{(i,j)}) \times WLoad_{(i,j)}, \quad (4.1)$$

where  $Finish(T_{(i,j)})$  is a Boolean-return function, if  $T_{(i,j)}$  is finished before its deadline (i.e.,  $DLine_{(i,j)}$ , see Definition 4.2),  $Finish(T_{(i,j)}) = 1$ , otherwise,  $Finish(T_{(i,j)}) = 0$ ; and  $WLoad_{(i,j)}$  is the workload of  $T_{(i,j)}$  (see Definition 4.2).

In disaster environments, each agent can only directly communicate with other agents within its communication range. In the proposed approach, we assume that the communication ranges of all agents in the environment are the same and equal to  $CR$  (units of distance). Based on the locations and communication ranges of agents, three types of communication relationships between agents can be defined as follows.

**Definition 4.3.** The *direct neighbours* of an agent ( $A_i$ ) are the agents within the communication range of  $A_i$  (i.e., the Euclidean distances [51] between  $A_i$  and its *direct neighbours* cannot be more than  $CR$ ), and  $A_i$  can directly communicate with its *direct neighbours*.

**Definition 4.4.** The *indirect neighbours* of  $A_i$  are the agents beyond its communication range (i.e., the Euclidean distances between  $A_i$  and its *indirect neighbours* are more than  $CR$ ), however,  $A_i$  can exchange information with its *indirect neighbours* through other agents (i.e., there are communication pathes between  $A_i$  and its *indirect neighbours*).

**Definition 4.5.** The *isolated agents* of  $A_i$  are the agents beyond its communication range (i.e., the Euclidean distances between  $A_i$  and its *isolated agents* are more than  $CR$ ), and  $A_i$  cannot exchange information with its *isolated agents* through other agents (i.e., there are no communication pathes between  $A_i$  and its *isolated agents*).

Examples of the three types of communication relationships are shown in Figure 4.1.

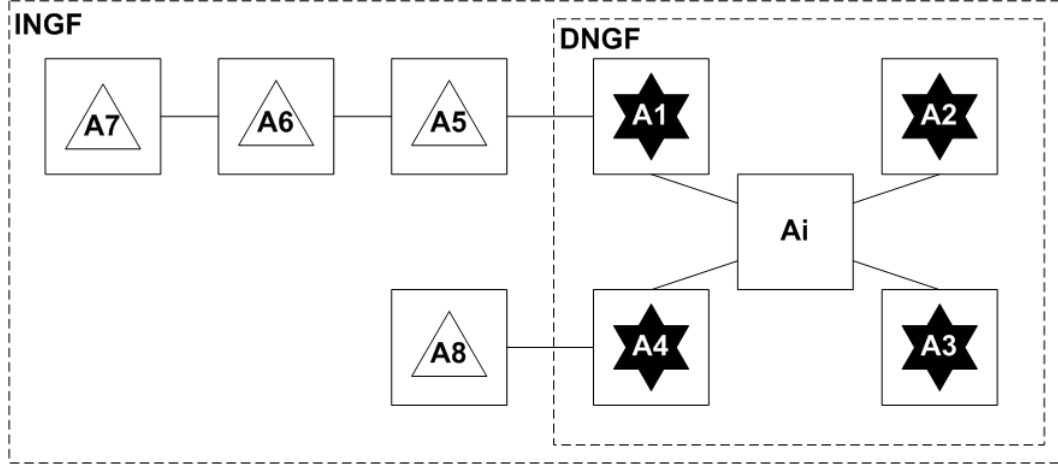


Figure 4.1: The communication relationships among agents

In Figure 4.1, the rectangles represent agents and solid lines between agents indicate that the two agents are within communication range of each other. Direct neighbours of  $A_i$  are  $A_1$  and  $A_2$  (stars), indirect neighbours of  $A_i$  are  $A_3$  to  $A_6$  (triangles) and isolated agents of  $A_i$  are  $A_7$  and  $A_8$  (squares).

In the proposed approach, tasks and agents are allocated to groups, each of which includes a set of tasks and a set of agents in charge of finishing the tasks. The definition of the group information of a group is given as follows.

**Definition 4.6.** The *Group Information* ( $GInf_k$ ) of a group ( $G_k$ ) can be defined as a five-tuple  $GInf_k = \langle TSet_k, UTSets_k, ASet_k, IASets_k, Rep_k \rangle$ , where  $TSet_k$  is the set of tasks of  $G_k$ ;  $UTSets_k$  is the set of unfinished tasks of  $G_k$ , where  $UTSets_k \subseteq TSet_k$ ;  $ASet_k$  is the set of agents of  $G_k$ ;  $IASets_k$  is the set of idle agents of  $G_k$ , where  $IASets_k \subseteq ASet_k$ ; and  $Rep_k$  is the representative agent of  $G_k$ , which is in charge of coordination with other groups.

In the proposed approach, groups periodically coordinate with each other at assembly points and one round of coordination is finished in  $TP$  time units. In the proposed approach, there are two types of coordination: the top-layer and the bottom-layer. These are carried out at two types of assembly points, i.e., the assembly point of the environment, and assembly points of the network.



**Definition 4.7.** The assembly point of the environment ( $AP_e$ ) is defined as the only location in a disaster environment, which is set for the top-layer coordination beforehand and can be well known by agents in the environment.

**Definition 4.8.** An assembly point of the network ( $AP_{n_p}$ ) is defined as a location in a disaster environment, which is set during the emergency for bottom-layer coordination of groups from the same communication network.

## 4.2 The Principle of the Dynamic Task Allocation Approach

The proposed approach consists of following three mechanisms, which are

- 1) the information collection mechanism;
- 2) the group task allocation mechanism; and
- 3) the group coordination mechanism.

The basic process of the proposed approach is shown in Figure 4.2.

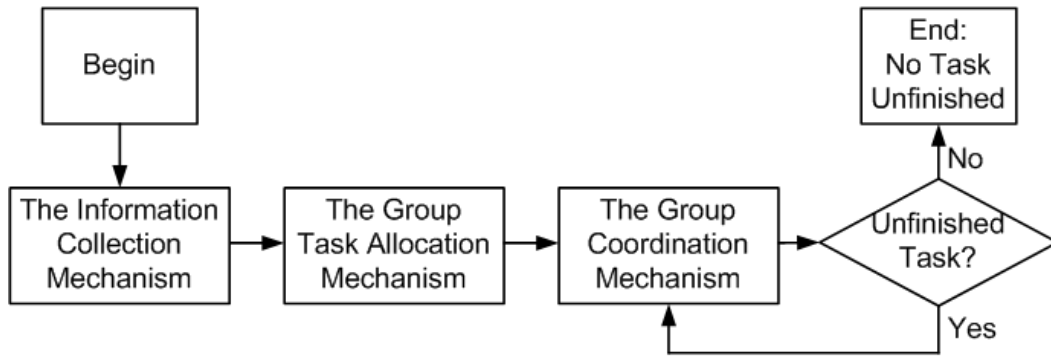


Figure 4.2: The basic process of the dynamic task allocation approach

The detailed process of the dynamic task allocation approach is described in Algorithm 4.1.

**Algorithm 4.1:** The process of dynamic task allocation approach

- 
- 1 Agents collect information for task allocation in a disaster environment
  - 2 Agents select network leaders to collect information for task allocation (i.e.,  $WLoad_{(i,j)}$ ,  $Loc_{(i,j)}$ ,  $RCap_{(i,j)}$ ,  $Uti_i$ ,  $Loc_i$ ,  $MSp_i$ , and  $Cap_i$ ) based on the information collection mechanism
  - 3 Network leaders allocate tasks and agents into different groups based on the group task allocation mechanism.
  - 4 **while** *There are available tasks in the environment* **do**
  - 5     Groups of agents periodically coordinate with each other based on the group coordination mechanism
  - 6 **end**
- 

Algorithm 4.1 is explained as follows. After collected information in a disaster environment, the information collection mechanism helps agents to reduce their connections and select an agent in each communication network as the network leader to collect information collection for task allocation. Based on this network, agents can pass the information about the tasks near them (i.e., workloads ( $WLoad_{(i,j)}$ ), locations ( $Loc_{(i,j)}$ ), and required capabilities ( $RCap_{(i,j)}$ ) of tasks, see Definition 4.2) and their own status (i.e., the work efficiency ( $Uti_i$ ), location ( $Loc_i$ ), moving speed ( $MSp_i$ ), and capabilities ( $Cap_i$ ), see Definition 4.1) to the network leader (Lines 1 to 2). Then, according to the group task allocation mechanism, each network leader first allocates tasks to isolated groups with suitable space ranges according to the task locations of tasks and then, allocates agents to suitable groups according to the capabilities of each agent (Line 3). The information collection mechanism and the group task allocation mechanism are performed for only once and network leaders are dismissed after performing the group task allocation mechanism. During task execution, due to the dynamics of disaster environments, the original allocation (by the group task allocation mechanism) of tasks and agents to groups may be unsuitable, where agents in some groups might finish their tasks and are idle, while agents in other groups are still working on unfinished tasks. In order to reallocate group members (heterogeneous agents) and achieve continuous coordination of isolated groups, the group coordination mechanism is periodically (in every  $TP$  time units) employed to enable the representative agent (i.e.,  $Rep_k$ , see Definition 4.6) of each group to use its latest group information (i.e.,  $GInf_k$ , see Definition 4.6) and coordinate with representative agents of other groups at the assembly point. The group coordination mechanism is continuously employed by groups during task execution until all tasks have been completed (Lines 4 to 6).

### 4.2.1 The information collection mechanism

The objective of the information collection mechanism is to help agents to reduce their communication connections and elect a network leader to collect information for task allocation in a decentralised manner, during which agents in the communication network eliminate all their communication connections except the one leading to the network leader. Therefore, after applying the information collection mechanism for a communication network with  $m$  number of agents, only  $m - 1$  number of connections are kept in the network. By doing so, each agent only needs to pass the information it has to the network leader through its only direct neighbour so as to reduce the overhead information exchange for task allocation.

In the information collection mechanism, three neighbour-related parameters are defined for each agent. These are the parent agent (represented by  $PA$ ), the network leader (represented by  $L$ ) and the number of direct neighbours of the network leader (represented by  $NNL$ ). For example, for an agent  $A_i$ , three neighbour-related parameters can be denoted as  $A_i.PA$ ,  $A_i.L$  and  $A_i.NNL$ , respectively. The information collection mechanism is described in Algorithm 4.2.

---

**Algorithm 4.2:** The information collection mechanism

---

```

1 for each agent (e.g.,  $A_i$ ) do
2    $A_i.PA \leftarrow A_i$ ;  $A_i.L \leftarrow A_i$ ;  $A_i.NNL \leftarrow$  the number of direct neighbours of  $A_i$ 
3   Broadcasts its three neighbour-related parameters
4 end
5 for each agent (e.g.,  $A_i$ ) do
6   Gets  $A_u$  from its received parameters, where  $A_u.NNL$  is maximum
7   if  $A_u.NNL > A_i.NNL$  then
8      $A_i.PA \leftarrow A_u$ ;  $A_i.L \leftarrow A_u.L$ ;  $A_i.NNL \leftarrow A_u.NNL$ 
9     Broadcasts its new neighbour-related parameters
10  end
11 end

```

---

At the beginning of Algorithm 4.2, each agent (e.g.,  $A_i$ ) initialises its three neighbour-related parameters as follows:  $A_i.PA$  is set to  $A_i$ ,  $A_i.L$  is set to  $A_i$  and  $A_i.NNL$  is set to the number of direct neighbours of  $A_i$  and broadcasts its initialised parameters to its direct neighbours (Lines 1 to 3). When an agent (e.g.,  $A_i$ ) receives parameters from its direct neighbours, it repeats the following two steps.

**Step 1:**  $A_i$  finds the agent (e.g.,  $A_u$ ) with the highest value of  $NNL$  (Lines 5 and 6); and

**Step 2:** If  $A_u.NNL$  is higher than the current value of  $A_i.NNL$ ,  $A_i$  updates its three neighbour-related parameters as follows:  $A_i.PA$  is set to  $A_u$ ,  $A_i.L$  is set to  $A_u.L$

and  $A_i.NNL$  is set to  $A_u.NNL$  and broadcasts its updated parameters to its direct neighbours (Lines 7 to 9).

The above two steps (Lines 5 to 9) will be repeated by each agent until no further updating for three neighbour-related parameters of any agent.

In the final stage of the information collection mechanism, each agent passes the information about its nearby tasks and status to its network leader (i.e.,  $A_i.L$ ) through its parent agent (i.e.,  $A_i.PA$ ) set in the mechanism. In addition, since a task can be discovered by multiple agents, agents need to identify superfluous information about tasks and not pass that information. In addition, if there are isolated agents in an environment, more than one communication networks exists in the environment and more than one network leaders are selected.

## 4.2.2 The group task allocation mechanism

The group task allocation mechanism helps each network leader to allocate tasks and agents within its network to groups with suitable space ranges under the consideration of task allocation and execution.

The group task allocation mechanism is executed by each network leader and this include three steps: **1)** task allocation, **2)** agent allocation and **3)** setting the assembly point of the network.

### 1) Task allocation

In the task allocation step, the tasks of each network are allocated to groups with suitable space ranges according to the task locations (i.e.,  $Loc_{(i,j)}$ , see Definition 4.2) and the communication range of agents (i.e.,  $CR$ , see Definition 4.3). The objective of this step is to restrict the space range of each group so that the travelling ranges of agents can be reduced so that ensure that agents working in the same group can always communicate with each other. By doing so, the centralised task allocation approaches can be employed by agents within each group [79], [113].

To achieve these objectives, the mean-shift algorithm [25] is employed by each network leader to allocate tasks. The mean-shift algorithm is a grouping algorithm, which is widely applied to data mining, pattern recognition etc [148], [20]. The only parameter of the mean-shift algorithm  $h$  represents the radius of the window, which decides the space ranges of groups. In order to enable agents working in the same group to always communicate with

each other,  $h$  is set equal to  $CR$  (see Definition 4.3). For an  $n$  tasks grouping problem in a 2-dimensional space, the multi-kernel density function can be calculated as follows.

$$f(x) = \frac{1}{n \cdot CR^2} \sum_{\forall T_{(i,j)} \in window} K\left(\frac{x - Loc_{(i,j)}}{CR}\right), \quad (4.2)$$

where  $K(x)$  is the kernel function,  $Loc_{(i,j)}$  is the location of  $T_{(i,j)}$  within the window,  $x$  is the centre (mean) of a window. In the proposed approach,  $K(x)$  can be described by the Euclidean distance between the centre (mean) of the window (i.e.,  $x$ ) and the location of  $T_{(i,j)}$  in the window (i.e.,  $Loc_{(i,j)}$ , see Definition 4.2). Based on the multi-kernel density function, the centre (mean) of the window always moves to the point with the greatest density value.

After employing the mean-shift grouping algorithm, tasks in each communication network are allocated. For example, 50 tasks are discovered in a  $50 \times 50$  environment. When  $CR$  is 15, groups allocated by the mean-shift algorithm are shown in Figures 4.3 and 4.4.

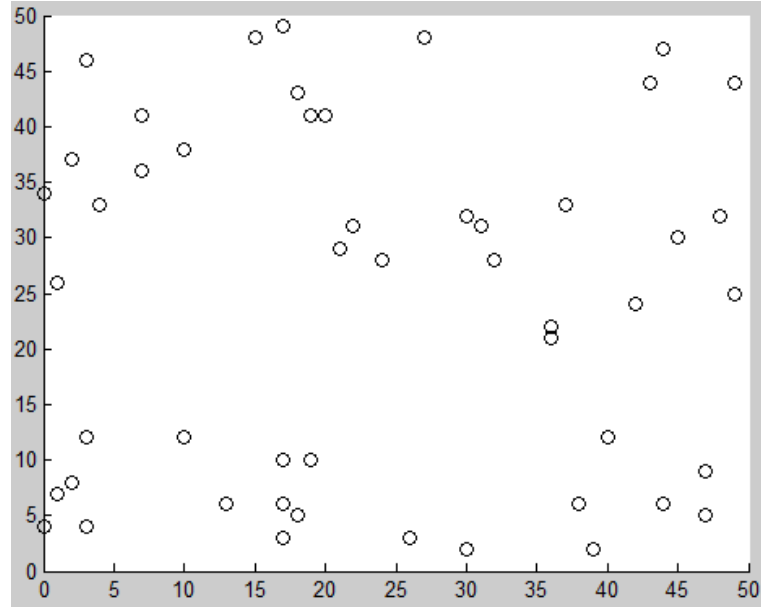


Figure 4.3: Locations of tasks before employing the mean-shift algorithm

In Figure 4.3, locations of tasks in the environment are represented by circles.

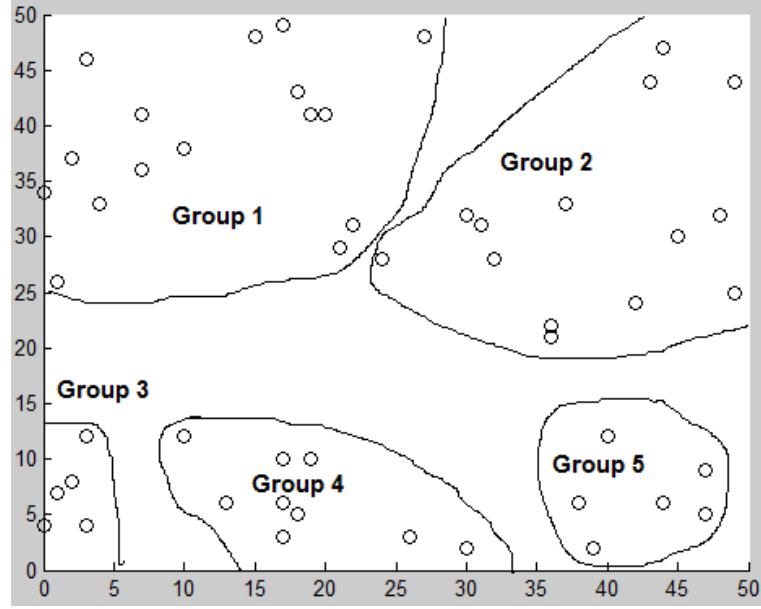


Figure 4.4: Groups of tasks after employing the mean-shift algorithm

In Figure 4.4, groups of tasks are represented after employing the mean-shift algorithm.

## 2) Agent allocation

In the agent allocation step, heterogeneous agents are allocated to suitable groups according to the missed capabilities requirements of tasks allocated to each group (e.g.,  $G_k$ ) and capabilities of each unallocated agent (e.g.,  $A_u$ ). In order to identify what is the missed capabilities requirements of tasks allocated to  $G_k$ , we first identify the capabilities requirements of tasks (i.e.,  $T_{(i,j)} \in G_k$ ) and capabilities of agents (i.e.,  $A_i \in G_k$ ) allocated to  $G_k$ , which can be calculated as follows.

$$\begin{aligned}
 R\vec{C}ap_k &= \sum_{T_{(i,j)} \in G_k} R\vec{C}ap_{(i,j)} \\
 &= \left( \sum_{T_{(i,j)} \in G_k} c_{(i,j)}^1, \sum_{T_{(i,j)} \in G_k} c_{(i,j)}^2, \dots, \sum_{T_{(i,j)} \in G_k} c_{(i,j)}^R \right),
 \end{aligned} \tag{4.3}$$

where  $R\vec{C}ap_{(i,j)}$  is the vector of the capabilities requirement of  $T_{(i,j)}$  allocated to  $G_k$  and  $c_{(i,j)}^r$  is the requirement indicator of the  $r^{th}$  capabilities of  $T_{(i,j)}$  allocated to  $G_k$  (see Definition 4.2).

$$\begin{aligned}
AC\vec{ap}_k &= \sum_{A_i \in G_k} C\vec{ap}_i \\
&= \left( \sum_{A_i \in G_k} c_i^1, \sum_{A_i \in G_k} c_i^2, \dots, \sum_{A_i \in G_k} c_i^R \right),
\end{aligned} \tag{4.4}$$

where  $C\vec{ap}_i$  is the vector of the capabilities of  $A_i$  allocated to  $G_k$  and  $c_i^r$  is the indicator of the  $r^{th}$  capability of  $A_i$  allocated to  $G_k$  (see Definition 4.1).

Then, the vector of the missed capabilities requirements of tasks allocated to  $G_k$  can be calculated as follows.

$$\begin{aligned}
MR\vec{Cap}_k &= norm(R\vec{Cap}_k) - norm(AC\vec{ap}_k) \\
&= (mc_k^1, mc_k^2, \dots, mc_k^R),
\end{aligned} \tag{4.5}$$

where  $norm(R\vec{Cap}_k)$  and  $norm(AC\vec{ap}_k)$  are the normalised vector of the capabilities requirements of tasks and capabilities of agents allocated to  $G_k$ , respectively; and  $mc_k^r$  is the indicator of the  $r^{th}$  capability, which describes that to what extent  $G_k$  misses the  $r^{th}$  capability, if  $mc_k^r \leq 0$ ,  $mc_k^r = 0$ , otherwise,  $mc_k^r = mc_k^r$ .

Finally, the similarity value between the vector of the missed capabilities requirements of tasks allocated to  $G_k$  and the normalised vector of capabilities of an unallocated agent  $A_u$  (i.e.,  $norm(C\vec{ap}_u) = (nc_u^1, nc_u^2, \dots, nc_u^R)$ ) can be calculated by the *dot product* of two vectors [73], which can be described as follows.

$$Sim_{(k,u)} = \frac{\sum_{r=1}^R (mc_k^r \times nc_u^r)}{\sqrt{(\sum_{r=1}^R (mc_k^r) \times (\sum_{r=1}^R (nc_u^r)^2))}} \tag{4.6}$$

$A_u$  will be allocated to the group with the highest similarity value.

Considering the example that there are two groups (e.g.,  $G_1$  and  $G_2$ ) in an environment, where  $G_1$  has four tasks (i.e.,  $T_{(1,1)}$  to  $T_{(1,4)}$ ) and  $G_2$  has three tasks (i.e.,  $T_{(2,1)}$  to  $T_{(2,3)}$ ). The vectors of capabilities requirements of tasks (i.e.,  $R\vec{Cap}_{(i,j)}$ , see Definition 4.2) allocated to  $G_1$  and  $G_2$  are shown as follows.

Tasks allocated to $G_1$	Tasks allocated to $G_2$
$R\vec{Cap}_{(1,1)} = (0, 1, 1, 0)$	$R\vec{Cap}_{(2,1)} = (1, 1, 0, 0)$
$R\vec{Cap}_{(1,2)} = (0, 1, 1, 0)$	$R\vec{Cap}_{(2,2)} = (1, 0, 0, 1)$
$R\vec{Cap}_{(1,3)} = (0, 0, 1, 0)$	$R\vec{Cap}_{(2,3)} = (1, 0, 0, 0)$
$R\vec{Cap}_{(1,4)} = (0, 0, 1, 0)$	

There are three agents (i.e.,  $A_1$ ,  $A_2$  and  $A_3$ ) in the environment, where  $A_1$  is already allocated to  $G_2$ , while  $A_2$  and  $A_3$  are unallocated. The vectors of capabilities of three agents (i.e.,  $\vec{Cap}_i$ , see Definition 4.1) are shown as follows.

$$\begin{aligned} A_1 : \vec{Cap}_1 &= (1, 0, 0, 0) \\ A_2 : \vec{Cap}_2 &= (0, 1, 1, 0) \\ A_3 : \vec{Cap}_3 &= (1, 0, 0, 1) \end{aligned}$$

First, the vectors of capabilities requirements of tasks allocated to  $G_1$  and  $G_2$  (i.e.,  $\vec{RCap}_1$  and  $\vec{RCap}_2$ ) can be calculated based on Equation 4.3 as follows.

$$\begin{aligned} \vec{RCap}_1 &= \sum_{T(i,j) \in G_1} \vec{RCap}_{(i,j)} \\ &= \vec{RCap}_{(1,1)} + \vec{RCap}_{(1,2)} + \vec{RCap}_{(1,3)} + \vec{RCap}_{(1,4)} \\ &= (0, 2, 4, 0) \\ \vec{RCap}_2 &= \sum_{T(i,j) \in G_2} \vec{RCap}_{(i,j)} \\ &= \vec{RCap}_{(2,1)} + \vec{RCap}_{(2,2)} + \vec{RCap}_{(2,3)} \\ &= (3, 1, 0, 1) \end{aligned}$$

Then, since there is no agent allocated to  $G_1$ , while  $A_1$  is already allocated to  $G_2$ , the vectors of capabilities of agents allocated to  $G_1$  and  $G_2$  (i.e.,  $\vec{ACap}_1$  and  $\vec{ACap}_2$ ) can be calculated based on Equation 4.4 as follows.

$$\begin{aligned} \vec{ACap}_1 &= \sum_{A_i \in G_1} \vec{Cap}_i \\ &= (0, 0, 0, 0) \\ \vec{ACap}_2 &= \sum_{A_i \in G_2} \vec{Cap}_i \\ &= \vec{Cap}_1 \\ &= (1, 0, 0, 0) \end{aligned}$$

After that, the vectors of the missed capabilities requirements of tasks of  $G_1$  and  $G_2$  (i.e.,  $\vec{MRCap}_1$  and  $\vec{MRCap}_2$ ) can be calculated based on Equation 4.5 as follows.

$$\begin{aligned} \vec{MRCap}_1 &= \text{norm}(\vec{RCap}_1) - \text{norm}(\vec{ACap}_1) \\ &= \text{norm}((0, 2, 4, 0)) - \text{norm}((0, 0, 0, 0)) \\ &= (0, 0.4472, 0.8944, 0) \end{aligned}$$



$$\begin{aligned}
MR\vec{C}ap_2 &= norm(R\vec{C}ap_2) - norm(A\vec{C}ap_2) \\
&= norm((3, 1, 0, 1)) - norm((1, 0, 0, 0)) \\
&= (0.9045, 0.3015, 0, 0.3015) - (1, 0, 0, 0) \\
&= (0, 0.3015, 0, 0.3015),
\end{aligned}$$

where,  $mk_1^1 = 0.9045 - 1 = -0.0955 < 0$  so that  $mk_1^1 = 0$ .

Finally, the normalised vectors of capabilities of unallocated agents  $A_2$  and  $A_3$  (i.e.,  $norm(C\vec{a}p_2)$  and  $norm(C\vec{a}p_3)$ ) are  $(0, 0.7071, 0.7071, 0)$  and  $(0.7071, 0, 0, 0.7071)$ , respectively. The similarity values between the vectors of the missed capabilities requirements of tasks allocated to  $G_1$  and  $G_2$  and the normalised vectors of capabilities of unallocated agents  $A_2$  and  $A_3$  are can be calculated based on Equation 4.6, the results of which are shown as follows.

	$G_1$	$G_2$
$A_2$	0.9486	0.2132
$A_3$	0	0.2132

Based on similarity values,  $A_2$  is allocated to  $G_1$  and  $A_3$  is allocated to  $G_2$ .

### 3) Assembly point of the network setting

At the final step of the group task allocation mechanism, one agent in each group is elected as the representative agent of the group (i.e.,  $Rep_k$ , see Definition 4.6) and the assembly point of the network (i.e.,  $APn_p$ , see Definition 4.5) is set for the coordination of groups from the same communication network. In order for representative agents of the groups to arrive at  $APn_p$  at the same time, the location of  $APn_p$  will be set at the centroid [4], [123] of locations of representative agents. After that, the network leader are dismissed and participates in task execution in allocated groups.

## 4.2.3 The group coordination mechanism

During task execution, the original allocation of tasks and agents among groups may be unsuitable, where agents in some groups finish all tasks and are idle, while agents in other groups are still working on unfinished tasks. In addition, due to space and communication constraints, the communication relationships among agents may change during task execution hence it is difficult to employ centralised coordination mechanisms to reallocate group members. To do this, if a group (e.g.,  $G_k$ ) has unfinished tasks or idle agents (i.e.,

$UTSet_k \neq \emptyset$  or  $IASet_k \neq \emptyset$ , see Definition 4.6), the group coordination mechanism is periodically executed by the representative agent (i.e.,  $Rep_k$ , see Definition 4.6) of the group.

One round of coordination can be finished in  $TP$  time units and includes: **1)** the representative agent departs; **2)** the representative agent waits; **3)** the representative agent coordinates; and **4)** the representative agent returns.

**1) The representative agent departs:**

The representative agent (e.g.,  $Rep_k$ ) of each group (i.e.,  $G_k$ ) takes the latest group information (i.e.,  $GInf_k$ , see Definition 4.6) and begins to move to the assembly point.

**2) The representative agent waits:**

Since different representative agents may have different moving speed or different distances to the assembly point, if  $Rep_k$  arrives at the assembly point earlier, it needs to wait for other representative agents until  $\frac{TP}{2}$  time units from its departure.

**3) The representative agent coordinates:**

The representative agents at the assembly point begins to coordinate with each other and updates their  $GInf_k$  accordingly. The detailed process of coordination at the assembly point will be introduced in Subsection 4.2.3.

**4) The representative agent returns:**

$Rep_k$  returns to  $G_k$  and adjusts the work of its group members according to the updated  $GInf_k$ .

**The two-layer coordination structure**

The group coordination mechanism has a two-layer structure, which are the bottom-layer coordination and the top-layer coordination.

• **The bottom-layer coordination:**

The representative agents (i.e.,  $Rep_k$ , see Definition 4.6) of groups from the same communication network first coordinate with each other at the assembly point of the network (i.e.,  $APn_p$ , see Definition 4.8).

• **The top-layer coordination:**

After the coordination at  $APn_p$ , if there still exist unfinished tasks or idle agents in some groups at  $APn_p$  (i.e.,  $UTSet_k \neq \emptyset$  or  $IASet_k \neq \emptyset$ ), one  $Rep_k$  at each  $APn_p$  collects the group information (i.e.,  $GInf_k$ , see Definition 4.6) of them, moves to the assembly point of the environment (i.e.,  $APe$ , see Definition 4.7) and coordinates with representative agents from other assembly points of the network there.

The reason for this two-layer structure is that group members must first be reallocated among the groups from the same communication network. After that, group members can be reallocated among groups in the environment. This is because group members are closer to their assembly points of the network (i.e.,  $AP_{n_p}$ , see Definition 4.8) than to the assembly point of the environment (i.e.,  $AP_e$ , see Definition 4.7). The mechanism enables most of the representative agents to periodically coordinate at their  $AP_{n_p}$  and only some of them need to coordinate at  $AP_e$ , which can reduce the travelling distances of representative agents in order to coordinates the groups.

In addition, when a new agent (e.g.,  $A_i$ ) enters the environment without knowing the locations of groups in the environment, it can move to the assembly point of the environment (i.e.,  $AP_e$ , see Definition 4.7) and waits for a new round of coordination at  $AP_e$ . After coordination at  $AP_e$ ,  $A_i$  can move to the allocated group with the representative agents (i.e.,  $Rep_k$ , see Definition 4.6).

An example of the two-layer group coordination mechanism is shown in Figure 4.5

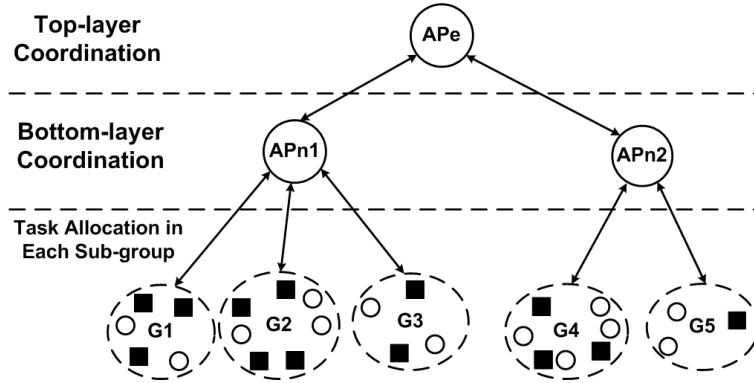


Figure 4.5: The two-layer group coordination mechanism

In Figure 4.5, black squares and white circles represent tasks and agents in groups, respectively.  $AP_e$  is the assembly point of the environment (see Definition 4.7). There are five groups (i.e.,  $G_1$ ,  $G_2$ ,  $G_3$ ,  $G_4$  and  $G_5$ ) in the environment, where  $G_1$ ,  $G_2$ , and  $G_3$  are from the same communication network and their assembly point of the network is  $AP_{n_1}$  (see Definition 4.8);  $G_4$  and  $G_5$  are from the same communication network and their assembly point of the network is  $AP_{n_2}$  (see Definition 4.8). During task execution, the representative agents of  $G_1$ ,  $G_2$  and  $G_3$  (i.e.,  $Rep_1$ ,  $Rep_2$  and  $Rep_3$ ) first coordinate with each other at  $AP_{n_1}$ . At the same time,  $Rep_4$  and  $Rep_5$  coordinate with each other at  $AP_{n_2}$ . After coordination at  $AP_{n_1}$  and  $AP_{n_2}$  (i.e., the bottom-layer coordination), one  $Rep_k$  at each  $AP_{n_p}$  (e.g.,  $Rep_1$  at  $AP_{n_1}$  and  $Rep_4$  at  $AP_{n_2}$ ) collects any remaining unfinished tasks and idle agents and

moves to  $APe$  to coordinate with each other. At the same time, other representative agents (i.e.,  $Rep_2$ ,  $Rep_3$  and  $Rep_5$ ) return to their groups (i.e.,  $G_2$ ,  $G_3$  and  $G_5$ ) and reallocate their group members. After coordination at  $APe$  (i.e., top-layer coordination) and adjustment in  $G_2$ ,  $G_3$ , and  $G_5$ ,  $Rep_1$  and  $Rep_4$  return to  $APn_1$  and  $APn_2$ , respectively from  $APe$ .  $Rep_2$  and  $Rep_3$  move to  $APn_1$  and  $Rep_5$  moves to  $APn_2$  from their groups. When  $Rep_1$ ,  $Rep_2$  and  $Rep_3$  arrive at  $APn_1$  and  $Rep_4$  and  $Rep_5$  arrive at  $APn_2$ , a new round of coordination at  $APn_1$  and  $APn_2$  begin (i.e., the bottom-layer coordination), respectively. After coordination, another one  $Rep_k$  at each  $APn_p$  (e.g.,  $Rep_2$  at  $APn_1$  and  $Rep_5$  at  $APn_2$ ) moves to  $APe$  to coordinate with each other. At the same time, other representative agents (i.e.,  $Rep_1$ ,  $Rep_3$  and  $Rep_4$ ) return to their group (i.e.,  $G_1$ ,  $G_3$  and  $G_4$ ) and adjust task allocation of their group members. The group coordination mechanism is continuously executed by groups during task execution until there is no unfinished task in the environment.

### Coordination at the assembly point

The coordination of groups can be achieved through reallocation of group members, which can allocate suitable idle agents in some groups to unfinished tasks in other groups. The suitability of an idle agent (e.g.,  $A_i$ ) allocated to an unfinished task (e.g.,  $T_{(i,j)}$ ) involves two aspects: **1**) whether the idle agent can finish the unfinished task before its deadline (i.e.,  $DLine_{(i,j)}$ , see Definition 4.2), and **2**) whether the capabilities of the idle agent fit the capabilities requirement of the unfinished task. The first aspect can be evaluated based on the estimated finishing time ( $Est_{((i,j),i)}$ ) of the unfinished task, which can be calculated as follows.

$$Est_{((i,j),i)} = (CTime + \frac{TP}{2} + \frac{Dis(Loc_{(i,j)}, Loc_i)}{Msp_i} + \frac{WLoad_{(i,j)}}{Uti_i}), \quad (4.7)$$

where  $CTime$  is the current time point;  $TP$  is the period of one round of coordination;  $Dis(Loc_{(i,j)}, Loc_i)$  is the distance between location of the unfinished task  $T_{(i,j)}$  and location of  $A_i$ ;  $Msp_i$  is the moving speed of  $A_i$  (see Definition 4.1);  $WLoad_{(i,j)}$  is the workload of  $T_{(i,j)}$  (see Definition 4.2); and  $Uti_i$  is the working efficiency of  $A_i$  (see Definition 4.1). The second aspect can be evaluated through the similarity value between the vectors of the required capabilities of the unfinished task (i.e.,  $RCap_{(i,j)}$ , see Definition 4.2) and the capabilities of the idle agent (i.e.,  $Cap_i$ , see Definition 4.1), which can be described as the dot product of two vectors (see Equation 4.6).

Many task allocation mechanisms can be employed by agents to reallocate group members. The Contract-Net Protocol (CNP) [132] is employed in our coordination mechanism.

The process of the CNP-based coordination mechanism to reallocate group members is described in Algorithm 4.3.

---

**Algorithm 4.3:** CNP based coordination at an assembly point

---

```

1 for each  $Rep_k$  whose  $UTSet_k \neq \emptyset$  do
2   Broadcasts each  $T_{(i,j)} \in UTSet_k$ 
3 end
4 for each  $Rep_u$  whose  $IASet_u \neq \emptyset$  do
5   for each received  $T_{(i,j)}$  do
6     for each  $A_i \in IASet_u$  do
7       Calculates  $Est_{((i,j),i)}$  for  $T_{(i,j)}$  conducted by  $A_i$ 
8       if  $Est_{((i,j),i)} \leq DLine_{(i,j)}$  then
9         Calculates  $Sim_{((i,j),i)}$  between  $RCap_{(i,j)}$  of  $T_{(i,j)}$  and  $Cap_i$  of  $A_i$ 
10      end
11    end
12    Records  $A_i$  with  $Max(Sim_{((i,j),i)})$  in  $RRes_u$ 
13  end
14  Sends  $RRes_u$  to  $Rep_k$ .
15 end
16 for each  $Rep_k$  whose  $UTSet_k \neq \emptyset$  do
17   for each  $T_{(i,j)} \in UTSet_k$  do
18     chooses suitable  $A_i$  in  $RRes_u$  that can finish  $T_{(i,j)}$  and informs  $Rep_u$ 
19      $Rep_k$  and  $Rep_u$  updates  $GInf_k$  and  $GInf_u$ , respectively.
20   end
21 end

```

---

At the initial stage, the representative agent (e.g.,  $Rep_k$ ) of each group (e.g.,  $G_k$ ) with unfinished tasks (i.e.,  $UTSet_k \neq \emptyset$ , see Definition 4.6) broadcasts its unfinished tasks (i.e.,  $\forall T_{(i,j)} \in UTSet_k$ ) to other representative agents at the assembly point (Lines 1 and 2). When a representative agent (e.g.,  $Rep_u$ ) of a group (e.g.,  $G_u$ ) receives unfinished tasks (e.g.,  $T_{(i,j)} \in UTSet_k$ ) and  $G_u$  has idle agents (i.e.,  $A_i \in IASet_u$ ),  $Rep_u$  first calculates the estimated finish time of each received  $T_{(i,j)}$  conducted by each idle  $A_i$  (i.e.,  $Est_{((i,j),i)}$ , see Equation 4.7) (Lines 4 to 7). If a received  $T_{(i,j)}$  can be finished by an idle  $A_i$  before its deadline, the similarity value (i.e.,  $Sim_{((i,j),i)}$ ) between the required capabilities of the received  $T_{(i,j)}$  and the capabilities of idle  $A_i$  is calculated (Lines 8 and 9). For each  $T_{(i,j)}$  records the  $A_i$  with the highest  $Sim_{((i,j),i)}$  in resource response  $RRes_u$ , which is sent back to  $Rep_k$  (Lines 12 to 14). After receiving all resource responses,  $Rep_k$  chooses the most suitable  $A_i$  for each  $T_{(i,j)}$  and informs the owner (i.e.,  $Rep_u$ ) of each  $A_i$  (Lines 16 to 18).  $Rep_k$  and  $Rep_u$  update  $Sglnf_k$  and  $GInf_u$ , respectively (Line 19).

## 4.3 Experiments and Analysis

Two experiments have been conducted to evaluate the performance of the proposed approach. **Experiment 1** is to evaluate the performance of the information collection mechanism under different communication ranges of agents.

**Experiment 2** is to evaluate the performance of task allocation under different communication ranges of agents.

The same as experiments in the approach proposed by Rmchurn et al. [113], we also borrow some settings (i.e., heterogeneous agents and dynamic environments) from the RCR platform [75] and develop programs in Matlab2010 [57] to conduct the two experiments of the proposed approach.

### 4.3.1 Experiment 1: test of information collection

The purpose of this experiment is to evaluate the performance of the proposed approach on information collection under different communication ranges of agents. The difference between information collection in the proposed approach and the max-sum-based approach is that the proposed approach reduces communication connections of agents in the communication network and selects a network leader for information collection, while the max-sum based approach directly use the communication connections of agents for information exchange. Therefore, the performance of the information collection mechanism has a great effect on the resulting performance of the proposed approach.

#### Experiment settings

The setting of Experiment 1 is described in Table 4.1.

Name	Value
Area size	$50 \times 50$
Number of agents	20
Communication ranges of agents ( $CR$ )	10, 20, 30, 40, 100

Table 4.1: The setting of Experiment 1

In Experiment 1, 20 agents with the same communication ranges are randomly distributed in a  $50 \times 50$  area. The Euclidean distance is employed to evaluate the distance between the locations of two agents [29], [96]. If the distance between these locations is

no more than the communication range of agents, these two agents are direct neighbours of each other (see Definition 4.3). Experiment 1 is conducted when the communication ranges of all agents are 10, 20, 30, 40 and 100, respectively.

The objective of the information collection mechanism is to reduce the communication connections of agents in the communication network so that the total number of communication connections of agents in the communication network before (Before Mechanism) and after (After Mechanism) applying the information collection mechanism is an indicator of Experiment 1. In addition, the communication overhead (i.e., broadcast times) spent on the information collection mechanism is another indicator of Experiment 1. In order to obtain the values of two indicators, the program can output topologies of communication networks before and after applying the information collection mechanism and has a counter to calculate how many broadcasts that agents sent during the information collection mechanism.

### Experimental results and analysis

The results of Experiment 1 are shown in Figure 4.6. In Figure 4.6, the X-axis is the communication ranges of agents. The Y-axes are the total number of communication connections and the communication overhead (broadcast times) spent for the information collection mechanism, which are represented by the bars and lines, respectively.

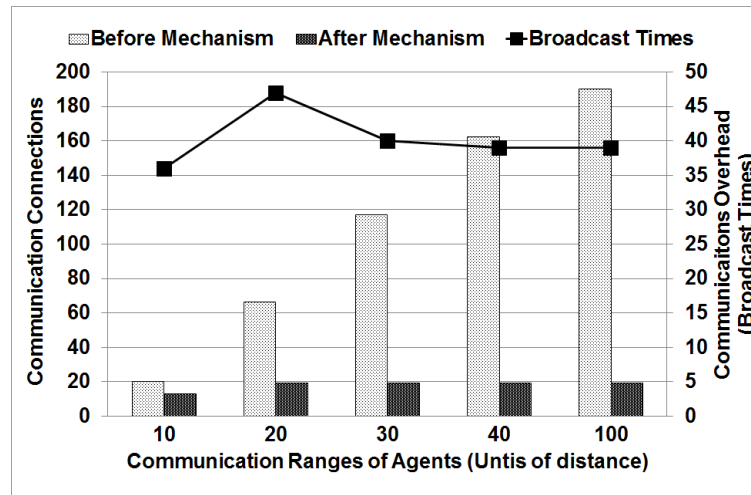


Figure 4.6: The results of Experiment 1

From Figure 4.6, it can be seen that when the communication range of agents is very limited (i.e., 10 units of distance), after employing the information collection mechanism, the total number of communication connections in the communication network does not greatly decrease. That is because due to the limited communication range, each agent only has one

or two communication connections initially. However, with the increase of the communication range, the total number of communication connections increases and the reducing of communication connections in the communication network after employing the information collection mechanism becomes more and more obvious. Since the information collection mechanism can help agents to reduce their communication connections by retaining only the connection leading to the network leader, for an  $m$  agents communication network, the number of communication connections after employing the information collection mechanism is  $m - 1$  (i.e., each agent keep one communication connection and the network leader does not retain a communication connection).

According to the information collection mechanism (i.e., Algorithm 4.2), an agent will broadcast its three neighbour-related parameters, only when it finds a new network leader (i.e., a new agent with the maximum direct neighbours, Line 7 in Algorithm 4.2). The communication overhead of the information collection mechanism is only related to the number of times that agents change their network leaders. When the communication range of agents is very limited (i.e., 10 units of distances), the structures of the networks are simple. Agents can easily find their network leaders without a lot of communications. When the communication range of agents is limited (i.e., 20 units of distance), the structures of the networks are complex and each agent can only have partial information about its communication network. In the such situation, agents need a lot of communications to exchange information with other agents in the same communication network so as to find the only network leader. When the communication range of agents is unlimited (i.e., 100 units of distance), the structures of the networks are complex. However, since each agent can have all information about its communication network, agents can easily find the network leader without a lot of communications.

Based on the above analysis, it can be seen that although when the communication range of agents is unlimited (i.e., 100 units of distance), the reducing of communication connections in the communication network is very large after employing the information collection mechanism (i.e., it reduces from 190 connections to 19 connections), the communication overhead spent for this reducing is small (i.e., less than 40 broadcast times). Therefore, the information collection mechanism can handle different communication situations and reduce a complex communication network to a simple structure without spending much communication overhead.



### 4.3.2 Experiment 2: test of task allocation

The purpose of this experiment was to evaluate the performance of the proposed approach on task allocation under different communication ranges of agents. The proposed approach (TASC) is compared with three common task allocation approaches, which are: 1) the MILP-based approach (MILP) [79], 2) the myopic approach (MySc) [113], and 3) the max-sum-based approach (FMS) [112]. The first two approaches are centralised task allocation approaches, while the third approach is a decentralised task allocation approach. The performance of TASC and three task allocation approaches are compared under four different communication ranges of agents (i.e.,  $CR$ , see Definition 4.3).

#### Experiment settings

The settings of Experiment 2 are described in Table 4.2.

Parameter	Value
Area size	$50 \times 50$
Number of tasks	100
Deadlines of tasks ( $DLine_{(i,j)}$ )	$20 \sim 200$
Workloads of tasks ( $WLoad_{(i,j)}$ )	$1 \sim 10$
Number of agents	12
Work efficiency of agents ( $Uti_i$ )	1 or 2
Moving speeds of agents ( $Msp_i$ )	1 or 2
Communication range of agents ( $CR$ )	10, 15, 20, 25
Time for one round of coordination ( $TP$ )	25

Table 4.2: The settings of Experiment 2

In Experiment 2, 100 tasks and 12 agents with the same communication ranges are randomly distributed in a  $50 \times 50$  area. The Euclidean distance is also employed to evaluate the distance between the locations [121], [17]. The deadlines of tasks are set from 20 to 200 units of time. The workloads of tasks vary from 1 to 10 units of workload. The work efficiencies of agents are set to 1 or 2 units of workload per unit of time. The moving speeds of agents are set to 1 or 2 units of distance. Experiment 2 are conducted when the communication ranges of all agents are 10, 15, 20 and 25, respectively. The time interval for one round of the group coordination mechanism of TASC is 25 units of time. There are four kinds of capabilities required by tasks and had by agents. Each task requires one or two kinds of capabilities and

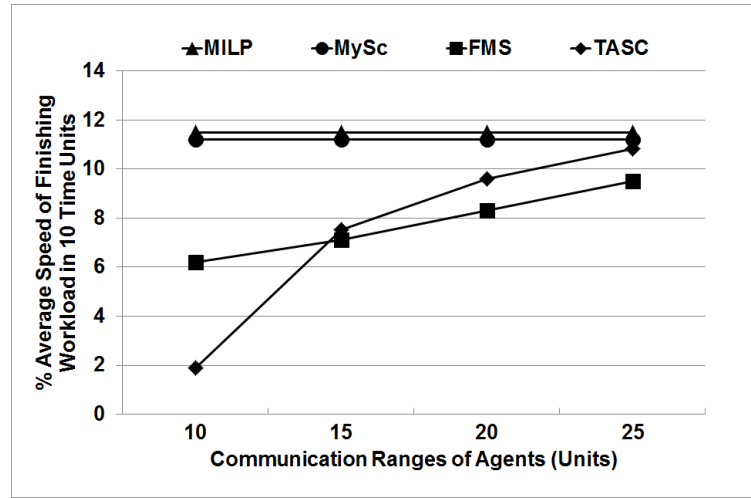
each agent has two random kinds of capabilities.

Since MILP and MySc do not consider communication constraints, we assume that there is no communication constraints when conducting MILP and MySc in the experiment. While communication relationships among agents in FMS and TASC are based on locations (i.e.,  $Loc_i$ , see Definition 4.1) and the communication range of agents (i.e.,  $CR$ , see Definition 4.3). In addition, two different kinds of environments are simulated: static and dynamic. In static environments, 100 tasks are discovered by agents at the beginning of the experiment, while in dynamic environments, only 40 tasks are discovered at the beginning of the experiment, and 60 tasks are discovered during task execution. In addition, since the main purpose of this experiment is to evaluate the performance of allocation solutions created by four approaches, we do not consider the time spent on computation or the amount of information exchanged for task allocation of four approaches. To compensate this weakness, we separately evaluate the time spent for computation and the amount of information exchanged for task allocation in the four approaches in dynamic environments.

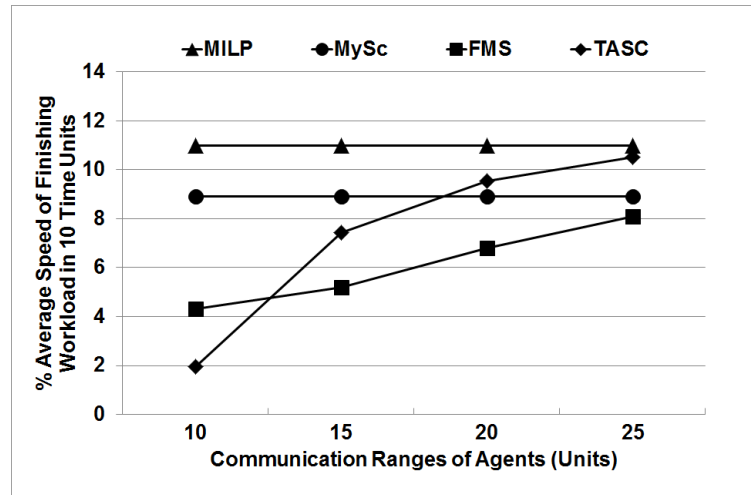
### Experimental results and analysis

The results of Experiment 2 are shown in Figures 4.7, 4.8 and 4.9.

Figure 4.7 shows the performance of the proposed approach on task allocation in static and dynamic environments under different communication ranges. The X-axes in Figures 4.7 (a) and (b) are communication ranges of agents, while the Y-axes in Figures 4.7 (a) and (b) are the average speed of finishing workload in 10 time units.



(a) Static environments



(b) Dynamic environments

Figure 4.7: The results of Experiment 2

From Figure 4.7 (a), it can be seen that when environments are static, MILP and MySc always have the same performance under different communication ranges of agents. That is because MILP and MySc do not consider communication constraints in the experiment. In addition, since MILP can create the optimal solution for task allocation, the solution created by MILP can be taken as the benchmark in the experiment. When the communication range of agents is very limited (i.e., 10 units of distance), FMS and TASC do not perform well. The reason for this in FMS is that most of agents in the environment are isolated and cannot communicate with each other so that agents cannot exchange comprehensive information and they make decisions for task allocation only based on their limited local views of the environment. In TASC, although the group coordination mechanism enables isolated agents in groups to communicate and coordinate with each other at assembly points, there are so many

groups formed in the environment that most agents need to become representative agents and cannot participate in task execution, which greatly reduces the work efficiency of agents in TASC. With the increase of communication ranges, the number of groups in the environment reduces and the work efficiency of the agents in TASC obviously increase. Although agent work efficiency in FMS also increases with the increase in the communication ranges, without coordination among agents, isolated agents cannot exchange information for task allocation so that the increase in the work efficiency of agents in FMS is not as great as that in TASC. When the communication range of agents is limited (i.e., 25 units of distance), the performance of TASC is similar to that of in MILP and MySc.

From Figure 4.7 (b), it can be seen that MILP still has the same performance under different communication ranges of agents in dynamic environments. However, the performance of MySc is worse in dynamic than in static environments. This is because MySc is a myopic scheduling approach, which makes decisions with only short-term task allocation and cannot handle the dynamics of disaster environments. FMS and TASC in dynamic environments have the same performance as that in static environments.

Figure 4.8 shows the time cost for task allocation under different communication ranges. In Figure 4.8, the X-axis is the communication ranges of agents, while the Y-axis is the time spent on computation for task allocation.

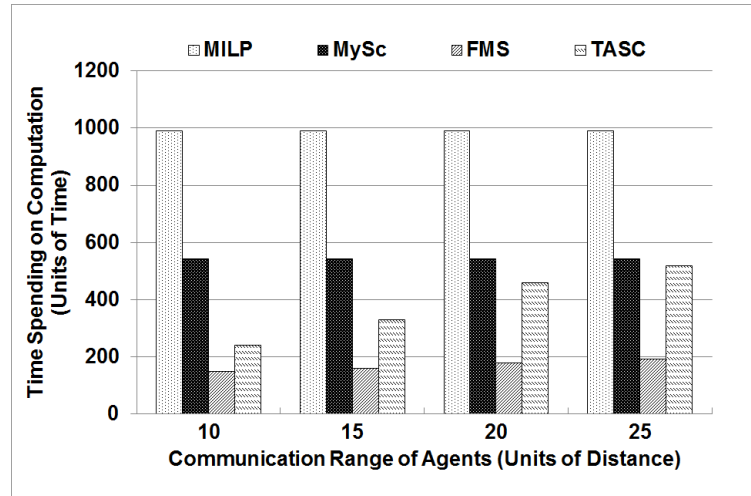


Figure 4.8: The time spent on computation for task allocation

From Figure 4.8, it can be seen that MILP spends the longest time on computation for task allocation. This is because finding the allocation solution in MILP is an NP-hard problem (the proof process can be found in [113]). With an increasing number of tasks and agents, the time spent on computation for task allocation in MILP increases exponentially. MySc spends

the second longest time on computation for task allocation. Although MySc simplifies the computation process of MILP, it is still a centralised task allocation approach, in which the central controller needs to choose the best allocation solution from all combinations of the 12 agents and the 100 tasks. Unlike MILP and MySc, in FMS and TASC, computations for task allocation are distributed to agents and groups, respectively, which can greatly reduce the time spent on computation for task allocation through synchronous computations of agents. FMS is a completely decentralised approach so that it spends the least time on computation for task allocation. In TASC, the computation in each group is still centralised, with the increase in communication ranges, the number of groups decreases, which makes TASC more and more centralised so that the time spent on computation for task allocation increases accordingly. Finally, when the communication range of agents is limited (i.e., 25 units of distance), there are only a few groups in TASC and the time spent on computation for task allocation is similar to that of MILP and MySc.

Figure 4.9 shows the message exchanges among agents during task allocation under different communication ranges. In Figure 4.9, the X-axis is the communication ranges of agents, while the Y-axis is the total amount of information exchanged for task allocation.

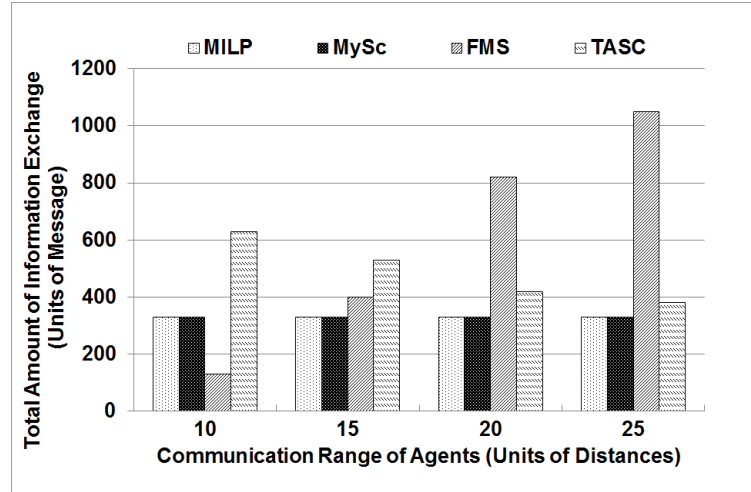


Figure 4.9: The amount of information exchange for task allocation

From Figure 4.9, it can be seen that since information is exchanged only between agents and the central controller in MILP and MySc, MILP and MySc have the same total amount of information exchange for task allocation in the experiment. The information exchange for task allocation in FMS increases exponentially with the increase in the communication ranges of agents. This is because FMS is a completely decentralised task allocation approach and in order for agents to exchange comprehensive information for task allocation,

each agent needs to pass the information to all its direct neighbours. When the communication range of agents is very limited (i.e., 10 units of distance), there are only a few agents having one or two direct neighbours. Therefore, there is nearly no information exchange among agents in FMS. While when the communication range of agents is limited (i.e., 25 units of distance), each agent has at least six direct neighbours, the total amount of information exchange for task allocation in FMS increases exponentially. The information exchange for task allocation in TASC consists of two parts, the information exchange for information collection and the information exchange for group coordination. The information collection mechanism in TASC enables agents to reduce their communication connections and select a network leader for information collection and this greatly reduces the total amount of information exchanged. The major information exchange for task allocation in TASC happens during the group coordination among representative agents. Therefore, when the communication range of agents is very limited (i.e., 10 units of distance), though the information exchange is low (i.e., only a few agents can have one or two direct or indirect neighbours), TASC needs to allocate tasks to so many groups that a large amount of information exchange is needed for coordination. While when the communication range of agents is limited (i.e., 25 units of distance), though the information exchange for information collection increases, the information exchange for group coordination reduces significantly because of the reduced number of groups. This means that the total amount of information exchange for task allocation in TASC is reduced.

### 4.3.3 Findings from experiments

From Experiments 1, and 2, we have the following discoveries.

1. By employing the information collection mechanism, agents with limited communication capabilities can efficiently share information for task allocation in disaster environments under space and communication constraints (to achieve **Objective 2**, see Section 1.3);
2. By employing the group task allocation mechanism, coordinators can create suitable task allocation solutions for heterogeneous agents, which can be as good as solutions created by the centralised task allocation approaches under space and communication constraints (to achieve **Objectives 3 and 5**, see Section 1.3);
3. By employing the group coordination mechanism, groups of agents can periodically re-allocate group members to fit dynamic changes of disaster environments under space and communication constraints (to achieve **Objective 3**, see Section 1.3).

## 4.4 Summary

In this chapter, a dynamic task allocation approach was proposed for heterogeneous agents in disaster environments under time, space and communication constraints. First, the problem description and definitions of the dynamic task allocation approach in disaster environments were given. Then, the process and the three mechanisms of the dynamic task allocation approach were introduced in detail. Finally, experiments to evaluate the performance of the dynamic task allocation approach were demonstrated and analysed. Experimental results showed that the proposed approach could have better performance than that of in many existing approaches in terms of information collection and dynamic task allocation in disaster environments under time, space and communication constraints.

# Chapter 5

---

## A Wireless Mobile Robots Search and Deployment Approach

Due to the complexity of disaster environments as well as capability limitations of wireless mobile robots, ad hoc network establishment in disaster environments is an important step for task allocation. In this chapter, a wireless mobile robot deployment approach is proposed. The proposed approach consists of a search process and a deployment process. In particular,

- the search process enables WRs to efficiently searching in disaster environments so as to collect information for task allocation by considering unknown and complexity of environments and limited energy and capabilities of agents (to solve **Challenging Issue 1**, identified in Section 1.2); and
- the deployment process can timely find deployment locations for WRs in disaster environments so as to establish ad hoc networks (to solve **Challenging Issues 6 and 7**, identified in Section 1.2) by considering communication constraints of environments and limited capabilities of WRs (to solve **Challenging Issue 2**, identified in Section 1.2).

### 5.1 Definitions

Let  $D$  be a two-dimension disaster environment, which is divided into many equivalent areas and the size of each area is  $S$ . In this chapter, each area is indexed and represented by the location at its center  $Loc_m \in D$ .  $Loc_m$  can be occupied by either an obstacle or a task  $T_i$ . Based on the occupation, all locations of  $D$  can be classified into three types, which are: 1) free locations  $FSet$ , 2) obstacle locations  $OSet$  and 3) task locations  $TSet$ . After entering the environment  $D$ , WRs have to collect information in the environment so as to enlarge their local views about the environment. The definition of a WR and the information collected by the WR (i.e., the local view of the WR) are defined as follows.



**Definition 5.1.** A *Wireless Mobile Robot*  $WR_j$  can be defined as a four-tuple  $WR_j = \langle Loc_j, Eng_j, Sta_j, ANet_k \rangle$ , where  $Loc_j$  is the current location of  $WR_j$ ,  $Eng_j$  is the remaining energy of  $WR_j$ , which is represented by the number of locations that  $WR_j$  can move to,  $Sta_j$  is the status of  $WR_j$ , which can be either ‘searching’ or ‘deployed’ and  $ANet_k$  is the information of the ad hoc network, to which  $WR_j$  belongs. If  $Sta_j$  is ‘searching’,  $ANet_k$  must be ‘ $\emptyset$ ’.

**Definition 5.2.** The *information* ( $Inf_j$ ) collected by the  $WR_j$  (i.e., the local view of  $WR_j$ ) about a disaster environment (i.e.,  $D$ ) can be defined as a two-tuple  $Inf_j = \langle WRSet_j, LSet_j \rangle$ , where  $WRSet_j$  is the set of WRs in  $D$ , which have ever communicated and shared information with  $WR_j$  and  $LSet_j$  is the locations in  $D$ , whose information is collected by  $WR_j$ . Based on the three types of locations,  $LSet_j$  can be further defined as a three-tuple  $LSet_j = \langle FSet_j, OSet_j, TSet_j \rangle$ , where  $FSet_j$ ,  $OSet_j$  and  $TSet_j$  are information of free locations, obstacle locations, and task locations collected by  $WR_j$ , respectively.

An ad hoc network (i.e.,  $ANet_k$ ) is established by several WRs so that the information of  $ANet_k$  is the joint information of all WRs to establish the network, which is defined as follows.

**Definition 5.3.** The *information of an ad hoc network* ( $ANet_k$ ) can be defined as a three-tuple  $ANet_k = \langle WRSet_k, LSet_k^C, LSet_k^U \rangle$ , where  $WRSet_k$  is the information of WRs, establishing  $ANet_k$ ,  $LSet_k^C$  is the locations, covered by  $ANet_k$ , and  $LSet_k^U$  is the locations, whose information is collected, but uncovered by  $ANet_k$ . Based on the three types of locations,  $LSet_k^C$  and  $LSet_k^U$  can be further defined as  $LSet_k^C = \langle FSet_k^C, OSet_k^C, TSet_k^C \rangle$  and  $LSet_k^U = \langle FSet_k^U, OSet_k^U, TSet_k^U \rangle$ , respectively.

In order to avoid obstacles, the A\* search algorithm [109] [135] is employed by WRs to create paths between two locations. The reason for this choice will be explained in Subsection 5.3.2. A path created by the A\* search algorithm is defined as follows.

**Definition 5.4.** A *path* of  $WR_j$  ( $Path(Loc_j, Loc_n)$ ) is from its current location (i.e.,  $Loc_j$ , see Definition 5.1) to the next location (i.e.,  $Loc_n$ ), which can be defined as a sequence of locations  $Path(Loc_j, Loc_n) = (Loc_j, Loc_{m_1}, \dots, Loc_{m_p}, Loc_n)$ , where  $Loc_{m_p}$  is the  $p^{th}$  location that  $WR_j$  moves to before arriving at  $Loc_n$ .

Since WRs rely on wireless technologies, the sensing and communication distances of WRs are limited. In this chapter, we assume that the sensing distances of all WRs are the same and represented by  $r$  and the communication distances of all WRs are the same and

represented by  $R$ . Based on the sensing and communication ranges of WRs, the three objectives of Task-Based Wireless mobile Robots Search and Deployment (TBWSD) for the ad hoc network establishment can be described as follows.

- 1) **The communication of WRs:** It means that two communicable WRs in the network are within the communication range of each other;
- 2) **The maximum coverage of tasks:** It means that the maximum tasks are within the sensing ranges of WRs in the network; and
- 3) **The maximum coverage of an environment:** It means that the maximum locations in the environment are within the communication ranges of WRs in the network.

In order to evaluate whether an ad hoc network achieves the above three objectives of TBWSD, the definition of the guidance locations of a task (e.g.  $T_i$ ) is proposed as follows.

**Definition 5.5.** The *guidance locations* ( $GLoc_i$ ) of a task (e.g.  $T_i$ ) are defined as the locations, at which first responders can be guided to preform  $T_i$ .

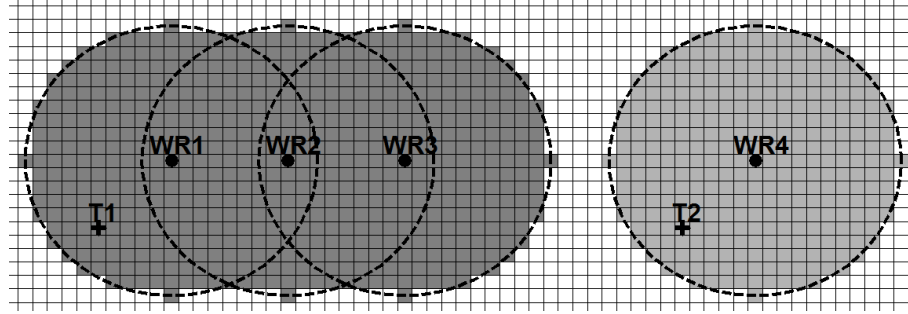


Figure 5.1: The guidance locations of tasks

An example of the guidance locations of tasks is illustrated in Figure 5.1. In the figure, the grids are locations in a disaster environment, the points represent deployment locations of WRs, the circle areas inside dash lines represent the communication range of WRs and black crosses represent locations of tasks. In Figure 5.1, four WRs (i.e.,  $WR_1$ ,  $WR_2$ ,  $WR_3$  and  $WR_4$ ) are deployed, where  $WR_1$  can directly communicate with  $WR_2$ ;  $WR_2$  can directly communicate with  $WR_3$ ;  $WR_1$  can indirectly communicate with  $WR_3$  through  $WR_2$ ; and  $WR_4$  cannot communicate with any other WRs in the environment. Two stationary tasks (i.e.,  $T_1$  and  $T_2$ ) are within the sensing ranges of  $WR_1$  and  $WR_4$ , respectively. Because  $WR_1$ ,  $WR_2$  and  $WR_3$  are communicable, the information of  $T_1$  can be shared among  $WR_1$ ,  $WR_2$  and  $WR_3$ . Hence, when suitable first responders move to a location within the communication range of any of  $WR_1$ ,  $WR_2$  or  $WR_3$ , these first responders can be guided to perform  $T_1$ . Therefore, the guidance locations of  $T_1$  are locations within the communication

ranges of  $WR_1$ ,  $WR_2$  and  $WR_3$  (i.e., locations in deep gray in Figure 5.1). Because  $WR_4$  cannot communicate with any other WRs in the environment, only when suitable first responders move to a location within the communication range of  $WR_4$ , these first responders can be guided to perform  $T_2$ . Therefore, the guidance locations of  $T_2$  are locations within the communication range of  $WR_4$  (i.e., locations in light gray in Figure 5.1).

Based on the guidance locations of tasks, the objective value function  $Objval_k$  is proposed to evaluate whether an ad hoc network (e.g.  $ANet_k$ ) can achieve the three objectives of TBWSD, which is calculated as follows.

$$Objval_k = \frac{S \cdot \sum_{T_i \in TSet_k^C} |GLoc_i|}{(|TSet_k^C| + |TSet_k^U|)|WRSet_k|\pi R^2}, \quad (5.1)$$

where  $S$  is the size of the area of each location,  $|GLoc_i|$  is the guidance locations of  $T_i$  covered by  $ANet_k$  (i.e.,  $T_i \in TSet_k^C$ , see Definition 5.3),  $|TSet_k^C|$  is the number of tasks covered by  $ANet_k$  (see Definition 5.3),  $|TSet_k^U|$  is the number of tasks uncovered by  $ANet_k$  (see Definition 5.3),  $|WRSet_k|$  is the number of WRs establishing  $ANet_k$  and  $\pi R^2$  is the communication range of a WR. The value of  $Objval_k$  is in-between  $[0, 1]$ , where 0 and 1 indicates that  $ANet_k$  either does not or does achieve the three objectives of TBWSD, respectively.

## 5.2 The Principle of the Proposed Approach

The principle of the WR search and deployment approach is illustrated in Figure 5.2.

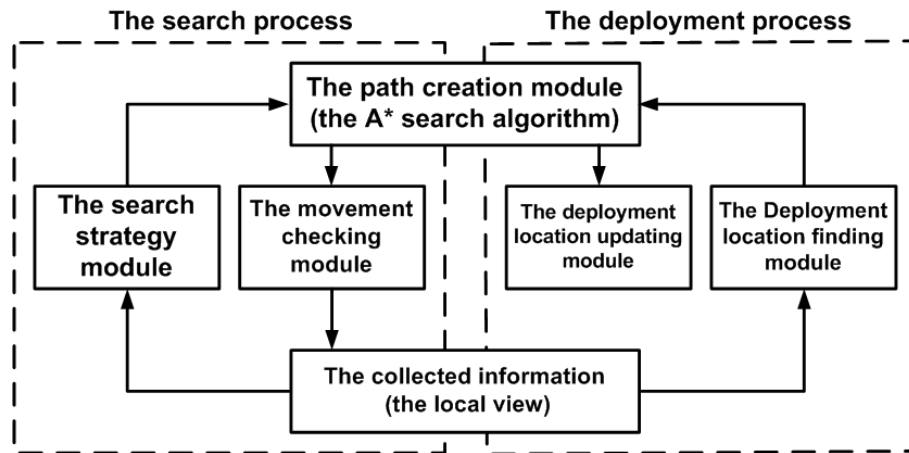


Figure 5.2: The principle of the proposed approach

From Figure 5.2, it can be seen that there are two concurrent processes: the search process and the deployment process. The search process contains four modules, which are the search

strategy module, the path creation module, the movement checking module and the collected information module. The deployment process also contains four modules, which are the collected information module, the deployment location finding module, the path creation module and the deployment location updating module. The two processes in a WR (e.g.  $WR_j$ ) are described in Algorithm 5.1.

---

**Algorithm 5.1:** The two current processes in  $WR_j$

---

```

1  repeat
2    The search strategy module creates the next location  $Loc_n$ .
3    The path creation module creates the path to the next location  $Path(Loc_j, Loc_n)$ .
4    The movement checking module calculates the energy of  $WR_j$ .
5    if The movement checking module returns 1 then
6       $WR_j$  moves to  $Loc_n$  and collect information during movement.
7    end
8    if The deployment location finding module finds a new deployment location
        $Loc_{new}$  then
9      The path creation module creates path to the new deployment location
        $Path(Loc_j, Loc_{new})$ .
10     The deployment location updating module calculates the energy of  $WR_j$ .
11     if The deployment location updating module returns 1 then
12        $Loc_{new}$  replaces  $Loc_{deploy}$ 
13     else
14        $WR_j$  still uses the current deployment location  $Loc_{deploy}$ .
15     end
16   end
17 until The movement checking module returns 0;
18  $WR_j$  moves to the current deployment location.

```

---

Algorithm 5.1 is explained as follows. The search process is a looping process, one round of which includes four steps.

**Step 1:** The search strategy module creates the next location (i.e.,  $Loc_n$ ) for  $WR_j$  (Line 2);

**Step 2:** The path creation module employs the A\* search algorithm to creates the path (i.e.,  $Path(Loc_j, Loc_n)$ , see Definition 5.4) from  $WR_j$ 's current location (i.e.,  $Loc_j$ , see Definition 5.1) to  $Loc_n$ ; The map used by the A\* search algorithm is the information stored in the collected information module (i.e., the local view) of  $WR_j$  (i.e.,  $Inf_j$ ,

see Definition 5.2) (Line 3);

**Step 3:** Before moving to  $Loc_n$ , the movement checking module of  $WR_j$  calculates that whether  $WR_j$  has enough energy to move to the deployment location (i.e.,  $Loc_{deploy}$ ) after moved to  $Loc_n$  (Line 4); and

**Step 4:** If the movement checking module returns value 1 (i.e.,  $WR_j$  has enough energy),  $WR_j$  moves to  $Loc_n$  through  $Path(Loc_j, Loc_n)$ , during which  $WR_j$  collects information of locations within its sensing range and stores the information in the collected information module (Line 5).

The above four steps are repeated until the movement checking module returns value 0 (i.e.,  $WR_j$  does not have enough energy) (Line 17). In the such circumstance,  $WR_j$  dose not move to  $Loc_n$  and begins to move to  $Loc_{deploy}$  (Line 18).

During the search process, the deployment process of  $WR_j$  continuously calculates the most suitable deployment locations based on the collected information (i.e., information in the collected information module). If the deployment location finding module finds a new deployment location (i.e.,  $Loc_{new}$ ) that is more suitable than the current deployment location (i.e.,  $Loc_{deploy}$ ), the path creation module of  $WR_j$  employs the A\* search algorithm to create the path (i.e.,  $Path(Loc_j, Loc_{new})$ ) from its current location (i.e.,  $Loc_j$ , see Definition 5.1) to  $Loc_{new}$  (Lines 8 to 9). Then, the deployment location updating module of  $WR_j$  calculates that whether  $WR_j$  has enough energy to move from  $Loc_j$  to  $Loc_{new}$  (Line 10). If the deployment location updating module returns value 1 (i.e.,  $WR_j$  has enough energy),  $Loc_{new}$  replaces  $Loc_{deploy}$  (Lines 11 to 12), otherwise (i.e.,  $WR_j$  does not have enough energy),  $Loc_{deploy}$  does not change (Lines 13 to 14).

The search process and the deployment process of the proposed approach will be introduced in detail in Section 5.3 and 5.4, respectively.

### 5.3 Technical Design of the Search Process

In the search process, a WR (e.g.  $WR_j$ ) trends to collect as much information of the environment as possible with limited capabilities so as to enable the deployment process to create the most suitable deployment location. In this section, the search strategy module, the A\* search algorithm and the movement checking module are introduced in detail.

### 5.3.1 The search strategy module

Due to the energy limitation, it is difficult for  $WR_j$  to search the entire environment. Hence, how to collect as much information of tasks as possible within the energy limitation is the primary concern of  $WR_j$  during search. To this end, a method adapted from the mean shift algorithm [151] [137] is employed by the search strategy module to continuously create next locations, which enables  $WR_j$  to always move to the locations that can cover the maximum tasks in an area. The mean shift-based search method employed by the module is described by Algorithm 5.2.

---

**Algorithm 5.2:** The mean shift-based search method

---

```

1 repeat
2    $Temp = \emptyset$ 
3   for each  $T_i \in TSet_j$  do
4     if  $Dist(Loc_j, Loc_i) \leq r$  then
5        $Temp = Temp \cup T_i$ 
6     end
7   end
8    $Loc_n = \frac{\sum_{T_i \in Temp} Loc_i}{|Temp|}$ 
9    $dis = Dist(Loc_j, Loc_n)$ 
10  until  $dis < sqrt(S)$ ;
11  records  $Loc_j$  as  $Loc_{local}$ 
```

---

Algorithm 5.2 is explained as follows. At the beginning, the temporal variable  $Temp$  is initialised as  $\emptyset$  (Line 2). After variable initialisation, the tasks within the sensing range of  $WR_j$  are recorded in  $Temp$  (Lines 3 to 5). Then, the mean value (i.e., the average value) of locations of all tasks in  $Temp$  is calculated as the next location (i.e.,  $Loc_n$ ) (Line 8). After that, the difference between the current location (i.e.,  $Loc_j$ ) and  $Loc_n$  is calculated and recorded in variable  $dis$  (Line 9). When  $dis$  is less than the width of one location (i.e.,  $sqrt(S)$ ,  $S$  is the size of one location), which means that  $WR_j$  has moved to the location that can cover the maximum tasks in an area (i.e., a local maximum task coverage location), which is recorded as  $Loc_{local}$  (Lines 10 to 11).

$Loc_{local}$  is just a local maximum task coverage location. The searching goal of  $WR_j$  is to find the global maximum task coverage location in the environment. Thus, after arriving at  $Loc_{local}$ ,  $WR_j$  should continue to search if it still has energy. However, at  $Loc_{local}$ ,  $WR_j$  cannot employ the mean shift-based search method described in Algorithm 5.2. In such

circumstances,  $WR_j$  should continuously move far away from  $Loc_{local}$  until the mean of tasks within the sensing range of  $WR_j$  does not point to the direction of  $Loc_{local}$ . The mean shift-based search method and the moving away method alternatively create next locations for  $WR_j$  during the search process.

### 5.3.2 The path creation module

The A\* search algorithm [107] is a common and popular path planning algorithm in disaster environments, which employs the grid map and heuristics to find the shortest pathes for robots by avoiding obstacles. The prerequisite of employing the A\* search algorithm to create the path between the current location of  $WR_j$  (i.e.,  $Loc_j$ ) and the next location (i.e.,  $Loc_n$ ) is that  $WR_j$  has the map around these two locations. From the search strategy module (i.e., see Section 5.3.1), it can be seen that the local view of  $WR_j$  contains the locations that were or are within the sensing range of  $WR_j$  (i.e.,  $Loc_m \in Inf_j$ , see Definition 5.2). In addition,  $Loc_n$  is calculated from all task locations within the sensing range of  $WR_j$  at  $Loc_j$  so that  $Loc_n$  must be within the sensing range of  $WR_j$ . Therefore, the information of locations between  $Loc_j$  and  $Loc_n$  can be known by  $WR_j$ . In such circumstances, the local view of  $WR_j$  (i.e.,  $Inf_j$ , see Definition 5.2) is the map used by the A\* search algorithm, which can quickly create the path between  $Loc_j$  and  $Loc_n$  by avoiding obstacles.

### 5.3.3 The movement checking module

Since a WR (e.g.  $WR_j$ ) only has limited energy, during the search process, the movement checking module of  $WR_j$  checks its remaining energy (i.e.,  $Eng_j$ , see Definition 5.1) and decides when  $WR_j$  should stop the search process and begin to move to the deployment location (i.e.,  $Loc_{deploy}$ ).  $Eng_j$  can be divided into two parts: **1)** the energy for the search process and **2)** the energy for moving to  $Loc_{deploy}$  (i.e.,  $Eng_{deploy}$ ). In particular, for each path (i.e.,  $Path(Loc_j, Loc_n)$ ) created by the A\* search algorithm, the movement checking module checks  $Eng_j$  and gives a return value (i.e.,  $Ret$ ) to  $WR_j$ , which is calculated as Equation 5.2.

$$Ret = \begin{cases} 1, & 2|Path(Loc_j, Loc_n)| + Eng_{deploy} \leq Eng_j \\ 0, & 2|Path(Loc_j, Loc_n)| + Eng_{deploy} > Eng_j \end{cases} \quad (5.2)$$

where  $|Path(Loc_j, Loc_n)|$  is the required energy for moving through  $Path(Loc_j, Loc_n)$ ,  $Eng_{deploy}$  is the energy for moving to  $Loc_{deploy}$ , created by the deployment location updating module (see Subsection 5.4.2). If after moved to  $Loc_n$ ,  $WR_j$  still has enough energy (i.e.,

$Eng_j$ ) to move to  $Loc_{deploy}$ , the movement checking module returns value 1, otherwise, the movement checking module returns value 0 and  $WR_j$  stops the search process and begins to move to  $Loc_{deploy}$ .

After moved to  $Loc_n$ ,  $WR_j$  loses some energy for this movement (i.e.,  $Eng_j = Eng_j - |Path(Loc_j, Loc_n)|$ , see Definition 5.1 and 5.4) and  $Eng_{deploy}$  is updated as  $Eng_{deploy} = Eng_{deploy} + |Path(Loc_j, Loc_n)|$ .

## 5.4 Technical Design of the Deployment Process

During the search process, the deployment process continuously finds the most suitable deployment location based on the collected information (i.e.,  $Inf_j$ , see Definition 5.2). In this section, the deployment location finding module and the deployment location updating module are introduced in detail.

### 5.4.1 The deployment location finding module

In the proposed approach, the deployment location finding module finds deployment locations for WRs based on two different mechanisms, which are the deployment location finding by two points (DLF-TP) and the deployment location finding by density graph (DLF-DG). In this subsection, the two kinds of deployment location finding mechanisms are introduced at first. The computational complexities of two mechanisms are compared and analysed, respectively.

#### Mechanism 1: the deployment location finding by two points (DLF-TP)

The theoretical background and the procedure of the deployment location finding by two points (DLF-TP) are introduced in detail.

##### I) The theoretical background of DLF-TP

During the search process, according to whether  $WR_j$  has collected information about established ad hoc networks in the environment, DLF-TP can be divided into two situations: **Situation 1:** DLF-TP without information of ad hoc networks, and **Situation 2:** DLF-TP with information of ad hoc networks.

##### **Situation 1:** DLF-TP without information of ad hoc networks

In this situation, since the communication range (i.e., the coverage area) of  $WR_j$  is fixed and  $WR_j$  does not find other deployed WRs to communicate with, the deployment location of  $WR_j$  only needs to cover the maximum number of tasks in the environment. Because



the sensing range of  $WR_j$  is a fixed size circle, the problem of deploying  $WR_j$  to cover the maximum number of tasks is the same as the problem of using a circle to cover scattered points. In papers [56], [62], researchers approved that if a circle could cover the maximum number of scattered points, there must be at least two points on the border of the circle. Thus, the deployment location of  $WR_j$  can be found based on locations of two tasks, which is described as follows.

**Proposition 5.1.** *Given two stationary tasks (i.e.,  $T_1$  and  $T_2$ ) and their locations (i.e.,  $Loc_1$  and  $Loc_2$ ) in an environment and the sensing distance of  $WR_j$  (i.e.,  $r$ ), if the distance between  $Loc_1$  and  $Loc_2$  is less than or equals  $2r$ , deployment locations (i.e.,  $Loc_c$ ) can be found. If  $WR_j$  is deployed at  $Loc_c$ ,  $T_1$  and  $T_2$  are on the border of the sensing range of  $WR_j$ .*

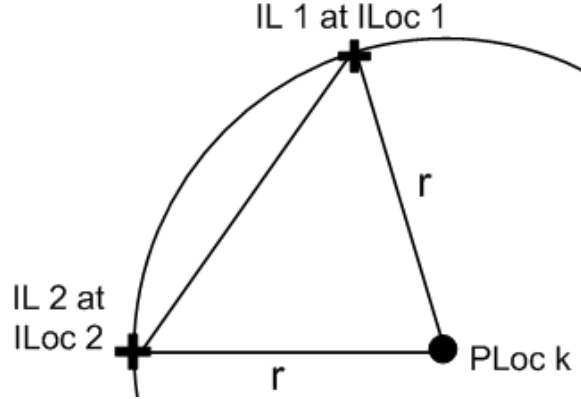


Figure 5.3: An example of DLF-TP based on Proposition 5.1

*Proof.* As shown in Figure 5.3, the two crosses are locations of two tasks (i.e.,  $T_1$  at  $Loc_1$  and  $T_2$  at  $Loc_2$ ); the circle area inside the dash line is the communication range of  $WR_j$  (i.e., the communication distance is  $R$ ); the circle area inside the solid line is the sensing range of  $WR_j$  (i.e., the sensing distance is  $r$ ); and the point is the deployment location for  $WR_j$  (i.e.,  $Loc_c$ ), which is found based on  $Loc_1$  and  $Loc_2$ . Based on the definition of the circle, the relationships between coordinates of  $Loc_1 = (x_1, y_1)$ ,  $Loc_2 = (x_2, y_2)$  and  $Loc_c = (x_c, y_c)$  are described as Equation 5.3.

$$\begin{cases} (x_1 - x_c)^2 + (y_1 - y_c)^2 = r^2 \\ (x_2 - x_c)^2 + (y_2 - y_c)^2 = r^2 \end{cases} \quad (5.3)$$

When  $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \leq 2r$ ,  $x_c$  and  $y_c$  in Equation 5.3 have two solutions for each combination of two tasks.

**Situation 2:** DLF-TP with information of ad hoc networks

In this situation,  $WR_j$  should join the ad hoc network with the maximum number of WRs (i.e.,  $ANet_k$ ). After  $WR_j$  joined the network,  $ANet_k$  should still achieve the three objectives of TBWSD. In order for  $ANet_k$  to achieve the communication of WRs and the maximum coverage of the environment, the distance between the deployment location of  $WR_j$  and its communicable WR in  $ANet_k$  should be  $R$ , which means that  $WR_j$  is on the border of the communication range of its communicable WR in  $ANet_k$ . In order for  $WR_j$  to achieve the maximum coverage of tasks, based on Proposition 5.1, there must be at least a task on the border of the sensing range of  $WR_j$ . Under this consideration, the deployment location of  $WR_j$  can be found based on the deployment location of a WR in  $ANet_k$  (i.e.,  $WR_u \in WRSet_k$ , see Definition 5.3) and the location of a task uncovered by  $ANet_k$  (i.e.,  $T_i \in TSet_k^U$ , see Definition 5.3), which is described as follows.

**Proposition 5.2.** *Given one WR in  $ANet_k$  (i.e.,  $WR_1$ ), one task uncovered by  $ANet_k$  (i.e.,  $T_2$ ) and their locations (i.e.,  $Loc_1$  and  $Loc_2$ ) in the environment and the communication and sensing distances of  $WR_j$  (i.e.,  $R$  and  $r$ , respectively), if the distance between  $Loc_1$  and  $Loc_2$  is more than or equals  $R - r$  and is less than or equals  $R + r$ , deployment locations (i.e.,  $Loc_c$ ) can be found. If  $WR_j$  is deployed at  $Loc_c$ ,  $WR_1$  and  $T_2$  are on the borders of the communication and sensing ranges of  $WR_j$ , respectively.*

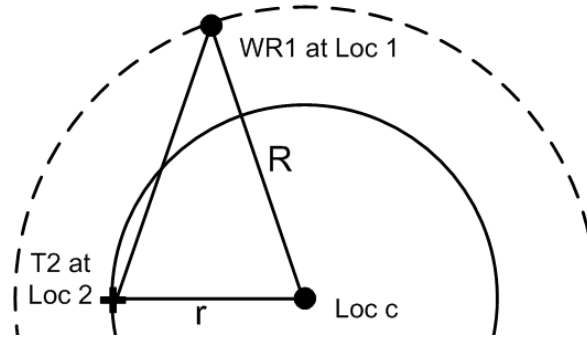


Figure 5.4: An example of DLF-TP based on Proposition 5.2

*Proof.* As shown in Figure 5.4, the point on the border of the circle with dash line is the location of a WR in  $ANet_k$  (i.e.,  $WR_1$  at  $Loc_1$ ); the cross is the location of a task uncovered by  $ANet_k$  (i.e.,  $T_2$  at  $Loc_2$ ); the circle area inside the dash line is the communication range of  $WR_j$  (i.e., the communication distance is  $R$ ); The circle area inside the solid line is the sensing range of  $WR_j$  (i.e., the sensing distance is  $r$ ); and the point inside the circle area is the deployment location of  $WR_j$  (i.e.,  $Loc_c$ ), which can be found based on  $Loc_1$

and  $Loc_2$ . Based on the definition of the circle, the relationships between coordinates of  $Loc_1 = (x_1, y_1)$ ,  $Loc_2 = (x_2, y_2)$  and  $Loc_c = (x_c, y_c)$  are described by Equation 5.4.

$$\begin{cases} (x_1 - x_c)^2 + (y_1 - y_c)^2 = R^2 \\ (x_2 - x_c)^2 + (y_2 - y_c)^2 = r^2 \end{cases} \quad (5.4)$$

When  $(R - r) \leq \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \leq (R + r)$ ,  $x_c$  and  $y_c$  in Equation 5.4 have two solutions for each combination of a WR in an ad hoc network and a task uncovered by the network.

Proposition 5.1 is described as  $Loc_c = Ctt(Loc_i, Loc_{i'}, r)$  in the proposed approach, where  $Loc_c$  is the deployment location of  $WR_j$ ,  $Loc_i$  and  $Loc_{i'}$  are locations of two tasks in the environment and  $r$  is the sensing range of  $WR_j$ . Proposition 5.2 is described as  $Loc_c = Cta(Loc_u, Loc_i, R, r)$  in the proposed approach, where  $Loc_c$  is the deployment location of  $WR_j$ ,  $Loc_u$  and  $Loc_i$  are locations of a WR in an ad hoc network and a task uncovered by the network, respectively,  $R$  and  $r$  are the communication and sensing distances of  $WR_j$ .

## II) The procedure of DLF-TP

Based on Propositions 5.1 and 5.2, the procedures of DLF-TP of a WR (e.g.  $WR_j$ ) in two situations, which are: **Situation 1:** DLF-TP without information of ad hoc networks, and **Situation 2:** DLF-TP with information of ad hoc networks, are described by Algorithms 5.3 and 5.4, respectively.

**Situation 1:** The procedure of DLF-TP without information of ad hoc networks is described by Algorithm 5.3

**Algorithm 5.3:** The procedure of DLF-TP without information of ad hoc networks

---

```

1  $Temp = \emptyset$ 
2 for each  $T_i \in TSet_j$  do
3   for each  $T_{i'} \in TSet_j$  do
4     if  $i \neq i' \ \&\& \ Dist(Loc_i, Loc_{i'}) \leq 2r$  then
5        $Loc_c = Ctt(Loc_i, Loc_{i'}, r)$ 
6        $Temp = Temp \cup Loc_c$ 
7     end
8   end
9 end
10 for each  $Loc_c \in Temp$  do
11   if  $Loc_c \notin OSet_j$  then
12     for each  $T_i \in TSet_j$  do
13       if  $Dist(Loc_i, Loc_c) \leq r$  then
14          $TSet_c = TSet_c \cup T_i$ 
15       end
16     end
17   end
18 end
19 choose the  $Loc_c$  with  $Max(|TSet_c|)$  as  $Loc_{deploy}$ 

```

---

In Algorithm 5.3, at the beginning, the temporary variable  $Temp$  is initialised to  $\emptyset$  (Line 1). After that, all potential locations found based on locations of two tasks (i.e.,  $T_i, T_{i'} \in TSet_j$ , see Definition 5.2) are calculated based on Proposition 5.1 and recorded in  $Temp$  (Lines 2 to 6). If a potential location (i.e.,  $Loc_c$ ) is not occupied by the obstacle, the tasks which will be covered by  $WR_j$  at  $Loc_c$  are calculated and recorded in  $TSet_c$  (Lines 10 to 14). Finally, the potential location at which  $WR_j$  can cover the maximum number of tasks is chosen to be the deployment location of  $WR_j$  (Line 19).

**Situation 2:** The procedure of DLF-TP with information of ad hoc networks is described by Algorithm 5.4

**Algorithm 5.4:** The procedure of DLF-TP with information of ad hoc networks

---

```

1  $Temp = \emptyset$ 
2 for each  $T_i \in TSet_k^U$  do
3   for each  $WR_u \in WRSet_k$  do
4     if  $(R - r) \leq Dist(Loc_i, Loc_u) \leq (R + r)$  then
5        $Loc_c = Cta(Loc_i, Loc_u, R, r)$ 
6        $Temp = Temp \cup Loc_c$ 
7     end
8   end
9 end
10 for each  $Loc_c \in Temp$  do
11   if  $Loc_c \notin OSet_j$  then
12     for each  $T_i \in TSet_k^U$  do
13       if  $Dist(Loc_i, Loc_c) \leq R$  then
14          $TSet_c = TSet_c \cup T_i$ 
15       end
16     end
17   end
18 end
19 choose the  $Loc_c$  with  $Max(|TSet_c|)$  as  $Loc_{deploy}$ 

```

---

In Algorithm 5.4, at the beginning, the temporary variable  $Temp$  is initialised to  $\emptyset$  (Line 1). After that, all potential locations found based on one location of a WR in  $ANet_k$  (i.e.,  $WR_u \in WRSet_k$ , see Definition 5.3) and one location of a task uncovered by  $ANet_k$  (i.e.,  $T_i \in TSet_k^U$ , see Definition 5.3) are calculated based on Proposition 5.2 and recorded in  $Temp$  (Lines 2 to 6). If a potential location (i.e.,  $Loc_c$ ) is not occupied by the obstacle, the tasks that are uncovered by  $ANet_k$ , but will be covered by  $WR_j$  at  $Loc_c$ , are calculated and recorded in  $TSet_c$  (Lines 10 to 14). Finally, the potential location at which  $WR_j$  can cover the maximum number of tasks that are uncovered by  $ANet_k$  is chosen to be the deployment location of  $WR_j$  (Line 19).

**Mechanism 2: the deployment location finding by density graph**

In this mechanism, the theoretical background and the procedures of the deployment location finding module by density graph (DLF-DG) are introduced in detail.

**I) The theoretical background of DLF-DG**

If a WR (e.g.  $WR_j$ ) with a limited sensing distance can cover a task ( $T_i$ ), the distance between locations of  $WR_j$  and  $T_i$  must be less than or equal to  $r$ . Since the distance between  $WR_j$  and  $T_i$  is mutual,  $WR_j$  can continuously update a density graph, in which each location (i.e.,  $Loc_m$ ) associates with a density value to represent how many tasks can be covered if  $WR_j$  is deployed at  $Loc_m$ . In particular, for each task (i.e.,  $T_i \in TSet_j$ , see Definition 5.2), a circle can be drawn with the center at  $Loc_i$  and the radius equals to  $r$ . The locations within the circle is called the coverage locations of  $T_i$  and the density values of these locations increase one for  $T_i$ . An example of the coverage locations of  $T_i$  is shown in Figure 5.5.

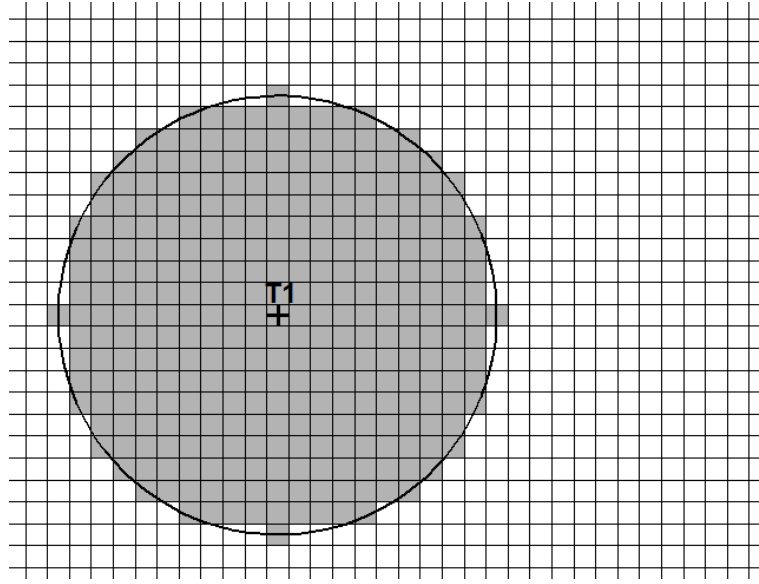


Figure 5.5: The coverage locations of a task

In Figure 5.5, the grids are locations in a disaster environment, the cross is the location of  $T_1$ , the circle area within the solid line is the sensing range of  $WR_j$  and the grey locations are the coverage locations of  $T_1$ . When  $WR_j$  updates the density graph for  $T_i$ , the density value of each location in grey increases one.

## II) The procedure of DLF-DG

Based on the coverage locations of tasks,  $WR_j$  can draw a density graph through updating the density values of locations (i.e.,  $Loc_m$ ). The update of density values of tasks includes two parts, which are: **i)** the density value that is increased by tasks discovered by  $WR_j$  and **ii)** the density values that is decreased by tasks, which have already covered by deployed WRs. The update of density values is described in Algorithm 5.5.

**Algorithm 5.5:** The update of density values

---

```

1 for each  $Loc_m$  do
2    $Dv_m = 0$ 
3   for each  $ANet_k$  do
4     for each  $T_i \in TSet_j$  do
5       if  $Dist(Loc_m, Loc_i) \leq r$  then
6          $Dv_m = Dv_m + 1$ 
7       end
8     end
9     for each  $T_i \in TSet_k^C$  do
10      if  $Dist(Loc_m, Loc_i) \leq r$  then
11         $Dv_m = Dv_m - 1$ 
12      end
13    end
14  end
15 end

```

---

At the beginning of Algorithm 5.5, the density value of each location is initialised to 0 (Line 2). Then, for each task discovered by  $WR_j$  (i.e.,  $T_i \in TSet_j$ ), the density values of corresponding locations increase one (Lines 3 to 6). Finally, for each task that has already covered by WRs in an ad hoc network, the density values of corresponding locations decrease one (Lines 9 to 11).

There can be two situations for DLF-DG, i.e., **Situation 1:** DLF-DG without information of ad hoc networks, and **Situation 2:** DLF-DG with information of ad hoc networks.

**Situation 1:** The procedure of DLF-DG without information of ad hoc networks

If  $WR_j$  has collected information of any existing ad hoc networks in the environment,  $WR_j$  will establish an ad hoc network by itself. Since the communication range (i.e., the coverage area) of  $WR_j$  is fixed and  $WR_j$  could not find other deployed WRs to communicate with, the deployment location of  $WR_j$  should be not occupied by the obstacle with the highest density value in the environment.

**Situation 2:** The procedure of DLF-DG with information of ad hoc networks

If  $WR_j$  has collected information about ad hoc networks in the environment,  $WR_j$  will join the ad hoc network (e.g.  $ANet_k$ ) with the highest number of WRs. After  $WR_j$  joined the network,  $ANet_k$  should also achieve the three objectives of TBWSD. Therefore, the deployment location of  $WR_j$  should be not occupied by the obstacle, having the highest

density value and within the communication range of a WR in  $ANet_k$ .

### The computational complexities of DLF-TP and DLF-DG

In this subsection, the computational complexities of DLF-TP and DLF-DG are analysed. In this analysis,  $M$  number of tasks and  $N$  number of WRs are employed and  $M \gg N$ . In DLF-TP, the main computation consuming process is the calculation of the potential locations for  $WR_j$  (Lines 2 to 6 in both Algorithms 5.3 and 5.4), while in DLF-DG, the main computation consuming process is the update of density values of locations (Lines 3 to 6 and 9 to 11 in Algorithm 5.5). The computational complexity of DLF-TP and DLF-DG are analysed as follows.

#### The computational complexity of DLF-TP

In DLF-TP, all potential locations are calculated from a combination of two given locations. Therefore, the number of potential locations for the  $N^{th}$  WR in an environment with  $M$  number of tasks are  $2M(N - 1) = 2MN - 2M$ . The computational complexity of DLF-TP is at most  $O(2MN)$ .

#### The computational complexity of DLF-DG

In DLF-DG,  $WR_j$  updates the density values of locations according to its discovered tasks. Therefore, for each task (e.g.  $T_i$ ), the density values of locations that need to be updated are in a circle area with the center at  $Loc_i$  and radius equivalent to  $r$ . Since the area of the circle is  $\pi \cdot r^2$ , the size of each location is  $S$  so that for each task there are at most  $\frac{2\pi r^2}{S}$  number of locations, whose density values need to be updated. For the  $N^{th}$  WR in an environment with  $M$  number of tasks, there are at most  $\frac{2M\pi r^2}{S}$  number of locations updated so that the computational complexity is about  $O(\frac{2M\pi r^2}{S})$ .

From the computational complexity of DLF-TP and DLF-DG, it can be seen that if  $r > \sqrt{\frac{N \cdot S}{\pi}}$ , the computational complexity of the DLF-TP is less than that of DLF-DG. If  $r \leq \sqrt{\frac{N \cdot S}{\pi}}$ , the computational complexity of DLF-DG is less than that of DLF-TP.

### 5.4.2 The deployment location updating module

The deployment process creates the current most suitable deployment location for  $WR_j$  based on the collected information during search process. With the enlarging of  $WR_j$ 's local view, new deployment locations (i.e.,  $Loc_{new}$ ) must be more suitable than the current deployment location (i.e.,  $Loc_{deploy}$ ). Since the movement checking module only monitors whether  $WR_j$  has enough energy to move to  $Loc_{deploy}$ , for each  $Loc_{new}$ , the deployment location updating module is employed to check whether  $WR_j$  has enough energy (i.e.,  $Eng_j$ ,



see Definition 5.1) to move to  $Loc_{new}$  from its current location (i.e.,  $Loc_j$ ). In the deployment location updating module, first, the A\* search algorithm is employed to create the path from  $Loc_j$  to  $Loc_{new}$  (i.e.,  $Path(Loc_j, Loc_{new})$ , see Definition 5.4). After the A\* search algorithm created the path, the decision for whether using  $Loc_{new}$  to replace  $Loc_{deploy}$  is made based on Equation 5.5.

$$\begin{cases} Loc_{deploy} = Loc_{new} & |Path(Loc_j, Loc_{new})| \leq Eng_j \\ Loc_{deploy} = Loc_{deploy} & |Path(Loc_j, Loc_{new})| > Eng_j \end{cases} \quad (5.5)$$

If  $WR_j$  has enough energy to move to  $Loc_{new}$ ,  $Loc_{new}$  replaces  $Loc_{deploy}$  and  $Eng_{deploy}$  is updated as  $|Path(Loc_j, Loc_{new})|$ . Otherwise,  $Loc_{deploy}$  will not be changed.

## 5.5 Experiments and analysis

Three experiments were conducted to evaluate the performance of the proposed approach (TBWSD).

**Experiment 1** is to evaluate the performance of the search process. The benchmark of Experiment 1 is the Robot-Sensors Deployment (RSD) approach proposed by Reich et al. [116].

**Experiment 2** is to evaluate the performance of the deployment process with the global view about the environment (DLF-TP and DLF-DG). The benchmark of Experiment 2 is the Dynamic Relays Deployment (DRD) approach proposed by Guo et al.[56].

**Experiment 3** is also to evaluate the performance of the deployment process with only local views about the environment. The benchmark of Experiment 3 is also DRD approach proposed by Guo et al. [56].

### 5.5.1 Experiment settings

In the three experiments, 100 tasks and 100 obstacles are randomly distributed in a disaster environment with  $100 \times 100$  locations. Each location could be occupied by a task or an obstacle. In addition, the sensing and communication distances of WRs are 10 and 15 locations, respectively. The settings of three experiments are shown in Table 5.1.

Table 5.1: The settings of three experiments

Parameters	Values
Size of the environment	$100 \times 100$
Number of obstacles	100
Number of tasks	100
The sensing distance of WRs	10
The communication distance of WRs	15

### The setting of Experiment 1

In Experiment 1, two WRs without prior information, search an environment based on the search strategies of RSD and TBWSD, respectively. The initial location of the two WRs is (1, 1) and the energy of both two WRs could support them to move to 85 locations. RSD uses the blind search strategy, while TBWSD uses the search strategy module of the proposed approach to guide the search. In order to avoid obstacles in the environment, the A\* search algorithm is employed by both RSD and TBWSD. When the two WRs exhaust their energy (i.e., after moved to 85 locations in the environment), the number of tasks discovered by the two WRs (i.e., tasks were or are within the sensing range of the two WRs) is the indicator of this experiment.

### The setting of Experiment 2

In Experiment 2, 10 WRs with the global view about the environment are deployed one by one to establish ad hoc networks according to DRD, DLF-TP and DLF-DG, respectively. DRD is a centralised relays deployment approach, however, the relays deployed by DRD need the assistance of satellites to coordinate with each other in a disaster environment. In this experiment, the performance of ad hoc networks established based on DRD are divided into two situations: with and without the assistance of satellites (DRD with satellite and DRD without satellite). The coverage of tasks (i.e., the number of tasks covered by the established ad hoc networks), the coverage of the environment (i.e., the number of locations covered by the established ad hoc networks) and the objective values (i.e.,  $Objval_k$ , see Equation 5.1) of the established ad hoc networks are the three indicators of this experiment.

### The setting of Experiment 3

In Experiment 3, a WR with different local view (i.e., collected 10% to 100% locations) about the environment is deployed based on TBWSD first. Then, another WR with the global view about the environment is deployed based on DRD. The numbers of tasks covered by the deployed WRs based on TBWSD and DRD are indicators of this experiment.

### 5.5.2 The results of Experiment 1

The results of Experiment 1 are illustrated in Figure 5.6. The X-axis in the figure represents the number of locations that the two WRs moved to, while the Y-axis represents the number of tasks, whose information was collected by WRs.

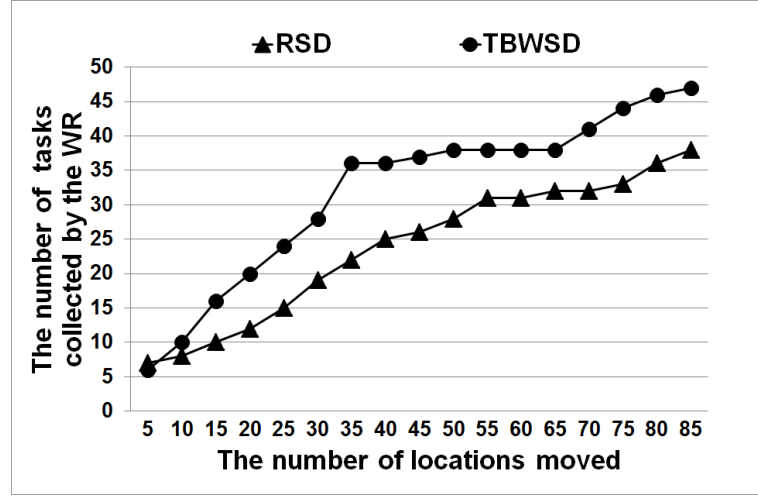


Figure 5.6: The results of Experiment 1

From Figure 5.6, it can be seen that at the beginning of the experiment, the numbers of tasks collected by the two WRs based on the two search strategies were the same. This is because the initial locations of the two WRs based on both RSD and TBWSD were the same. After moved to 10 locations in the environment, the WR based on TBWSD could collect more information of tasks in the environment than that of RSD. This is because the WR based on TBWSD could move to the locations that can cover the maximum tasks in areas, while the WR in RSD only randomly moved to locations in the environment. In addition, based on the search strategy module (i.e., see, Section 5.3.1), the search process of an agent is alternatively moving to a local maximum task coverage location (i.e.,  $Loc_{local}$ , see Section 5.3.1) and moving far away from the last  $Loc_{local}$ . From the line of TBWSD in Figure 5.6, it can be seen that when the WR moves from the 1<sup>st</sup> location to the 35<sup>th</sup> location, the WR is moving to a  $Loc_{local}$  so that the number of collected tasks keeps on increasing. When the WR moves from the 36<sup>st</sup> location to the 65<sup>th</sup> location, the WR is moving far away from the last  $Loc_{local}$ . Since the WR has collected most of tasks at  $Loc_{local}$ , during this period (i.e., from the 36<sup>st</sup> location to the 65<sup>th</sup> location), the number of tasks collected by the WR does not change. When the WR moves from the 66<sup>st</sup> location to the 85<sup>th</sup> location, the WR is moving to another  $Loc_{local}$  so that the number of collected task increases again.

### 5.5.3 The results of Experiment 2

The results of Experiment 2 are illustrated in Figures 5.7, 5.8 and 5.9. In Figures 5.7, 5.8 and 5.9, the X-axes represent the number of deployed WRs, while the Y-axes represent the number of covered tasks, the number of covered locations and the objective values of the established ad hoc networks, respectively.

Figure 5.7 shows the coverage of tasks of established ad hoc networks.

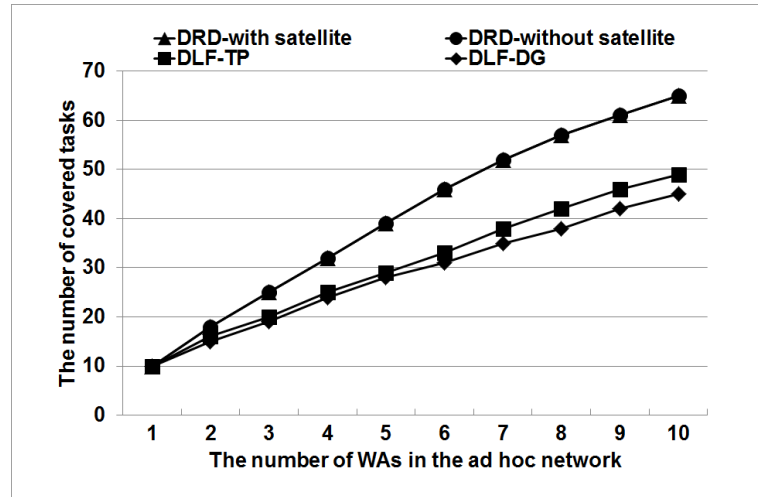


Figure 5.7: The coverage of tasks in Experiment 2

From Figure 5.7, it can be seen that DRD with and without satellites can cover the same amount of tasks. Meanwhile, they could cover much more tasks in the environment than that of DLF-TP and DLF-DG and the numbers of tasks covered by ad hoc networks established based on DLF-TP and DLF-DG were nearly the same. This is because DRD is a centralised approach without the consideration of the communication ranges of WRs in the established networks so that DRD-based approaches could find the most suitable deployment locations for WRs from the whole environment, while in order to guarantee the communication of WRs in the established networks, DLF-TP and DLF-DG had to find such locations from the areas around the WRs that were deployed in the environment. Therefore, DRD-based approaches could always find locations with maximum tasks coverage globally to deploy WRs, while DLF-TP and DLF-DG could only find locations with maximum tasks coverage near the deployment locations of WRs.

Figure 5.8 shows the coverage of the environment of established ad hoc networks.

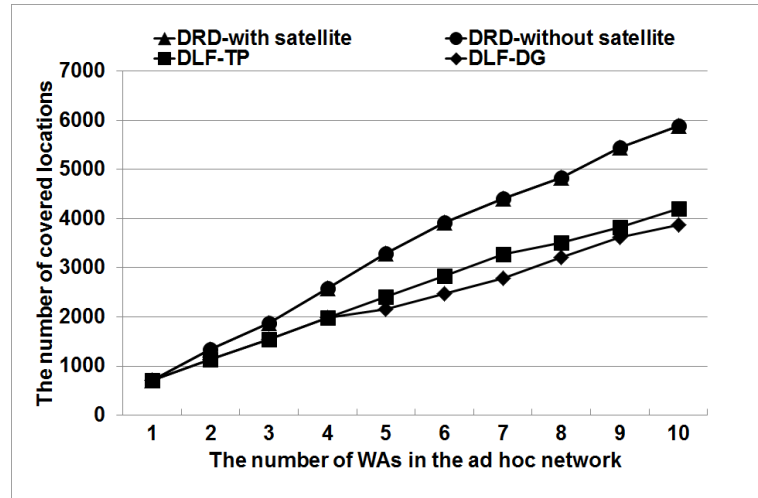


Figure 5.8: The coverage of the environment (i.e., locations) in Experiment 2

From Figure 5.8, it can be seen that DRD with and without satellites can cover much more areas (i.e., including more locations) in the environment than that of DLF-TP and DLF-DG. This is because generally, the locations with global maximum task coverage are far from each other so that overlaps between communication ranges of WRs in the network established based on DRD are small, while in the network established based on DLF-TP and DLF-DG, each deployed WR must lose a certain amount of communication range as the overlap to achieve the communication of WRs, which decreases the coverage area of the established network. In addition, the ad hoc networks established based on DLF-TP can cover more areas (i.e., including more locations) than that of DLF-DG, this is because the distance between two communicable WRs deployed based on DLF-TP must be the same as the WR's communication distance, while the distance between two communicable WRs deployed based DLF-DG may be a little less than the WR's communication distance, which means that ad hoc networks established based on DLF-DG might loss more coverage area as overlaps than those established based on DLF-TP.

Figure 5.9 shows the objective values of established ad hoc networks.

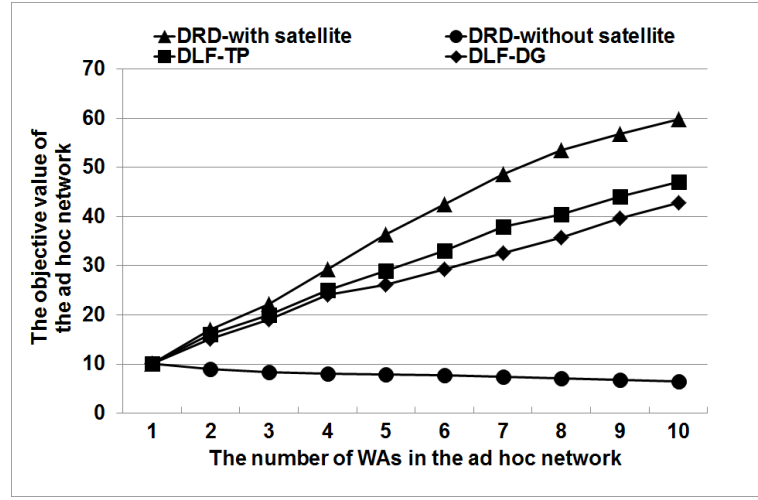


Figure 5.9: The objective values of ad hoc networks in Experiment 2

From Figure 5.9, it can be seen that the objective values (i.e.,  $Objval_k$ , see Equation 5.1) of the ad hoc networks established by DRD with satellites are much higher than those established by other three approaches. This is because with the help of satellites, relays (i.e., WRs), in the established ad hoc network, can freely communicate with each other without the limitation of the communication distance. Therefore, DRD with satellites can be used as the benchmark of this experiment. Although the ad hoc networks established by DRD without satellite can cover more tasks and areas in the environment, without the help of satellites, relays in the network cannot communicate with each other, which greatly decreases the guidance locations of covered tasks (i.e.,  $GLoc_i$ , see Definition 5.5). In DRD without satellite, the guidance location of each covered task equals to the communication range of one WR, which has the worst performance among four approaches. In ad hoc networks, established by DLF-TP and DLF-DG, each WR loses a certain amount of its coverage area (communication range) as the overlap to achieve the communication of WRs. This communication, however, enables information of covered tasks to be shared within the network so that in the ad hoc network established based on DLF-TP and DLF-DG, the guidance locations of each covered task always equal to the coverage area of the entire network. Therefore, the objective values of ad hoc networks (i.e.,  $Objval_k$  see Equation 5.1) established based on DLF-TP and DLF-DG increase linearly with the increase number of WRs.

### 5.5.4 The results of Experiment 3

The results of Experiment 3 is illustrated in Figure 5.10. The X-axis in the figure represents the percentage of locations searched by a WR in the environment, while the Y-axis represents

the numbers of tasks covered by the WR (i.e., within the sensing range of the WR) deployed based on DRD and TBWSD, respectively.

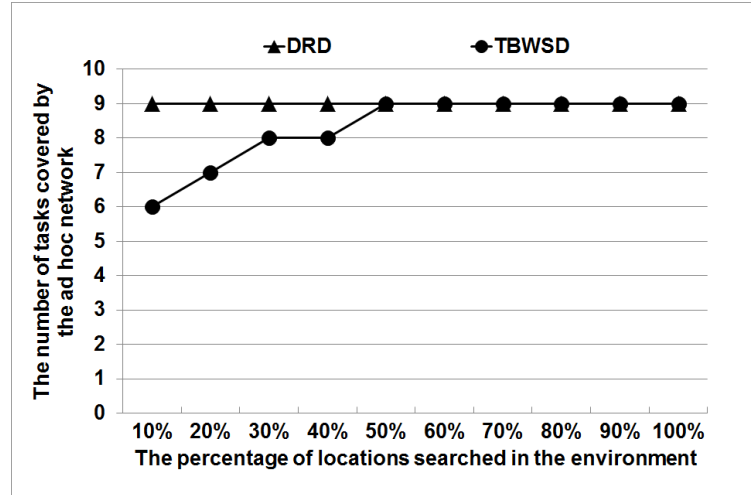


Figure 5.10: The results of Experiment 3

From Figure 5.10, it can be seen that the WR deployed based on DRD can always cover 9 tasks. This is because that in the experiment, DRD could always obtain the global view about the environment so that the number of tasks covered by the WR deployed based on DRD could be taken as the benchmark of this experiment. Different from DRD, the WR, deployed based on TBWSD, could only cover 6 tasks in the environment when the WR searched 10% of locations (i.e., 1000 locations of 10000 locations) in the environment. With the increasing percentage of searched locations, the WR, deployed based on TBWSD, could cover more and more tasks in the environment. When the WR searched 50% percentage of locations (i.e., 5000 locations) in the environment, the WR, deployed based on TBWSD, could cover the same number of tasks in the environment as that of the WR deployed based on DRD. From this experiment, it can be seen that the WR, deployed based on TBWSD, can cover as many tasks as that of the WR deployed based on DRD (i.e., the approach with the global view about the environment), with only the local view about the environment.

### 5.5.5 Findings from experiments

From Experiments 1, 2 and 3, we have the following discoveries.

1. Based on the search process, WRs with limited energy and capabilities can efficiently search to collect information for task allocation in unknown and complex disaster environments (to achieve **Objective 1**, see Section 1.3);

2. Based on the deployment process, WRs in the established ad hoc networks are communicable and can cover as many tasks and areas in disaster environments as possible (to achieve **Objectives 6 and 7**, see Section 1.3);
3. Based on the WR search and deployment approach, the ad hoc network established by WRs with local views about a disaster environment is as good as that established by WRs with global views about the environment. (to achieve **Objective 2**, see Section 1.3);

## 5.6 Summary

In this chapter, a WR search and deployment approach was proposed for ad hoc network establishment in disaster environments. First, the definitions of the WR search and deployment approach were given. Then, the technical designs of two processes of the WR search and deployment approach for the establishment of ad hoc network in disaster environments were introduced in detail. Finally, experiments to evaluate the performance of the WR search and deployment approach were demonstrated and the results were analysed.



# Chapter 6

---

## Two Wireless Mobile Robots Deployment Approaches

In many disaster environments, the number of WRs usually is much less than the number of important locations in the environment so that how to maximise the important locations covered by the established ad hoc network is the primary objective of WR deployment approaches for task allocation in such circumstances. To maximise the coverage of important locations, most of current approaches in the literature were developed based on greedy algorithms. Due to the myopia of greedy algorithms, these approaches can only maximise the coverage of important locations of each WR rather than that of the whole network. In this chapter, two mathematical programming-based WR deployment approach are proposed for ad hoc network establishment in disaster environments. In particular,

- a linear programming (LP)-based WR deployment approach can find suitable deployment locations for a limited number of WRs (to solve **Challenging Issue 8**, identified in Section 1.2) to cover the maximum important locations and areas in disaster environments (to solve **Challenging Issue 6**, identified in Section 1.2) by considering multiple constraints of environments and limited sensing and communication capabilities of WRs (to solve **Challenging Issue 2**, identified in Section 1.2);
- a quadratic programming (QP)-based WR deployment approach can create the same deployment locations for the WRs as the LP-based approach with less computational complexity.

### 6.1 Problem Description and Definitions

Let  $D$  be a 2-dimension disaster environment, which are divided into a number of equivalent size square locations. At a time  $T$ ,  $M$  number of ILs are distributed in  $D$ . Let  $ILSet = \{IL_1, IL_2, IL_3, ..., IL_M\}$  represent all ILs, where  $IL_i$  represents the  $i^{th}$  IL and  $1 \leq i \leq M$ . At

the same time,  $N$  number of WRs are going to establish the ad hoc network in  $D$ . Let  $\langle WR_1, WR_2, WR_3, \dots, WR_N \rangle$  represent all WRs, where  $WR_j$  represents the  $j^{th}$  WR and  $1 \leq j \leq N$ . In addition, the number of ILs is much more than the number of WRs (i.e.,  $M \gg N$ ). The IL is formally defined as follows.

**Definition 6.1.** An *important location* ( $IL_i$ ) can be defined as a two-tuple  $IL_i = \langle ILNo, ILoc_i \rangle$ , where  $ILNo$  is the ID of  $IL_i$  and  $ILoc_i$  is the location of  $IL_i$ .

The term of a WR was defined in Definition 5.1 in Chapter 5. To fit the ad hoc network establishment problem in this chapter, it is redefined in this chapter as follows.

**Definition 6.2.** A *wireless mobile robot*  $WR_j$  can be defined as a two-tuple  $WR_j = \langle WRNo, WLoc_j \rangle$ , where  $WRNo$  is the ID of  $WR_j$  and  $WLoc_j$  is the deployment location of  $WR_j$ .

The distance between two locations in  $D$  is calculated by Euclidean distance [90], [50]. Since WRs relies on wireless technologies, the sensing distances of WRs are limited. We assume that the sensing distances of all WRs in an environment are same and equal to  $r$  so that if the distance between locations of an IL (e.g.  $IL_i$ ) and a WR (e.g.  $WR_j$ ) is less than or equals to  $r$  (i.e.,  $Dis(ILoc_i, WLoc_j) \leq r$ , see Definition 6.1 and 6.2), we say that  $IL_i$  is covered by  $WR_j$ .

The objective of the MILCP is to deploy all WRs in a disaster environment to establish an ad hoc network (i.e.,  $ANet$ ) so as to maximise the number of ILs covered by the deployed WRs in  $ANet$ . The objective value  $objval$  of  $ANet$  can be calculated by Equation 6.1.

$$objval = \underset{WLoc_j}{argmax} \sum_{\forall IL_i \in ILSet} Cover(IL_i), \quad (6.1)$$

where  $Cover(IL_i)$  is a Boolean function. If  $IL_i$  is covered by at least one deployed WR in  $ANet$ ,  $Cover(IL_i) = 1$ ; Otherwise,  $Cover(IL_i) = 0$ ; The value of  $Objval$  is an integer number between 0 and  $M$ , where 0 and  $M$  represent that  $ANet$  does not or does achieve the objective of the MILCP, respectively.

### 6.1.1 The NP-hardness and non-linearity of the MILCP

In this subsection, the NP-hardness and non-linearity of the MILCP are proven.

**Proposition 6.1.** *The MILCP is an NP-hard problem.*

*Proof.* To prove the NP-hardness of the MILCP, we first introduce a well-studied NP-hard problem, which is termed as the Minimum WR Deployment Problem (MWRDP) [56]. The MWRDP can be described as that given  $M$  number of ILs in an environment and the sensing distances of WRs (i.e.,  $r$ ), how to find the minimum number of deployment locations for WRs, at which all ILs in the environment can be covered by at least one WR. Zhu et al. [155] have approved that the MWRDP is an NP-hard problem so that if the MILCP can be proven to have the same computational complexity as the MWRDP, the MILCP is NP-hardness. First, we can use Guo et al.'s binary integer programming approach (i.e., Subsection 5.3. in [56]) for the MWRDP to create the minimum number of deployment locations (i.e.,  $N'$ ) for WRs to cover all  $M$  number of ILs in the environment. Comparing  $N'$  (i.e., the minimum number of WRs for the MWRDP) to  $N$  (i.e., the number of WRs of the MILCP), there could be two kinds of situations.

**Situation 1:**  $N' \leq N$

In this situation, the MILCP is the same as the MWRDP, so the MILCP is an NP-hard problem.

**Situation 2:**  $N' > N$

In this situation, the following two steps are employed, which are

- 1) to eliminate some ILs from the environment and
- 2) to employ the approach for the MWRDP again to get another  $N'$  for the remaining ILs.

The above two steps are repeated until  $N' \leq N$ . Then, same as Situation 1, the MILCP is the same as the MWRDP again.

Therefore, no matter in what situation, the MILCP is at least an NP-hard problem.  $\square$

**Proposition 6.2.** *The MILCP is a non-linear problem.*

*Proof.* The objective of the MILCP is to deploy  $N$  number of WRs in a disaster environment to establish an ad hoc network so as to maximise the number of ILs covered the established ad hoc network. However, two deployed WRs can cover duplicated ILs so that in order to calculate the maximum number of ILs covered by the network, the number of duplicated ILs covered by any two WRs in the network should be eliminated from the sum number of ILs covered by all WRs in the established ad hoc network. The number of duplicated ILs covered by any two WRs can only be calculated based on a quadratic function so that the MILCP is a non-linear problem.  $\square$

## 6.2 Non-greedy Algorithm-based Approaches vs. Greedy Algorithm-based Approaches

To handle the WR deployment, the difference between greedy algorithm-based approaches and non-greedy algorithm-based approaches is demonstrated by the following example. Supposing there are a number of ILs distributed in a disaster environment and two WRs are going to be deployed to maximise the coverage of ILs, which are shown in Figure 6.1.

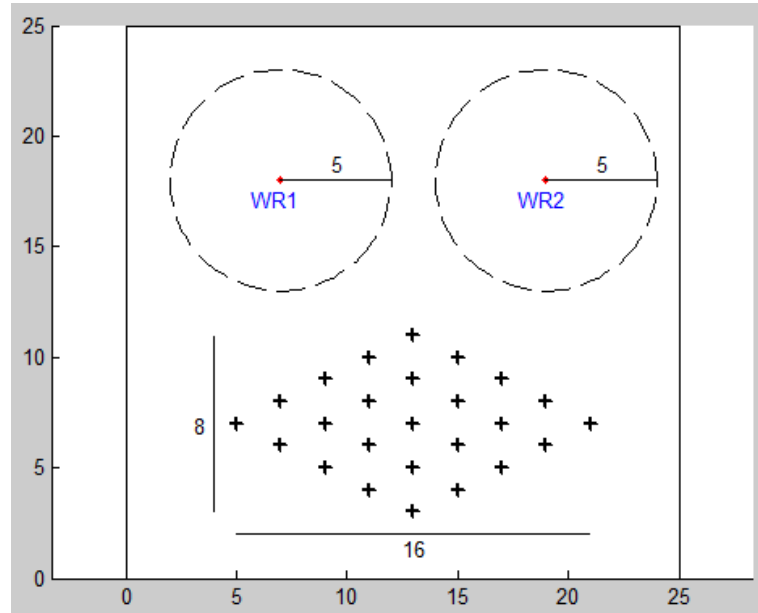


Figure 6.1: An example of WAs deployment

In Figure 6.1, the crosses represent ILs in the environment, which form a diamond area. The longest and the highest distances of the diamond area are 16 and 8, respectively. Two points represent the two WRs and the areas inside the dash lines represent their sensing and communication ranges, both of which are 5. The WR deployment based on the non-greedy algorithm-based approach and the greedy algorithm-based approach is shown in Figure 6.2.

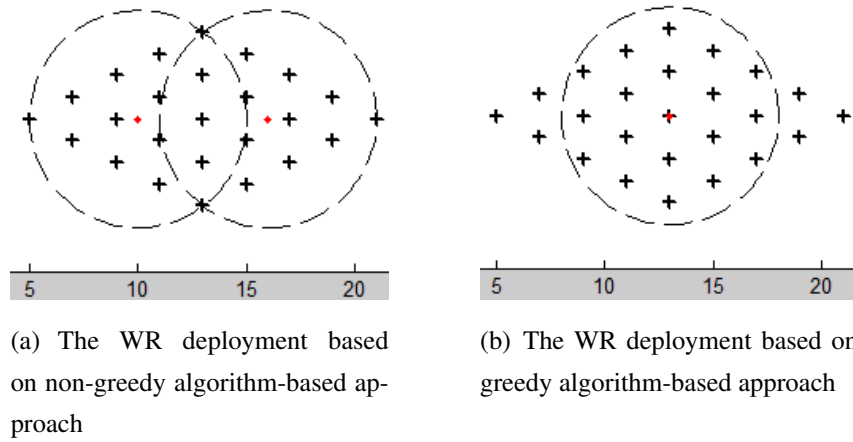


Figure 6.2: The WR deployment between two kinds of approaches

From Figure 6.2 (a), it can be seen that based on the non-greedy algorithm-based approach, the two deployed WRs can cover all ILs in the environment (as shown in Figure 6.2 (a)). However, based on the greedy algorithm approach, the first WR will be deployed at the centre of the diamond area (as shown in Figure 6.2 (b)). In this situation, without adjusting the deployment location of the first WR, no matter how to deploy the second WR, it is impossible for the two WRs to cover all ILs in the environment.

Theoretically, the maximum difference on ILs coverage between the non-greedy algorithm-based approaches and the greedy algorithm-based approaches can reach 25%. Supposing there are 4 ILs arranging in a line and intervals between any two adjacent ILs are 10 and the two WRs employed in the last example are going to cover the four ILs. The maximum difference of the IL coverage between WRs deployed by the non-greedy algorithm-based approach and the greedy algorithm-based approach is shown in Figure 6.3.

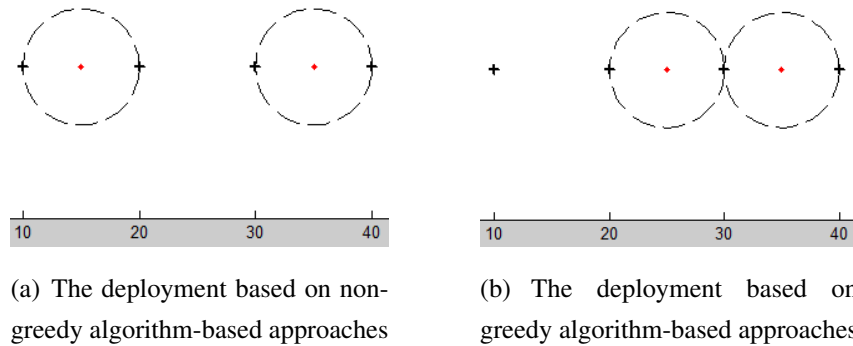


Figure 6.3: The maximum difference of the IL coverage between WRs deployed by two kinds of approaches

From Figure 6.3, it can be seen that based on the non-greedy algorithm-based approach,

the two deployed WRs can cover all 4 ILs in the environment (as shown in Figure 6.3 (a)). However, based on the greedy algorithm-based approach, the two deployed WRs could only cover 3 ILs in the environment (as shown in Figure 6.3 (b)). In this situation, the difference on the IL coverage between two kinds of approaches is 25%.

From the above examples, it can be seen that in some circumstances, the IL coverage between the non-greedy algorithm-based approaches and the greedy algorithm-based approaches for all the situations have a significant difference.

## 6.3 The Linear Programming-based Approach for the MILCP

In order to create the suitable deployment locations for WRs in a disaster environment for the MILCP, a LP-based approach, adjusted from Guo et al.'s binary integer programming approach (i.e., Subsection 5.3. in [56]) for the MWRDP, is proposed. In WR deployment problem, the MWRDP is one of the well-studied problems and can be handled by the LP. From the description of the MWRDP (i.e., Proof of Proposition 1 in Subsection 6.1.1), it can be seen that the MWRDP is also based on the assumption of having enough WRs to cover all ILs in an environment.

The proposed LP-based approach aims at solving the MILCP by removing the assumption of the MWRDP.

The finding of potential deployment locations (i.e.,  $PLSet$ ), the construction of the covering relationship matrix (i.e.,  $\beta$ ), the LP formulation of the MWRDP, and the proposed LP-based approach are introduced in detail in the following subsections.

### 6.3.1 The finding of potential deployment locations

In a disaster environment, WRs can be deployed at any locations. However, most of these deployment locations are not very useful, at which WRs cannot cover or can only cover one or two ILs. In order to reduce the computational complexities in the following steps, the potential deployment locations should be found by removing these useless deployment locations. Since the coverage area of a WR is a circle, the problem of deploying WRs to cover a number of ILs is the same as the problem of using circles to cover scattered points in a plane. In papers [56], [62], authors have approved that if a fixed-size circle can cover the maximum number of scattered points in a two-dimensional plane, there are at least two points on the border of the circle so that potential deployment locations of WRs can be found based on Proposition 6.3 described as follows.

**Proposition 6.3.** *Given two ILs (i.e.,  $IL_1$  and  $IL_2$ ) and their locations (i.e.,  $ILoc_1$  and  $ILoc_2$ ) in an environment and the sensing distance of  $WR_j$  (i.e.,  $r$ ), if the distance between  $ILoc_1$  and  $ILoc_2$  is less than or equals to  $2r$ , potential deployment locations (i.e.,  $PLoc_k$ ) can be found. If  $WR_j$  is deployed at  $PLoc_k$ ,  $IL_1$  and  $IL_2$  are on the border of the sensing range of  $WR_j$ .*

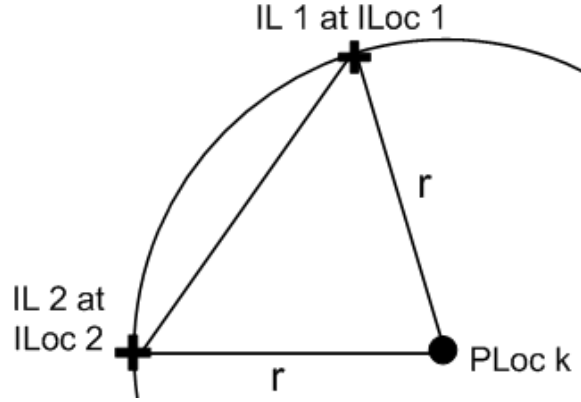


Figure 6.4: The potential deployment locations found based on Proposition 6.3

*Proof.* As shown in Figure 6.4, the two crosses are locations of two ILs (i.e.,  $IL_1$  at  $ILoc_1$  and  $IL_2$  at  $ILoc_2$ ); the circle area inside the solid line is the sensing range of  $WR_j$  (i.e., the sensing distance  $r$ ); and the point is the potential deployment location (i.e.,  $PLoc_k$ ), found based on  $ILoc_1$  and  $ILoc_2$ . Based on the definition of the circle, the relationship between coordinates of  $Loc_1 = (x_1, y_1)$ ,  $Loc_2 = (x_2, y_2)$  and  $PLoc_k = (x_k, y_k)$  are described as Equation 6.2.

$$\begin{cases} (x_1 - x_k)^2 + (y_1 - y_k)^2 = r^2 \\ (x_2 - x_k)^2 + (y_2 - y_k)^2 = r^2 \end{cases} \quad (6.2)$$

When  $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \leq 2r$ ,  $x_k$  and  $y_k$  in Equation 6.2 have two solutions for each combination of two ILs.  $\square$

Proposition 6.3 is described as  $PLoc_k = PLFound(IL_i, IL_{i'}, r)$  in this chapter. Based on Proposition 6.3, all potential deployment locations can be found based on all ILs (i.e.,  $ILSet$ ) in a disaster environment, which is described in Algorithm 6.1.

**Algorithm 6.1:** The finding of all potential deployment locations

---

```

1  $PLSet = \emptyset$ 
2 for each  $IL_i \in ILSet$  do
3   for each  $IL_{i'} \in ILSet$  do
4     if  $i \neq i' \ \&\& \ Dis(IL_i, IL_{i'}) \leq 2r$  then
5        $PLoc_k = PLFound(IL_i, IL_{i'}, r)$ 
6        $PLSet = PLSet \cup PLoc_k$ 
7     end
8   end
9 end

```

---

Algorithm 6.1 is explained as follows. At the beginning,  $PLSet$  is initialised to  $\emptyset$  (Line 1). After that, for any two different ILs (i.e.,  $IL_i$  and  $IL_{i'}$ ), if the Euclidean distance of their locations is not more than  $2r$  (i.e.,  $Dis(IL_i, IL_{i'}) \leq 2r$ ), potential deployment locations (i.e.,  $PLoc_k$ ) are found based on Proposition 6.3 and recorded in  $PLSet$  (Lines 2 to 6).

After employing Algorithm 6.1, all potential deployment locations in a disaster environment are found and recorded as  $PLSet = \{PLoc_1, PLoc_2, PLoc_3, \dots, PLoc_L\}$ , where  $L$  is the number of potential deployment locations and  $PLoc_k$  represents the  $k^{th}$  potential deployment location.

### 6.3.2 The construction of the covering relationship matrix

After finding all potential deployment locations  $PLSet$ , the covering relationships between ILs in  $ILSet$  and potential deployment locations in  $PLSet$  can be calculated and represented by an  $M \times L$  coefficient matrix  $\beta$ , which is described in Equation 6.3.

$$\beta = \begin{bmatrix} b_{1,1} & b_{1,2} & \dots & b_{1,L} \\ b_{2,1} & b_{2,2} & \dots & b_{2,L} \\ \dots & \dots & \dots & \dots \\ b_{M,1} & b_{M,2} & \dots & b_{M,L} \end{bmatrix}, \quad (6.3)$$

where  $M$  is the number of ILs in  $ILSet$ ,  $L$  is the number of potential deployment locations in  $PLSet$  and each element  $b_{i,k}$  in  $\beta$  is a binary value, if the  $i^{th}$  IL (i.e.,  $IL_i$ ) can be covered by the WR at the  $k^{th}$  potential deployment location (i.e.,  $PLoc_k$ ),  $b_{i,k} = 1$ ; Otherwise,  $b_{i,k} = 0$ .



### 6.3.3 The LP formulation of the MWRDP

Based on potential deployment locations (i.e.,  $PLSet$ , see Subsection 6.3.1) and the covering relationship matrix (i.e.,  $\beta$ , see Subsection 6.3.2), the MWRDP can be formulated by the LP [56]. The variables of this LP formulation are represented by a binary vector  $X = \{x_1, x_2, x_3, \dots, x_L\}$ , where  $x_k$  is a binary value and 0 and 1 represent the  $k^{th}$  potential deployment location is not chosen or is chosen as a deployment location for a WR, respectively. The LP formulation of the MWRDP is described in Equation 6.4.

$$\min \sum_{k=1}^L x_k \quad (6.4a)$$

$$\text{s.t. } x_k \in \{0, 1\}, \quad 1 \leq k \leq L \quad (6.4b)$$

$$\sum_{k=1}^L b_{i,k} \cdot x_k \geq 1, \quad 1 \leq i \leq M \quad (6.4c)$$

Equation 6.4 is explained as follows. The objective of the LP formulation of the MWRDP is to minimise the number of chosen deployment locations (i.e., Equation 6.4a). Then, the value of each  $x_k$  in  $X$  should be a binary value, which is either 0 or 1 (i.e., Equation 6.4b). Finally, the LP formulation of the MWRDP can ensure that each IL in the environment (i.e.,  $\forall IL_i \in ILSet$ ) should be covered by at least one WR (i.e., Equation 6.4c).

In this chapter, the LP formulation of the MWRDP is described as  $X = LPF(PLSet, \beta)$ , where  $ILSet$  is the set of ILs in the environment, and  $PLSet$  is the set of potential deployment locations and  $\beta$  is the  $M \times L$  coefficient matrix, where  $M$  and  $L$  are the number of ILs in  $ILSet$  and the number of potential deployment locations in  $PLSet$ , respectively.

### 6.3.4 The proposed LP-based approach

In this subsection, the LP formulation of the MWRDP is adjusted to solve the MILCP. The idea of this adjustment can be described as follows. Suppose that we employ the LP formulation of the MWRDP to create the minimum number of WRs (i.e.,  $N'$ ) to cover all ILs in a disaster environment based on  $PLSet$  and  $\beta$ . Comparing  $N'$  with  $N$  of the MILCP (i.e., the number of WRs in the environment), there are two situations, which are 1)  $N' \leq N$  and 2)  $N' > N$ .

#### Situation 1: $N' \leq N$

In this situation, the number of WRs in the environment is enough to cover all ILs in the environment, the deployment locations of WRs in the LP formulation of the MWRDP are also the solution for WRs in the MILCP.

**Situation 2:**  $N' > N$ 

In this situation, since the number of WRs in the environment is not enough to cover all ILs in the environment, we can repeat the following two steps, which are

**Step 1:** eliminating a number of ILs from  $ILSet$  to obtain  $ILSet'$  and reconstruct  $PLSet'$  and  $\beta'$  based on  $ILSet'$  (i.e., see Subsections 6.3.1 and 6.3.2); and

**Step 2:** employing the LP formulation of the MWRDP to create a new minimum number of WRs  $N'$  for  $ILSet'$ ,  $PLSet'$  and  $\beta'$ .

With the increasing number of eliminated ILs from  $ILSet$ , the above two steps are repeated until  $N \geq N'$ . However, for the same number of eliminated ILs, if the eliminated ILs are different, different  $N'$  created by the LP formulation of the MWRDP are various. In order to obtain the optimal deployment locations for WRs, for each number of eliminated ILs (i.e.,  $en$ ), all  $en$ -combinations of eliminated ILs should be calculated based on the LP formulation of the MWRDP. The proposed LP-based approach is described in Algorithm 6.2.

---

**Algorithm 6.2:** The proposed LP-based approach

---

```

1   $en = 0$ 
2   $(N'_{en}, X'_{en}) \leftarrow LPF(PLSet, \beta)$ 
3  repeat
4       $en = en + 1$ 
5       $Temp = \emptyset$ 
6       $ESet_{en} \leftarrow$  all  $en$ -combinations of ILs
7      for each  $e_{en,q} \in ESet_{en}$  do
8           $ILSet' = ILSet \setminus e_{en,q}$ 
9          Reconstruct  $PLSet'$  based on  $ILSet'$ 
10         Reconstruct  $\beta'$  based on  $ILSet'$  and  $PLSet'$ 
11          $(N'_{en,q}, X'_{en,q}) \leftarrow LPF(PLSet', \beta')$ 
12          $Temp = Temp \cup (N'_{en,q}, X'_{en,q})$ 
13     end
14      $(N'_{en}, X'_{en}) \leftarrow Min(N'_{en,q} \in Temp)$ 
15 until  $N \geq N'_{en}$ ;
16  $X'_{en}$  is the deployment location of the MILCP

```

---

Algorithm 6.2 is explained as follows. At the beginning, the number of eliminated ILs (i.e.,  $en$ ) is initialised to 0 (Line 1). Then, the LP formulation of the MWRDP is employed

to create the solution (i.e.,  $N'_{en}$  and  $X'_{en}$ ) for  $ILSet$ ,  $PLSet$  and  $\beta$  (Line 2). If the number of WRs in the MILCP is less than the minimum number of WRs to cover all ILs in the environment (i.e.,  $N < N'_{en}$ ), we begin to eliminate ILs from  $ILSet$  and the number of eliminated ILs increases (i.e.,  $en$  increases from 1 to  $M$ ) (Lines 3 to 4). For each value of  $en$ , the following four steps are executed in order.

**Step 1:** The temporary set  $Temp$  is initialised to  $\emptyset$  (Line 5);

**Step 2:** All  $en$ -combinations of ILs (i.e.,  $C_M^{en}$  kinds of combinations) are calculated and recorded in  $ESet_{en} = \{e_{en,1}, e_{en,2}, \dots, e_{en,Q}\}$ , where  $e_{en,q} \subset ILSet$  and  $|e_{en,q}| = en$  (Line 6);

**Step 3:** For each combination of  $en$  number of ILs (i.e.,  $e_{en,q} \in ESet_{en}$ ),  $ILSet'$  is obtained from eliminating  $IL_i \in e_{en,q}$  from  $ILSet$ ,  $PLSet'$  is reconstructed based on  $ILSet'$  (i.e., see Subsection 6.3.1) and  $\beta'$  is reconstructed based on  $ILSet'$  and  $PLSet'$  (i.e., see Subsection 6.3.2) (Lines 8 to 10). Then, the LP formulation of the MWRDP is employed to create solutions (i.e.,  $N'_{en,q}$  and  $X'_{en,q}$ ) for  $PLSet'$  and  $\beta'$ , which are recorded in  $Temp$  (Lines 11 and 12); and

**Step 4:** After creating the solutions for all  $en$ -combinations of ILs (i.e.,  $\forall e_{en,q} \in ESet_{en}$ ), the solution with the minimum value of  $N'_{en,q}$  is chosen as the final solution of  $en$  number of eliminated ILs and recorded in  $(N'_{en}, X'_{en})$  (Line 13).

The above four steps (Lines 3 to 13) will be repeated until  $N'_{en}$  in the optimal solution of  $en$  number of eliminated ILs is less than or equals to  $N$  (i.e.,  $N \geq N'_{en}$ ) (Line 14). Then,  $X'_{en}$  contains the optimal deployment locations for  $N$  number of WRs in the MILCP (Line 15).

### 6.3.5 The computational complexity of the LP-based approach

In this subsection, the computational complexity of the LP-based approach is analysed. In this analysis, it is assumed that there are  $M$  number of ILs,  $N$  number of WRs and  $L$  number of potential deployment locations in the environment. In addition, the potential deployment locations are derived from ILs (i.e., see Subsection 6.3.1), since the number of potential deployment locations mainly depends on the distance between any two ILs,  $M$  and  $L$  do not have a direct relationship. In the LP-based approach, the main computational consuming process is to calculate combinations of eliminated ILs (see Lines 6 to 13 in Algorithm 6.2). All combinations of all numbers of eliminated ILs need to be calculated for  $\sum_{en=0}^M \binom{M}{en}$  times. However, in most of time, the WRs do not need to calculate through all combinations of all

numbers of eliminated ILs and the calculation of the LP-based approach ends when  $N \geq N'_{en}$ . Generally, the computational complexity of the LP-based approach is not stable, which mainly depends on the difference between the number of WRs in the environment (i.e.,  $N$ ) and the minimum number of WRs that can cover all ILs in the environment (i.e.,  $N'$ , the solution of the LP formulation of the MWRDP). If the difference between  $N$  and  $N'$  is big, the LP-based approach needs more computations to find the solution for the MILCP and vice versa.

## 6.4 The QP-based approach for the MILCP

Due to the unstable computational complexity of the LP-based approach for the MILCP, sometimes, it is not efficient to employ the LP-based approach to create the deployment locations for WRs in a disaster environment. To this end, a Quadratic Programming (QP) formulation of the MILCP is proposed. The basic principle of the QP-based approach for the MILCP is described as follows.

1. All potential deployment locations that can cover a number of ILs are found and recorded in  $PLSet$  (i.e., see Subsection 6.3.1).
2. The covering relationships between ILs in  $ILSet$  and potential deployment locations  $PLSet$  are calculated and recorded in an  $M \times L$  coefficient matrix  $\beta$  (i.e., the covering relationship matrix, see Subsection 6.3.2), where  $M$  and  $L$  are the number of ILs and the number of potential deployment locations in the environment, respectively.
3. The duplicated covering relationships between any two potential deployment locations are calculated and recorded in an  $L \times L$  upper triangle matrix  $\alpha$  (i.e., the duplicated covering matrix), where  $L$  is the number of potential deployment locations in the environment.
4. The QP-based approach for the MILCP is constructed based on  $PLSet$ ,  $\beta$  and  $\alpha$ .

In this section, the construction of the duplicated covering matrix (i.e.,  $\alpha$ ) and the QP-based approach for the MILCP are introduced in detail in the following subsections.

### 6.4.1 The construction of the duplicated covering matrix

In this subsection, the duplicated covering relationships between WRs at two potential deployment locations in  $PLSet$  can be calculated and represented by an  $L \times L$  upper triangular matrix  $\alpha$ , which is described in Equation 6.5.

$$\alpha = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,L} \\ 0 & a_{2,2} & \dots & a_{2,L} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{L,L} \end{bmatrix}, \quad (6.5)$$

where  $L$  is the number of potential deployment locations, each element  $a_{k_1, k_2}$  in  $\alpha$  is a positive integer value to represent the number of duplicated ILs covered by WRs at the  $(k_1)^{th}$  and the  $(k_2)^{th}$  potential deployment locations.

The procedure of constructing  $\alpha$  is described in Algorithm 6.3.

---

**Algorithm 6.3:** The procedure of constructing  $\alpha$

---

```

1  for each  $PLoc_k \in PLSet$  do
2       $Cov_k = \emptyset$ 
3      for each  $IL_i \in ILSet$  do
4          if  $Dis(PLoc_k, IL_i) \leq r$  then
5               $Cov_k = Cov_k \cup IL_i$ 
6          end
7      end
8  end
9  for each  $PLoc_{k_1} \in PLSet$  do
10     for each  $PLoc_{k_2} \in PLSet$  do
11          $a_{k_1, k_2} = 0$ 
12         if  $k_1 < k_2$  then
13             for each  $IL_i \in ILSet$  do
14                 if  $IL_i \in Cov_{k_1} \ \&\& \ IL_i \in Cov_{k_2}$  then
15                      $a_{k_1, k_2} = a_{k_1, k_2} + 1$ 
16                      $Cov_{k_2} = Cov_{k_2} \setminus IL_i$ 
17                 end
18             end
19         end
20         if  $k_1 = k_2$  then
21              $a_{k_1, k_2} = |Cov_{k_1}|$ 
22         end
23     end
24 end

```

---

Algorithm 6.3 is explained as follows. For each potential deployment location (i.e.,  $PLoc_k$ ), the set of covered ILs (i.e.,  $Cov_k$ ) by the WR at  $PLoc_k$  is initialised to  $\emptyset$  (Lines 1 to 2). For each IL (i.e.,  $IL_i$ ), if  $IL_i$  can be covered by the WR at  $PLoc_k$ ,  $IL_i$  is included into  $Cov_k$  (Lines 3 to 5). For any two potential deployment locations (i.e.,  $PLoc_{k_1}$  and  $PLoc_{k_2}$ ), the number of duplicated ILs between  $PLoc_{k_1}$  and  $PLoc_{k_2}$  (i.e.,  $a_{k_1,k_2}$ ) is initialised to 0 (Line 6 to 8). After that, if  $k_1$  is less than  $k_2$ , for each duplicated IL (i.e.,  $IL_i$ ) covered by WRs at  $PLoc_{k_1}$  and  $PLoc_{k_2}$ ,  $a_{k_1,k_2}$  can increase one (Lines 9 to 12). Finally, to avoid the repetitive computation of duplicated ILs,  $IL_i$  is only kept in the set of covered ILs at  $PLoc_{k_1}$  (i.e.,  $Cov_{k_1}$ ) and is eliminated from the set of covered ILs at  $PLoc_{k_2}$  (i.e.,  $Cov_{k_2}$ ) (Line 13). If  $k_1$  equals to  $k_2$ ,  $a_{k_1,k_2}$  equals to the number of covered ILs by the WR at  $PLoc_{k_1}$  (Lines 14 to 15).

### 6.4.2 The quadratic programming formulation of the MILCP

Based on potential deployment locations (i.e.,  $PLSet$ ), the covering relationship matrix  $\beta$  and the duplicated covering matrix  $\alpha$ , the MILCP can be formulated by quadratic programming (QP). Same as the variables in the LP formulation of the MWRDP (i.e., see Subsection 6.3.3), the variables of this QP formulation are represented by a binary vector  $X = \{x_1, x_2, x_3, \dots, x_L\}$  as well, where  $x_k$  is a binary value and 0 and 1 represent that the  $k^{th}$  potential deployment location is not chosen or is chosen as a deployment location for a WR, respectively. The QP-based approach for the MILCP is described in Equation 6.6.

$$\max \quad \sum_{k=1}^L (x_k \sum_{i=1}^M b_{i,k}) - \sum_{k_1=1}^L \sum_{k_2=1}^L x_{k_1} x_{k_2} a_{k_1,k_2} \quad (6.6a)$$

$$\text{s.t.} \quad \sum_{k=1}^L x_k = N \quad (6.6b)$$

$$x_k \in \{0, 1\} \quad 1 \leq k \leq L \quad (6.6c)$$

$$\sum_{k=1}^L b_{i,k} x_k \geq 0 \quad 1 \leq i \leq M \quad (6.6d)$$

Equation 6.6 is explained as follows. The objective of the QP formulation is to choose  $N$  number of deployment locations from all potential deployment locations (i.e.,  $PLSet$ ) so as to maximise the number of ILs covered by all WRs in an ad hoc network, which is calculated from the difference between the sum of ILs covered by each chosen deployment location (i.e.,  $\sum_{k=1}^L (x_k \sum_{i=1}^M b_{i,k})$ ) and the sum of duplicated ILs covered by any two chosen deployment locations (i.e.,  $\sum_{k_1=1}^L \sum_{k_2=1}^L x_{k_1} x_{k_2} a_{k_1,k_2}$ ) (i.e., Equation 6.6a). The number of

chosen deployment locations should equal to  $N$  (i.e., Equation 6.6b). In addition, the value of each  $x_k$  in  $X$  should be a binary value, which is either 0 or 1 (i.e., Equation 6.6c). Finally, each IL in  $ILSet$  may be or may not be covered by WRs (i.e., Equation 6.6d).

### 6.4.3 The computational complexity of the QP formulation

In this subsection, the computational complexity of the QP-based approach for the MILCP is analysed. The same as the analysis of the computational complexity of the LP-based approach (i.e., see Subsection 6.3.5), there are also  $M$  number of ILs,  $N$  number of WRs and  $L$  number of potential deployment locations in the environment. In the QP formulation, the main computational consuming process is the construction of the upper triangular matrix  $\alpha$  (i.e., see Subsections 6.4.1). In order to construct the  $L \times L$  adjacent matrix  $\alpha$ , the WRs need to calculate for  $\frac{L(L+1)}{2}$  times. Totally, the computational complexity of the QP formulation is  $O(\frac{L(2M+L+1)}{2})$ . From this calculation, it can be seen that the computational complexity of the QP-based approach for the MILCP is only related to the numbers of ILs and potential deployment locations. Therefore, comparing to the LP-based approach, the computational complexity of the QP formulation is stable, which is proportional to  $M$  and  $L$ .

### 6.4.4 The extension of the QP-based approach for the MILCP

From Equation 6.6, it can be seen that the number of ILs covered by the established ad hoc network can be calculated based on potential deployment locations  $PLSet$ , the covering relationship matrix  $\beta$  and the duplicated covering matrix  $\alpha$ . Based on this, the QP-based approach for the MILCP can be extended to calculate the numbers of multiple kinds of locations, such as less important locations, worthless locations, etc. covered by the established network. If WRs assign each kind of covered locations a value to represent its contribution to the established network, the utility of the established ad hoc network can be calculated and maximised by the QP.

Based on this consideration, the extension of the QP-based approach for the MILCP is proposed. The basic principle of the extension of the QP-based approach for the MILCP is described as follows.

1. All  $C$  kinds of locations and their contribution values to the utility of the established ad hoc network are identified and recorded in  $CSet = \{LSet_1, LSet_2, \dots, LSet_C\}$  and  $\{\lambda_1, \lambda_2, \dots, \lambda_C\}$ , respectively.
2. All potential deployment locations that can deploy WRs are found and recorded in  $PLSet$ .

3. The covering relationship matrices (i.e.,  $\{\beta_1, \beta_2, \dots, \beta_C\}$ ) are constructed based on all kinds of locations in  $CSet$  and potential deployment locations  $PLSet$ .
4. The duplicated covering matrices (i.e.,  $\{\alpha_1, \alpha_2, \dots, \alpha_C\}$ ) are constructed based on all kinds of locations in  $CSet$  and potential deployment locations  $PLSet$ .
5. The QP formulation for all kinds of locations are constructed based on  $PLSet$ ,  $\{\beta_1, \beta_2, \dots, \beta_C\}$  and  $\alpha_1, \alpha_2, \dots, \alpha_C$ .

In the proposed QP-based approach, the utility of an established ad hoc network (i.e.,  $ANet$ ) can be calculated as Equation 6.7

$$Utility(ANet) = \sum_{LSet_c \in CSet} \lambda_c Num(Loc_i \in LSet_c), \quad (6.7)$$

where  $Utility(ANet)$  is the utility of the established ad hoc network;  $LSet_c$  is the set of the  $c^{th}$  kind of locations in the disaster environment;  $\lambda_c$  is the contribution value of the  $c^{th}$  kind of location to the utility of the established ad hoc network; and  $Num(Loc_i \in LSet_c)$  is the number of the  $c^{th}$  kind of locations covered by the established ad hoc network, which can be calculated based on  $PLSet$ ,  $\beta_c$  and  $\alpha_c$ .

The QP formulation for the proposed QP-based approach can be described as follows

$$\max \sum_{LSet_c \in CSet} \lambda_c \left( \sum_{k=1}^L (x_k \sum_{i=1}^M b_{i,k,c}) - \sum_{k_1=1}^L \sum_{k_2=1}^L x_{k_1} x_{k_2} a_{k_1,k_2,c} \right) \quad (6.8a)$$

$$\text{s.t.} \quad \sum_{k=1}^L x_k = N \quad (6.8b)$$

$$x_k \in \{0, 1\} \quad 1 \leq k \leq L \quad (6.8c)$$

$$\sum_{k=1}^L b_{i,k,c} x_k \geq 0 \quad Loc_i \in LSet_c \in CSet \quad (6.8d)$$

Equation 6.8 is explained as follows. The objective function is to choose  $N$  number of deployment locations from all potential deployment locations (i.e.,  $PLSet$ ) so as to maximise the utility of the established ad hoc network, which is calculated from the contribution values of all kinds of locations and the numbers of there kinds of locations covered by the established ad hoc network (i.e., Equation 6.8a). The number of chosen deployment locations should equal to  $N$  (i.e., Equation 6.8b). In addition, the value of each  $x_k$  in  $X$  should be a binary value, which is either 0 or 1 (i.e., Equation 6.8c). Finally, each location in  $CSet$  may be or may not be covered by WRs (i.e., Equation 6.8d).



## 6.5 Experiments and Analysis

Two experiments are conducted in Matlab2014a [52] to evaluate the performance of the LP-based approach (represented by LP) and the QP-based approach for the MILCP (represented by QP), respectively.

**Experiment 1** is to evaluate the optimisation of the deployment locations created by LP and QP. The benchmark approach of Experiment 1 is the greedy algorithm-based replay deployment approach proposed by Guo et al. [56] (represented by GA).

**Experiment 2** is to compare computational complexities of LP and QP.

### 6.5.1 The experimental settings

In two experiments, 20 ILs are randomly distributed in a  $25 \times 25$  disaster environment and a number of WRs are going to be deployed to the environment. The sensing distances of all WR are 5. The distribution of ILs in the environment is illustrated in Figure 6.5.

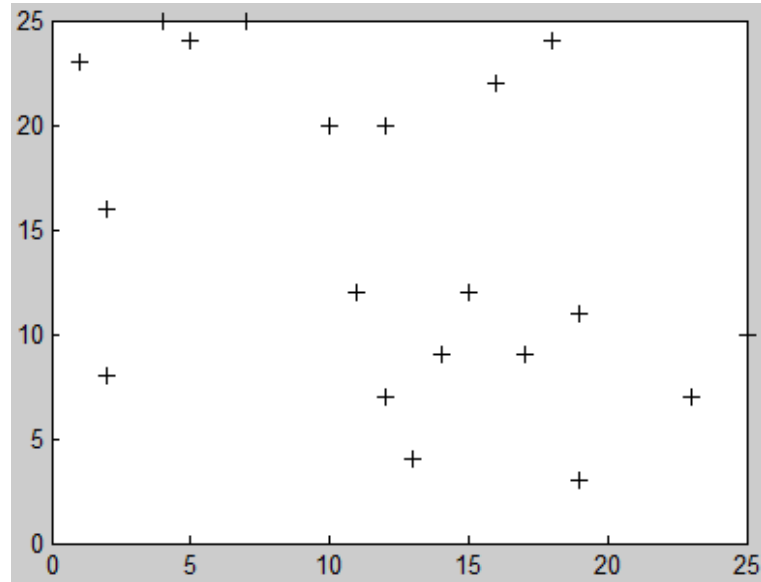


Figure 6.5: The distribution of ILs in the environment

In Figure 6.5, the crosses represent the locations of ILs in the environment. Base on the LP formulation of the MWRDP, the minimum number of WRs that can cover all ILs in the environment is 6. The settings of two experiments are shown in Table 6.1.

Table 6.1: The experimental settings

Parameters	Values
Size of the disaster environment	$25 \times 25$
Number of ILs	20
Number of WRs	1-5
The sensing distance of WRs	5

In Experiment 1, GA, LP and QP are employed to create deployment locations for WRs in the environment, respectively, where there are 1 to 5 WRs (i.e.,  $N$ ) in the environment. The maximum numbers of ILs covered by all WRs in the network are used to be the indicator of Experiment 1.

In Experiment 2, LP and QP are employed to create deployment locations for WRs in the environment, respectively, where there are 1 to 5 WRs (i.e.,  $N$ ) in the environment. The number of computations to create the deployment locations for WRs are used to be the indicator of Experiment 2.

### 6.5.2 The results and analysis of Experiment 1

The results of Experiment 1 about the maximum number of ILs covered by the ad hoc network are illustrated in Figure 6.6.

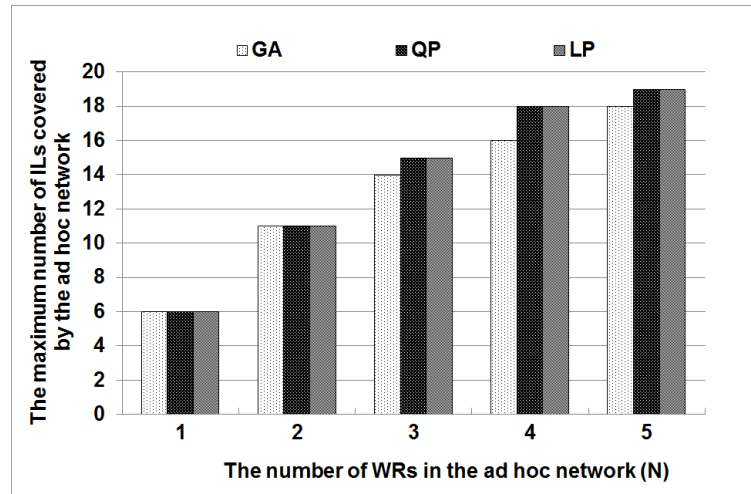


Figure 6.6: The maximum number of ILs covered by WRs

In Figure 6.6, the X-axis represents the number of WRs (i.e.,  $N$ ) in the environment, while the Y-axis represents the number of ILs covered by WRs in the environment. From

Figure 6.6, it can be seen that the numbers of ILs covered by the ad hoc network established by LP and QP are always the same. We can say that QP can create the same deployment locations for WRs as LP. When there is only 1 WR in the environment (i.e.,  $N = 1$ ), the maximum numbers of ILs covered by the ad hoc network established by GA, LP and QP are same. This is because that the GA aims to maximise the number of ILs covered by the new deployed WR (i.e., the only WR). LP and QP also aim to maximise the number of ILs covered by all WRs in the environment (i.e., the only WR) so that when  $N = 1$ , the objectives of the GA, LP and QP are the same. However, with the number of WRs in the environment increasing, the drawback of the GA begins to appear, because it can only maximise the number of ILs covered by the new deployed WR rather than all WRs in the established ad hoc network. Different from GA, LP and QP create deployment locations for all WRs in the environment in one time so that LP and QP can create the suitable deployment locations for WRs to maximise the number of ILs covered by all WRs in the environment. Therefore, when there are 4 WRs in the environment (i.e.,  $N = 4$ ), the deployment locations of WRs created by GA can only cover 16 ILs, while the deployment locations of WRs created by LP and QP can cover 18 ILs. The difference between the numbers of ILs covered by WRs deployed by the greedy algorithm-based approach (GA) and programming based approaches (LP and QP) is 2, which is 10% of all ILs in the environment. The number of ILs in this experiment is small, if the greedy algorithm-based approach (GA) and programming based approaches (LP and QP) are employed to deployed WRs in a large disaster environment with thousands of ILs, the difference between the maximum numbers of ILs covered by WRs deployed based on the greedy algorithm-based approach (GA) and programming based approaches (LP and QP) could be very big.

### 6.5.3 The results and analysis of Experiment 2

From the environmental setting shown in Table 6.1, there are 20 ILs in the environment (i.e.,  $M = 20$ ). Based on the finding of potential deployment locations (i.e., see Subsection 6.3.1), 90 number of potential deployment locations are found in the environment (i.e.,  $L = 90$ ), which is much less than the number of deployment locations in the environment ( $25 \times 25 = 625$ ). The results of Experiment 2 are illustrated in Figure 6.7.

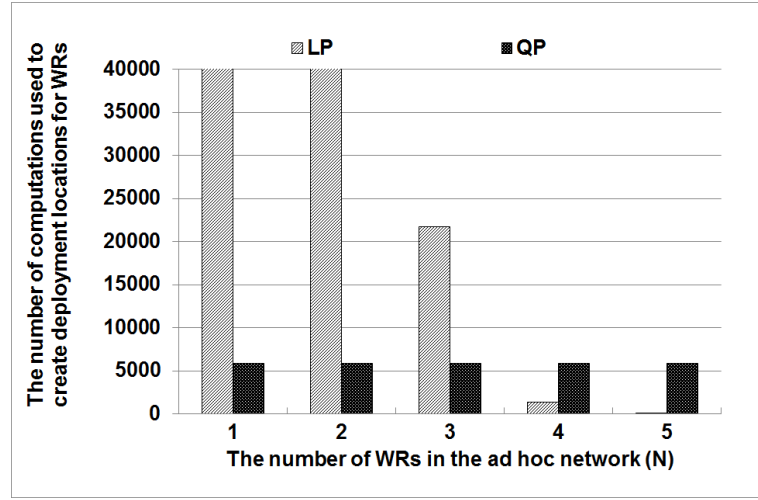


Figure 6.7: The number of computations used by LP and QP

In Figure 6.7, the X-axis represents the number of WRs (i.e.,  $N$ ) in the environment, while the Y-axis represents the number of computation used to create deployment locations for WRs. From Figure 6.7, it can be seen that the number of computations used by LP is unstable, which is related to the number of WRs in the environment (i.e.,  $N$ ). With the increase of  $N$ , the number of computations used by LP decreased sharply, which is 1026875, 431909, 21699, 1350 and 20, when  $N$  is 1, 2, 3, 4, and 5, respectively (i.e., for clearness, the maximum of Y-axis is only 40000). Different from LP, the number of computations used by QP is stable. This is because the most of computation used by QP is to construct the  $L \times L$  upper triangle matrix  $\alpha$  (i.e., see Subsection 6.1.1), which is only related with the number of ILs (i.e.,  $M$ ) and the number of potential deployment locations (i.e.,  $L$ ) in the environment rather than the number of WRs (i.e.,  $N$ ). The number of computations used by QP to construct  $\alpha$  are constant, which is  $\frac{L(2M+L+1)}{2} = 45 \times 131 = 5895$ , which is less than the number of computations used by LP in some of situations.

#### 6.5.4 Findings from experiments

The results of Experiments 1 and 2 have shown that the LP-based approach and the QP-based approach have the following three features.

1. By removing the assumption of the MWRDP, the LP-based approach can deploy the limited number of WRs to solve the MILCP in disaster environments (to achieve **Objective 8**, see Section 1.3);.
2. Based on the LP-based approach, the deployed WRs with limited sensing and communication capabilities can cover the maximum ILs and areas in a disaster environment

(to achieve **Objectives 2 and 6**, see Section 1.3);

3. The QP-based approach can create the same deployment locations for WRs as the LP-based approach with less computational complexity.

## 6.6 Summary

In this chapter, two mathematical programming-based WR deployment approaches were proposed for the ad hoc network establishment. First, the problem description and definitions of WR deployment problem were given. Then, the LP-based WR deployment approach and the QP-based WR deployment approach for ad hoc network establishment in disaster environments were introduced in detail. Finally, experiments to evaluate the performance of the mathematical programming-based WR deployment approaches were demonstrated and the results were analysed. Theoretically, the coverage of important locations in the proposed approaches could reach 25% more than that of in greedy algorithm-based approaches in some disaster environments.

## Conclusion and Future Work

In this thesis, the challenging issues of task-based resource management in disaster environments were investigated. In order to solve these challenging issues to achieve efficient task allocation in disaster environments, four multi-agent approaches were proposed in the thesis. This chapter concludes the thesis and outlines future directions of the research.

### 7.1 Contributions of the Thesis

This thesis aims at (1) investigating challenging issues of task-based resource management in disaster environments and (2) developing multi-agent approaches to achieve efficient task-based resource management in disaster environments. The contributions of this thesis include:

- **A coordinated approach for weighted task allocation**

A coordinated approach was proposed for weighted task allocation in disaster environments, which enables agents with limited capabilities to achieve efficient task allocation under communication constraints by considering different urgent degrees of tasks. In the proposed approach, firstly, *the group formation mechanism* enables agents to make use of their limited communication capabilities to form groups and select coordinators to collect information for task allocation under communication constraints. Then, *the token passing mechanism* enables coordinators to efficiently collect information for task allocation from their group members. Based on the collected information, *the utility calculation mechanism* enables coordinators to create suitable task allocation solutions for their group members by considering time and space constraints as well as different urgent degrees of tasks in disaster environments. Experimental results showed the advantages of the proposed approach in terms of information collection under communication constraints and weighted task allocation in disaster environments.

- **A dynamic task allocation approach for heterogeneous agents by group formation**

A dynamic task allocation approach was proposed for heterogeneous agents in disaster environments, which enables agents with different capabilities to achieve dynamic task allocation in disaster environments under communication constraints. In the proposed approach, *the information collection mechanism* enables agents with limited communication capabilities to prune their connections in communication networks and select network leaders to collect information for task allocation under communication constraints. Based on the collected information, *the group task allocation mechanism* enables network leaders to allocate tasks and agents into different groups based on locations of tasks and capabilities of agents so as to facilitate agents to cooperatively perform tasks. In addition, *the group coordination mechanism* enables different groups of agents to dynamically coordinate at assembly points so as to suit changes of open and dynamic disaster environments under communication constraints. Experimental results demonstrated that the proposed approach had better performance than many existing approaches in terms of information collection and dynamic task allocation in disaster environments under space and communication constraints.

- **A wireless mobile robot (WR) search and deployment approach for ad hoc network establishment**

A WR search and deployment approach was proposed for ad hoc network establishment in disaster environment, which can help WRs with limited energy and capabilities to efficiently search and deploy them to cooperatively establish ad hoc networks in unknown and complex disaster environments so as to improve task allocation of first responders in such environments. In the proposed approach, *the search process* enables WRs without prior knowledge to efficiently search important locations (ILs) (i.e., locations of tasks) in disaster environments. In order to avoid obstacles during the search, the search process employs the A\* search algorithm to create pathes for WRs. In addition, *the deployment process* enables WRs to timely find suitable deployment locations based on the information collected in the search process. The ad hoc networks established by the proposed approach can achieve three following objectives.

1. **The communication of WRs:** WRs are communicable in the established ad hoc network so as to improve the work efficiency of WRs on task allocation in the network;

2. **The maximum coverage of ILs:** WRs can cover as many ILs in a disaster environment as possible so as to ensure tasks at these ILs can be performed with the guidance of WRs; and
3. **The maximum coverage of areas:** WRs can cover as many areas in a disaster environment as possible so as to have more opportunities to guide first responders.

Experimental results showed that ad hoc networks established by the proposed approach can greatly improve the task allocation of first responders in disaster environments.

- **Two mathematical programming-based WR deployment approaches for disaster rescues**

Two WR deployment approaches based on mathematical programming were proposed for ad hoc network establishment in disaster environments. The proposed approaches can deploy a limited number of WRs to cover the maximum number of ILs and areas in disaster environments so as to improve task allocation of first responders in such environments. *The linear programming (LP)-based approach* deploys the limited number of WRs to cover the maximum ILs and areas in disaster environments by considering limited sensing and communication capabilities of WRs. *The quadratic programming (QP)-based approach* can create the same deployment locations for WRs as the LP-based approach with less computational complexity. Experimental results showed the advantages of the proposed approaches in terms of the IL and area coverage of WRs in disaster environments, especially, when WRs are not sufficient.

## 7.2 Future Work

Although the proposed approaches in this thesis can overcome challenging issues to achieve efficient task-based resource management in disaster environments, there are still some limitations that need to be improved in the future.

- **Further experiments**

The experiments in Chapters 3, 4, 5 and 6 have shown the advantages of the proposed approaches on the task allocation and the establishment of ad hoc networks. However, further experiments are needed to comprehensively evaluate the effectiveness, correctness and scalability of the proposed approaches. In addition, since the situations



of disaster environments are various, multiple experiments in different environmental settings are also needed to be performed in the future to verify the robustness and generality of the proposed approaches.

- **The combination of four proposed approaches**

The approaches in Chapters 3 and 4 can help agents to allocate tasks in disaster environments, while the approaches in Chapters 5 and 6 can help agents to establish ad hoc networks in disaster environments. Although the four proposed approaches have different functionalities, the aim of them is the same, which is to help agents to achieve the efficient resource allocation in disaster environments. Therefore, it is interesting to combine the four proposed approaches together in the future, which means that in a disaster environment, first, some agents establish ad hoc network according to approaches in Chapters 5 and 6. Then, these agents can employ the approaches in Chapters 3 and 4 to efficiently allocate other agents to suitable tasks in the environment.

- **The dynamic task allocation approach for heterogeneous agents**

The group task allocation mechanism of the approach proposed in **Chapter 4** can allocate agents to suitable groups based on their capabilities and tasks in the groups. This allocation is myopic, where allocated agents are only suitable to perform currently existing tasks in groups. Due to the dynamic feature of disaster environments, tasks in groups could continuously change. Therefore, in the future, it is important to incorporate learning or prediction mechanisms to the approach so as to enable the group task allocation mechanism to make long-term decisions on group task allocation.

- **The WR search and deployment approach**

Disaster environments are open and highly dynamic environments so that important locations (ILs) (i.e., locations of tasks) are continuously changing in such environments [133], [21]. Although the WR search and deployment approach proposed in **Chapter 5** can quickly find deployment locations for WRs by achieving the three objectives, without suitable adaptive approaches, currently, WRs in the established ad hoc network cannot dynamically change their deployment locations to suit the change of ILs in disaster environments. In the future, an adaptive approach will be added to the current approach so as to let WRs to dynamically adapt their deployment locations based on ILs in disaster environments.

- **Mathematical programming-based WR deployment approaches**

Although two mathematical programming-based approaches proposed in **Chapter 6**

---

enable WRs to cover the maximum ILs and areas in disaster environments, without the communication of WRs, the work efficiency of WRs in the established ad hoc network is low. One of the future directions is to extend the current two approaches to enable WRs in the established ad hoc network to be communicable so as to overcome the limitation of the approaches and improve the work efficiency of the established ad hoc network.

# Bibliography

---

- [1] A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *Proceedings of the 33rd Hawaii International Conference on System Sciences*, volume 6 of *HICSS '00*, pages 6007–6015, Maui, Hawaii, 2000.
- [2] R. Addis, G. Fraser, F. Girardi, G. Graziani, Y. Inoue, N. Kelly, W. Klug, A. Kulmala, K. Nodop, and J. Pretel. Etex: A european tracer experiment; observations, dispersion modelling and emergency response. *Atmospheric Environment*, 32(24):4089–4094, 1998.
- [3] J. Allen, N. Blaylock, and G. Ferguson. A problem solving model for collaborative agents. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS'02, pages 774–781, Bologna, Italy, 2002.
- [4] N. Altshiller-Court. *College Geometry: An Introduction to the Modern Geometry of the Triangle and the Circle*. Dover Publications, 2007.
- [5] F. Aurenhammer. Voronoi diagrams: a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405, 1991.
- [6] N. Avouris, A. Dimitracopoulou, V. Komis, and C. Fidas. Ocaf: An object-oriented model of analysis of collaborative problem solving. In *Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community*, CSCL '02, pages 92–101, Boulder, Colorado, 2002.
- [7] S. Balakirsky, S. Carpin, A. Kleiner, M. Lewis, A. Visser, J. Wang, and V. A. Ziparo. Towards heterogeneous robot teams for disaster mitigation: Results and performance metrics from robocup rescue. *Journal of Field Robotics*, 24(11):943–967, 2007.
- [8] J. Q. Bao and W. C. Lee. Rapid deployment of wireless ad hoc backbone networks for public safety incident management. In *Proceedings of the IEEE Global Telecommunications Conference*, GLOBECOM'07, pages 1217–1221, Washington, DC, 2007.

- 
- [9] L. Barbulescu, Z. B. Rubinstein, S. F. Smith, and T. L. Zimmerman. Distributed coordination of mobile agent teams: The advantage of planning ahead. In *Proceedings of the 9th International Joint Conference on Autonomous Agents and Multiagent Systems*, volume 1 of *AAMAS'10*, pages 1331–1338, Toronto, Canada, 2010.
- [10] D. Bedford and L. Faust. Role of online communities in recent responses to disasters: Tsunami, china, katrina, and haiti. *Proceedings of the American Society for Information Science and Technology*, 47(1):1–3, 2010.
- [11] J. Bellingham, M. Tillerson, A. Richards, and J. P. How. Multi-task allocation and path planning for cooperating uavs. In *Cooperative Control: Models, Applications and Algorithms*, pages 23–41. Springer, 2003.
- [12] R. Bellman and L. Zadeh. Decision-making in a fuzzy environment. *Management Science*, 17(4):141–164, 1970.
- [13] K. Bennett and E. Parrado-Hernández. The interplay of optimization and machine learning research. *Journal of Machine Learning Research*, 7:1265–1281, 2006.
- [14] M. N. Bilbao, S. Gil-López, J. Del Ser, S. Salcedo-Sanz, M. Sánchez-Ponte, and A. Arana-Castro. Novel hybrid heuristics for an extension of the dynamic relay deployment problem over disaster areas. *TOP*, 22(3):997–1016, 2014.
- [15] P. Biswas and Y. Ye. Semidefinite programming for ad hoc wireless sensor network localization. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, IPSN '04, pages 46–54, Berkeley, CA, 2004.
- [16] S. C. Botelho and R. Alami. M+: A scheme for multi-robot cooperation through negotiated task allocation and achievement. In *Proceedings of the 7th IEEE International Conference on Robotics and Automation*, volume 2, pages 1234–1239, Detroit, MI, 1999.
- [17] H. Breu, J. Gil, D. Kirkpatrick, and M. Werman. Linear time euclidean distance transform algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):529–533, 1995.
- [18] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of the 4th annual ACM/IEEE international Conference on Mobile Computing and Networking*, pages 85–97, Dallas, TX, 1998.

- 
- [19] R. Buyya, D. Abramson, and J. Giddy. Nimrod/g: An architecture for a resource management and scheduling system in a global computational grid. In *Proceedings of the 4th International Conference/Exhibition on High Performance Computing in the Asia-Pacific Region*, volume 1, pages 283–289, Beijing, China, 2000.
- [20] S. Chang-Chien, W. L. Hung, and M. S. Yang. On mean shift-based clustering for circular data. *Soft Computing*, 16(6):1043–1060, 2012.
- [21] A. C. Chapman, R. A. Micillo, R. Kota, and N. R. Jennings. Decentralised dynamic task allocation: A practical game: Theoretic approach. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems*, volume 2 of AAMAS’09, pages 915–922, Budapest, Hungary, 2009.
- [22] P. Chen, J. Chang, T. Liang, C. Shieh, and Y. Zhuang. A multi-layer resource re-configuration framework for grid computing. In *Proceedings of the 4th international workshop on Middleware for grid computing*, MCG ’06, pages 13–18, Melbourne, Australia, 2006.
- [23] R. Chen, J. Coles, J. Lee, and H. R. Rao. Emergency communication and system design: The case of indian ocean tsunami. In *Proceedings of the 3rd International Conference on Information and Communication Technologies and Development*, pages 300–309, Doha, Qatar, 2009.
- [24] S. H. Clearwater. *Market-based Control: A Paradigm for Distributed Resource Allocation*. World Scientific, 1996.
- [25] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.
- [26] D. P. Coppola. *Introduction to international disaster management*. Butterworth-Heinemann, 2006.
- [27] F. C. Cuny. Introduction to disaster management: Lesson 5xtechnologies of disaster management. *Prehospital and Disaster Medicine*, 8(04):372–374, 1993.
- [28] R. I. da Silva, V. Del Duca Almeida, A. M. Poersch, and J. M. S. Nogueira. Wireless sensor network for disaster management. In *Proceedings of the IEEE Network Operations and Management Symposium*, pages 870–873, Osaka, Japan, 2010.

- 
- [29] P. Danielsson. Euclidean distance mapping. *Computer Graphics and Image Processing*, 14(3):227–248, 1980.
- [30] H. Deitel, P. Deitel, and T. Nieto. *e-Business & e-Commerce How to Program*. Prentice Hall, 2001.
- [31] D. Delling, P. Sanders, D. Schultes, and D. Wagner. Engineering route planning algorithms. In *Algorithmics of Large and Complex Networks*, pages 117–139. Springer, 2009.
- [32] N. Deshpande, E. Grant, and T. C. Henderson. Target-directed navigation using wireless sensor networks and implicit surface interpolation. In *Proceedings of the 2012 IEEE International Conference on Robotics and Automation*, pages 457–462, Saint Paul, MN, 2012.
- [33] A. V. Devereaux, J. R. Dichter, M. D. Christian, N. N. Dubler, C. E. Sandrock, J. L. Hick, T. Powell, J. A. Geiling, D. E. Amundson, and T. E. Baudendistel. Definitive care for the critically ill during a disaster: A framework for allocation of scarce resources in mass critical care: From a task force for mass critical care summit meeting, January 26-27, 2007, Chicago, IL. *CHEST Journal*, 133:51S–66S, 2008.
- [34] N. Dimakis, A. Filippoupolitis, and E. Gelenbe. Distributed building evacuation simulator for smart emergency management. *The Computer Journal*, 53(9):1384–1400, 2010.
- [35] J. R. Elliott and J. Pais. Race, class, and hurricane katrina: Social differences in human responses to disaster. *Social Science Research*, 35(2):295–321, 2006.
- [36] J. F. Engelberger. *Robotics in Practice: Management and Applications of Industrial Robots*. Springer Science & Business Media, 2012.
- [37] H. R. Faragardi, R. Shojaei, M. A. Keshtkar, and H. Tabani. Optimal task allocation for maximizing reliability in distributed real-time systems. In *Proceedings of the 12th IEEE/ACIS International Conference on Computer and Information Science*, pages 513–519, Niigata, Japan, 2013.
- [38] A. Farinelli, A. Rogers, A. Petcu, and N. R. Jennings. Decentralised coordination of low-power embedded devices using the max-sum algorithm. In *Proceedings of the 7th international Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS’08*, pages 639–646, Estoril, Portugal, 2008.

- 
- [39] J. Ferber. *Multi-Agent System: An Introduction to Distributed Artificial Intelligence*. Addison-Wesley, 1999.
- [40] A. Filippoupolitis and E. Gelenbe. A distributed decision support system for building evacuation. In *Proceedings of the 2nd Conference on Human System Interactions*, pages 320–327, Catania, Italy, 2009.
- [41] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *The International Journal of High Performance Computing Applications*, 15(3):200–222, 2001.
- [42] H. Gao, G. Barbier, and R. Goolsby. Harnessing the crowdsourcing power of social media for disaster relief. *IEEE Intelligent Systems*, (3):10–14, 2011.
- [43] M. E. Gaston and M. DesJardins. Agent-organized networks for dynamic team formation. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 230–237, Utrecht, Netherlands, 2005.
- [44] M. E. Gaston and M. DesJardins. Agent-organized networks for multi-agent production and exchange. In *Proceedings of the 20th National Conference on Artificial intelligence*, volume 1 of AAAI’05, pages 77–82, Pittsburgh, PA, 2005.
- [45] B. P. Gerkey and M. J. Mataric. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research*, 23(9):939–954, 2004.
- [46] R. Grinton, K. Sycara, and P. Scerri. Agent organized networks redux. In *Proceedings of the 23rd National Conference on Artificial intelligence*, volume 1 of AAAI’08, pages 83–88, Chicago, IL, 2008.
- [47] B. L. Golden, S. Raghavan, and E. A. Wasil. *The vehicle routing problem: Latest advances and new challenges: Latest advances and new challenges*, volume 43. Springer Science & Business Media, 2008.
- [48] M. Gombolay, R. Wilcox, and J. A. Shah. Fast scheduling of multi-robot teams with temporospatial constraints. In *Proceedings of Robotics: Science and Systems*, pages 1–8, Seattle, WA, 2013.
- [49] N. Görmez, M. Köksalan, and F. S. Salman. Locating disaster response facilities in istanbul. *Journal of the Operational Research Society*, 62(7):1239–1252, 2011.

- 
- [50] J. C. Gower. Euclidean distance geometry. *Mathematical Scientist*, 7(1):1–14, 1982.
- [51] J. C. Gower. Properties of euclidean and non-euclidean distance matrices. *Linear Algebra and its Applications*, 67:81–97, 1985.
- [52] M. Grant, S. Boyd, and Y. Ye. Cvx: Matlab software for disciplined convex programming, 2008.
- [53] Y. Guan, W. Xiao, R. K. Cheung, and C. Li. A multiprocessor task scheduling model for berth allocation: heuristic and worst-case analysis. *Operations Research Letters*, 30(5):343–350, 2002.
- [54] Z. Guessoum, M. Ziane, and N. Faci. Monitoring and organizational-level adaptation of multi-agent systems. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '04, pages 514–521, New York, NY, 2004.
- [55] W. Guo and X. Huang. On coverage and capacity for disaster area wireless networks using mobile relays. *EURASIP Journal on Wireless Communications and Networking*, 2009:38:1–38:17, 2009.
- [56] W. Guo, X. Huang, and Y. Liu. Dynamic relay deployment for disaster area wireless networks. *Wireless Communications and Mobile Computing*, 10(9):1238–1252, 2010.
- [57] D. Hanselman and B. C. Littlefield. *Mastering MATLAB 5: A comprehensive tutorial and reference*. Prentice Hall PTR, 1997.
- [58] G. H. Harel and S. S. Tzafrir. The effect of human resource management practices on the perceptions of organizational and market performance of the firm. *Human Resource Management*, 38(3):185–199, 1999.
- [59] L. C. Harris and E. Ogbonna. Strategic human resource management, market orientation, and organizational performance. *Journal of Business Research*, 51(2):157–166, 2001.
- [60] S. Hassard, A. Blandford, and A. Cox. Analogies in design decision-making. In *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology*, BCS-HCI '09, pages 140–148, Swinton, UK., 2009.



- 
- [61] N. Heo and P. K. Varshney. A distributed self spreading algorithm for mobile wireless sensor networks. In *Proceedings of the 2003 IEEE Wireless Communications and Networking Conference*, volume 3, pages 1597–1602, New Orleans, LA, 2003.
- [62] D. S. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and vlsi. *Journal of Association for Computing Machinery*, 32(1):130–136, 1985.
- [63] S. Hua, Y. Guo, Y. Liu, H. Liu, and S. S. Panwar. Scalable video multicast in hybrid 3g/ad-hoc networks. *IEEE Transactions on Multimedia*, 13(2):402–413, 2011.
- [64] H. Huang and S. Chung. Dynamic visibility graph for path planning. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004*, volume 3 of *IROS'04*, pages 2813–2818, Sendai, Japan, 2004.
- [65] M. A. Huselid. The impact of human resource management practices on turnover, productivity, and corporate financial performance. *Academy of Management Journal*, 38(3):635–672, 1995.
- [66] K. Islam. An e-market framework to determine the strength of business relationships between intelligent agents. In *Proceedings of the 6th Australasian Conference on Data Mining and Analytics*, volume 70 of *AusDM '07*, pages 215–224, Darlinghurst, Australia, 2007. .
- [67] N. R. Jennings. Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. *Artificial Intelligence*, 75:195–240, 1995.
- [68] N. R. Jennings and M. Wooldridge. On agent-based software engineering. *Artificial Intelligence*, 117:277–296, 2000.
- [69] J. Jiang, Y. Lai, and F. Deng. Mobile robot coordination and navigation with directional antennas in positionless wireless sensor networks. *International Journal of Ad Hoc and Ubiquitous Computing*, 7(4):272–280, 2011.
- [70] A. Jotshi, Q. Gong, and R. Batta. Dispatching and routing of emergency vehicles in disaster mitigation using data fusion. *Socio-Economic Planning Sciences*, 43(1):1–24, 2009.
- [71] N. Kalra and A. Martinoli. Comparative study of market-based and threshold-based task allocation. In *Distributed Autonomous Robotic Systems 7*, pages 91–101. Springer, 2006.

- 
- [72] K. Kanchanasut, T. Wongsardsakul, M. Chansutthirangkool, A. Laouiti, H. Tazaki, and K. R. Arefin. Dumbo ii: A v-2-i emergency network. In *Proceedings of the 4th Asian Conference on Internet Engineering*, pages 37–38, Pattaya, Thailand, 2008.
- [73] R. J. Kang and T. Müller. Sphere and dot product representations of graphs. In *Proceedings of the 27th Annual Symposium on Computational Geometry*, pages 308–314, Paris, France, 2011.
- [74] K. Kise, A. Sato, and M. Iwata. Segmentation of page images using the area voronoi diagram. *Computer Vision and Image Understanding*, 70(3):370–382, 1998.
- [75] H. Kitano. Robocup rescue: A grand challenge for multi-agent systems. In *Proceedings of the 4th International Conference on Multi-Agent Systems*, pages 5–12, Boston, MA, 2000.
- [76] H. Kitano, S. Tadokoro, I. Noda, H. Matsubara, T. Takahashi, A. Shinjou, and S. Shimada. Robocup rescue: Search and rescue in large-scale disasters as a domain for autonomous agents research. In *IEEE International Conference on System, Man and Cybernetics*, volume 6, pages 739–746, Tokyo, Japan, 1999.
- [77] T. Knabe, M. Schillo, and K. Fischer. Improvements to the fipa contract net protocol for performance increase and cascading applications. In *International Workshop for Multi-Agent Interoperability at the German Conference on AI*, 2002.
- [78] S. Koenig, C. Tovey, and W. Halliburton. Greedy mapping of terrain. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, volume 4, pages 3594–3599, Seoul, South Korea, 2001.
- [79] M. Koes, I. Nourbakhsh, and K. Sycara. Heterogeneous multirobot coordination with spatial and temporal constraints. In *Proceedings of the 20th National Conference on Artificial Intelligence*, volume 3, pages 1292–1297, Pittsburgh, PA, 2005.
- [80] M. Koes, K. Sycara, and I. Nourbakhsh. A constraint optimization framework for fractured robot teams. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '06, pages 491–493, Hakodate, Japan, 2006.
- [81] B. Kohout. The darpa coordinators program: A retrospective. In *Proceedings of the 2011 International Conference on Collaboration Technologies and Systems*, pages 342–342, Philadelphia, PA, 2011.

- 
- [82] K. Konstantinides, R. T. Kaneshiro, and J. R. Tani. Task allocation and scheduling models for multiprocessor digital signal processing. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(12):2151–2161, 1990.
  - [83] R. Kontchakov, Y. Nenov, I. Pratt-Hartmann, and M. Zakharyashev. Topological logics with connectedness over euclidean spaces. *ACM Transactions on Computational Logic*, 14(2):13:1–13:48, 2013.
  - [84] V. Kordic. *Petri Net: Theory and Applications*. I-Tech Education and Publishing, Vienna, Austria, 2008.
  - [85] M. Krieger, J. Billeter, and L. Keller. Ant-like task allocation and recruitment in cooperative robots. *Nature*, 406(6799):992–995, 2000.
  - [86] Y. Kwok and I. Ahmad. Dynamic critical-path scheduling: An effective technique for allocating task graphs to multiprocessors. *IEEE Transactions on Parallel and Distributed Systems*, 7(5):506–521, 1996.
  - [87] Y. Kwok and I. Ahmad. Static scheduling algorithms for allocating directed task graphs to multiprocessors. *ACM Computing Surveys*, 31(4):406–471, 1999.
  - [88] J. Latombe. *Robot Motion Planning*, volume 124. Springer Science & Business Media, 1991.
  - [89] E. Lee and M. Tsuchiya. A task allocation model for distributed computing systems. *IEEE Transactions on Computers*, (1):41–47, 1982.
  - [90] S. Lele and J. T. Richtsmeier. Euclidean distance matrix analysis: A coordinate-free approach for comparing biological shapes using landmark data. *American Journal of Physical Anthropology*, 86(3):415–427, 1991.
  - [91] V. Lesser. Cooperative multiagent systems: A personal view of the state of the art. *IEEE Transactions on Knowledge and Data Engineering*, 11:133–142, 1999.
  - [92] M. Llorens and J. Oliver. Structural and dynamic changes in concurrent systems: Reconfigurable petri nets. *IEEE Transactions on Computers*, 53(9):1147–1158, 2004.
  - [93] K. Lorincz, D. J. Malan, T. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, M. Welsh, and S. Moulton. Sensor networks for emergency response: challenges and opportunities. *IEEE Pervasive Computing*, 3(4):16–23, 2004.

- 
- [94] W. Lu, W. K. G. Seah, E. W. C. Peh, and Y. Ge. Communications support for disaster recovery operations using hybrid mobile ad-hoc networks. In *Proceedings of the 32nd IEEE Conference on Local Computer Networks*, pages 763–770, Dublin, Ireland, 2007.
- [95] E. Luis, I. S. Dolinskaya, and K. R. Smilowitz. Disaster relief routing: Integrating research and practice. *Socio-economic Planning Sciences*, 46(1):88–97, 2012.
- [96] P. Mattila. *Geometry of sets and measures in Euclidean spaces: Fractals and rectifiability*. Number 44. Cambridge University Press, 1999.
- [97] P. J. May. Formulating disaster relief when needs are unknown. *Journal of Policy Analysis and Management*, 2(1):39–54, 1982.
- [98] J. Michie and M. Sheehan-Quinn. Labour market flexibility, human resource management and corporate performance. *British Journal of Management*, 12(4):287–306, 2001.
- [99] W. Mio, J. C. Bowers, and X. Liu. Shape of elastic strings in euclidean space. *International Journal of Computer Vision*, 82(1):96–112, 2009.
- [100] D. J. Musliner, E. H. Durfee, J. Wu, D. A. Dolgov, R. P. Goldman, and M. S. Boddy. Coordinated plan management using multiagent mdps. In *Proceedings of the AAAI Spring Symposium: Distributed Plan and Schedule Management*, pages 73–80, Stanford, CA, 2006.
- [101] R. Nair, T. Ito, M. Tambe, and S. Marsella. Task allocation in the robocup rescue simulation domain: A short note. In *RoboCup 2001: Robot Soccer World Cup V*, pages 751–754. Springer, 2002.
- [102] Y. Nakajima, H. Shiina, S. Yamane, T. Ishida, and H. Yamaki. Disaster evacuation guide: Using a massively multiagent server and gps mobile phones. In *Proceedings of the 2007 International Symposium on Applications and the Internet*, pages 2–8, Hiroshima, Japan, 2007.
- [103] D. Nakamura, N. Uchida, H. Asahi, K. Takahata, K. Hashimoto, and Y. Shibata. Wide area disaster information network and its resource management system. In *Proceedings of the 17th International Conference on Advanced Information Networking and Applications*, pages 146–149, Okinawa, Japan, 2003.

- 
- [104] N. C. Narendra, U. Bellur, S. K. Nandy, and K. Kalapriya. Functional and architectural adaptation in pervasive computing environments. In *Proceedings of the 3rd International Workshop on Middleware for Pervasive and Ad Hoc Computing*, MPAC '05, pages 1–7, Grenoble, France, 2005.
- [105] D. Niculescu and B. R. Badrinath. Ad hoc positioning system (aps) using aoa. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications*, volume 3, pages 1734–1743, San Francisco, CA, 2003.
- [106] N. J. Nilsson. A mobile automaton: An application of artificial intelligence techniques. Technical report, DTIC Document, 1969.
- [107] N. J. Nilsson. *Principles of Artificial Intelligence*. Morgan Kaufmann, 1982.
- [108] C. Pian and H. L. Haberer. Dynamic task allocation in a multi-processor system employing distributed control processors and distributed arithmetic processors, 1994.
- [109] K. L. Pokorny and R. E. Vincent. Multiple constraint satisfaction problems using the A-star (A\*) search algorithm: Classroom scheduling with preferences. *Journal of Computing Sciences in Colleges*, 28(5):152–159, 2013.
- [110] F. F. Pollitz, R. S. Stein, V. Sevilgen, and R. Bürgmann. The 11 April 2012 east Indian Ocean Earthquake triggered large aftershocks worldwide. *Nature*, 490(7419):250–253, 2012.
- [111] K. Ramamritham, J. A. Stankovic, and W. Zhao. Distributed scheduling of tasks with deadlines and resource requirements. *IEEE Transactions on Computers*, 38(8):1110–1123, 1989.
- [112] S. D. Ramchurn, A. Farinelli, K. S. Macarthur, and N. R. Jennings. Decentralized coordination in robocup rescue. *The Computer Journal*, 53(9):1447–1461, 2010.
- [113] S. D. Ramchurn, M. Polukarov, A. Farinelli, C. Truong, and N. R. Jennings. Coalition formation with spatial and temporal constraints. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, volume 3, pages 1181–1188, Toronto, Canada, 2010.
- [114] R. Rashid, A. Tevanian Jr, M. Young, D. Golub, R. Baron, D. Black, W. J. Bolosky, and J. Chew. Machine-independent virtual memory management for paged uniprocessor and multiprocessor architectures. *IEEE Transactions on Computers*, (8):896–908, 1988.

- 
- [115] S. Ratick, B. Meacham, and Y. Aoyama. Locating backup facilities to enhance supply chain disaster resilience. *Growth and Change*, 39(4):642–666, 2008.
- [116] J. Reich. Toward automatic reconfiguration of robot sensor networks for urban search and rescue. In *Proceedings of the 1st International Workshop on Agent Technology for Disaster Management*, pages 18–23, Hakodate, Japan, 2006.
- [117] J. Reich and E. Sklar. Robot-sensor networks for search and rescue. In *Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics*, pages 1–8, Kobe, Japan, 2005.
- [118] F. Ren, M. Zhang, and K. Sim. Adaptive conceding strategies for automated trading agents in dynamic, open markets. *Decision Support Systems*, 48(2):331–341, 2009.
- [119] A. Rogers, E. David, and N. R. Jennings. Self-organized routing for wireless microsensor networks. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 35(3):349–359, 2005.
- [120] D. Saha and A. Mukherjee. Pervasive computing: a paradigm for the 21st century. *IEEE Computer*, 36(3):25–31, 2003.
- [121] T. Saito and J. Toriwaki. New algorithms for euclidean distance transformation of an n-dimensional digitized picture with applications. *Pattern Recognition*, 27(11):1551–1565, 1994.
- [122] A. K. Salkintzis. Evolving public safety communication systems by integrating wlan and tetra networks. *IEEE Communications Magazine*, 44(1):38–46, 2006.
- [123] G. Salton, A. Wong, and C. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.
- [124] S. Sariel and H. L. Akin. A novel search strategy for autonomous search and rescue robots. In *RoboCup 2004: Robot Soccer World Cup VIII*, pages 459–466. Springer, 2005.
- [125] P. Scerri, A. Farinelli, S. Okamoto, and M. Tambe. Allocating tasks in extreme teams. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 727–734, Utrecht, Netherlands, 2005.
- [126] N. Schurr, P. Patil, F. Pighin, and M. Tambe. Lessons learned from disaster management. In *Proceedings of the 1st International Workshop on Agent Technology for Disaster Management*, pages 40–47, Hakodate, Japan, 2006.

- 
- [127] C. Sengul, A. C. Viana, and A. Ziviani. A survey of adaptive services to cope with dynamics in wireless self-organizing networks. *ACM Computing Surveys*, 44(4):23:1–23:35, 2012.
- [128] Z. Shao, Y. Liu, Y. Wu, and L. Shen. A rapid and reliable disaster emergency mobile communication system via aerial ad hoc bs networks. In *Proceedings of the 7th International Conference on Wireless Communications, Networking and Mobile Computing*, pages 1–4, Wuhan, China, 2011.
- [129] M. Sharmin, S. Ahmed, and S. Ahamed. An adaptive lightweight trust reliant secure resource discovery for pervasive computing environments. In *Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications*, pages 258–263, Pisa, Italy, 2006.
- [130] O. Shehory and S. Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101:165 – 200, 1998.
- [131] O. M. Shehory, K. Sycara, and S. Jha. Multi-agent coordination through coalition formation. In *Intelligent Agents IV Agent Theories, Architectures, and Languages*, pages 143–154. Springer, 1998.
- [132] R. G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113, 1980.
- [133] S. F. Smith, A. Gallagher, and T. Zimmerman. Distributed management of flexible times schedules. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS’07*, pages 1–8, Honolulu, Hawaii, 14-18 May 2007.
- [134] A. Srinivas, G. Zussman, and E. Modiano. Construction and maintenance of wireless mobile backbone networks. *IEEE/ACM Transactions on Networking*, 17(1):239–252, 2009.
- [135] D. M. Stocco. A star search: implications in controlling steroidogenesis. *Biology of Reproduction*, 56(2):328–336, 1997.
- [136] T. Sugawara, K. Fukuda, T. Hirotsu, and S. Kurihara. Effect of alternative distributed task allocation strategy based on local observations in contract net protocol. In *Principles and Practice of Multi-Agent Systems*, pages 90–104. Springer, 2012.

- [137] X. Sun, H. Yao, and S. Zhang. Robust object tracking via inertial potential based mean shift. In *Proceedings of the 3rd International Conference on Internet Multimedia Computing and Service*, ICIMCS '11, pages 178–181, Cheng Du, China, 2011.
- [138] S. Tadokoro, H. Kitano, T. Takahashi, I. Noda, H. Matsubara, A. Shinjoh, T. Koto, K. Takeuchi, F. Matsuno, and M. Hatayama. The robocup-rescue project: A robotic approach to the disaster mitigation problem. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 4, pages 4089–4094, San Francisco, CA, 2000.
- [139] J. Tang, B. Hao, and A. Sen. Relay node placement in large scale wireless sensor networks. *Computer Communications*, 29(4):490–501, 2006.
- [140] C. Theocharopoulou, I. Partsakoulakis, G. A. Vouros, and K. Stergiou. Overlay networks for task allocation and coordination in dynamic large-scale networks of cooperative agents. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, page 55, Honolulu, Hawaii, 2007.
- [141] K. Uno and K. Kashiya. Development of simulation system for the disaster evacuation based on multi-agent model using gis. *Tsinghua Science & Technology*, 13:348–353, 2008.
- [142] S. Venkataraman, W. Benger, A. Long, B. Jeong, and L. Renambot. Visualizing hurricane katrina: Large data management, rendering and display challenges. In *Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia*, pages 209–212, Kuala Lumpur, Malaysia, 2006.
- [143] W. E. Walsh and M. P. Wellman. Modeling supply chain formation in multiagent systems. In *Agent Mediated Electronic Commerce II*, pages 94–101. Springer, 2000.
- [144] C. Walters. Adaptive management of renewable resources. 1986.
- [145] J. Wang, M. Lewis, and P. Scerri. Cooperating robots for search and rescue. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 9–16, Hakodate, Japan, 2006.
- [146] C. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.
- [147] E. Welzl. Constructing the visibility graph for n-line segments in  $O(n^2)$  time. *Information Processing Letters*, 20(4):167–171, 1985.



- 
- [148] K. L. Wu and M. S. Yang. Mean shift-based clustering. *Pattern Recognition*, 40(11):3035–3052, 2007.
- [149] Y. Xie and W. Hung. Temperature-aware task allocation and scheduling for embedded multiprocessor systems-on-chip (mpsoc) design. *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology*, 45(3):177–189, 2006.
- [150] J. Yao, C. Lin, X. Xie, A. J. Wang, and C. Hung. Path planning for virtual human motion using improved A\* star algorithm. In *Proceedings of the 7th International Conference on Information Technology: New Generations*, pages 1154–1158, Las Vegas, NV, 2010.
- [151] J. W. Yoon, H. Lee, and H. Kim. Effective measurement selection in truncated kernel density estimator: Voronoi mean shift algorithm for truncated kernels. In *Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication*, ICUIMC '11, pages 1–7, Seoul, South Korea, 2011.
- [152] M. Zeleny. *Multiple criteria decision making*. McGraw-Hill series in quantitative methods for management. McGraw-Hill, 1982.
- [153] M. Zhang, J. Tang, and J. Fulcher. *Advanced Computational Intelligence Paradigms: Theory and Applications*, *Studies in Computational Intelligence (SCI)*, volume 115, chapter Agent Based Grid Computing, pages 439–483. Springer-Verlag, 2008.
- [154] X. Zhang, H. Gong, M. Liu, S. Lu, and J. Wu. Quantitative analysis of the effect of transmitting power on the capacity of wireless ad hoc networks. In *Proceedings of the 11th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '10, pages 231–240, Chicago, IL, 2010.
- [155] Y. Zhu, C. Xue, H. Cai, J. Yu, L. Ni, M. Li, and B. Li. On deploying relays for connected indoor sensor networks. *Journal of Communications and Networks*, 16(3):335–343, 2014.
- [156] G. Zussman and A. Segall. Energy efficient routing in ad hoc disaster recovery networks. *Ad Hoc Networks*, 1(4):405–421, 2003.