

# Scheduling the Refuelling Activities of Multiple Heterogeneous Autonomous Mobile Robots

R. Huisman

Master of Science Thesis



# **Scheduling the Refuelling Activities of Multiple Heterogeneous Autonomous Mobile Robots**

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Systems and Control at Delft  
University of Technology

R. Huisman

October 20, 2014

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of  
Technology



---

# Abstract

When a team of autonomous mobile robots (MRs) has to perform a mission of a duration that exceeds their energy capacities, the mission can only be fulfilled if the MRs are resupplied. Nowadays most MRs are powered by batteries, which can either be recharged, or replaced. Since it is not necessary that the MRs are battery powered, we will use the general term refuelling instead of recharging. In general there are two options for MRs to refuel themselves: by travelling to a fuelling station (FS) that is situated at a fixed location, or rendezvousing with a mobile fuelling station (MFS). When multiple MRs operate in the same environment, it is desired that the MRs share the available FSs instead of each MR having its own dedicated FS. Sharing the FSs will reduce the purchase and maintenance costs, and less space will be needed for the placement of FSs. The FSs considered during this thesis, can only refuel a single robot at a time. This raises the need for properly scheduling the refuelling activities, such that the FSs are shared in an efficient manner, and depletion of the MRs can be prevented.

This thesis presents several methods to schedule the refuelling activities of multiple heterogeneous autonomous MRs. The scheduling is focused on the selection of the refuel events, and the allocation of the FSs as a shared resource. The following problem is considered: For an environment which contains an arbitrary number of FSs, and MRs, the refuelling activities should be scheduled in such a way that the overall mission time is minimized. Each mission entails an assignment of a unique set of waypoints for each MR which have to be visited in a pre-determined order, in order to complete the mission. The total mission is accomplished when the last MR is completely refuelled, after visiting its last waypoint. Scheduling the refuelling activities using a time-based metric is complicated compared to a distance-based metric. Since a FS can refuel only a single MR at a time, the duration that each MR spends refuelling, and the ordering in which the MRs are refuelling have to be taken into account. Furthermore since the MRs share the FSs, each refuel event of one MR can affect all future refuel events of all other MRs.

Global optimal solutions for this problem can be found by using a centralized approach as shown in this thesis. However, since all refuel events of all MRs can influence each other, the complexity increases very quickly when the problem size increases. In order to obtain a global optimum, all possible refuel events have to be taken into account. Due to the computational complexity, the problem size that can be solved by this approach is limited. In order to solve

larger scale problems, a trade-off has been made between computation time and solution quality. A distributed, and hierarchical architecture are proposed, in order to distribute the computations and decision making over the robotic team. The core of the distributed approach is that MRs make individual decisions based on local knowledge. The decision making of the hierarchical approach is done for individuals or clusters of MRs. Simulation case studies indicate that these decentralized approaches can be used to solve large scale problems in real-time, at the cost of a suboptimal solution.

---

# Table of Contents

|                                                                                                  |            |
|--------------------------------------------------------------------------------------------------|------------|
| <b>List of Figures</b>                                                                           | <b>v</b>   |
| <b>List of Tables</b>                                                                            | <b>vii</b> |
| <b>Acknowledgements</b>                                                                          | <b>ix</b>  |
| <b>1 Introduction</b>                                                                            | <b>1</b>   |
| 1-1 Problem Statement . . . . .                                                                  | 2          |
| 1-2 Problem Analysis . . . . .                                                                   | 6          |
| 1-3 Outline . . . . .                                                                            | 8          |
| <b>2 Centralized Approach</b>                                                                    | <b>9</b>   |
| 2-1 Distance-Based Metric . . . . .                                                              | 11         |
| 2-1-1 MILP Formulation . . . . .                                                                 | 11         |
| 2-2 Time-Based Metric . . . . .                                                                  | 15         |
| 2-2-1 MILP Formulation . . . . .                                                                 | 15         |
| 2-2-2 Heuristic 1: Fixed Refuel Orderings . . . . .                                              | 22         |
| 2-2-3 Solving the Problem Using a Receding Horizon Principle . . . . .                           | 23         |
| 2-3 Simulation Results and Discussion . . . . .                                                  | 26         |
| 2-3-1 Simulation Setup . . . . .                                                                 | 26         |
| 2-3-2 Simulation 1: Patrol Mission . . . . .                                                     | 27         |
| 2-3-3 Simulation 2: Randomly Placed Waypoints . . . . .                                          | 28         |
| 2-3-4 Comparison Between Optimal Approach, Heuristic 1, and Receding Horizon Principle . . . . . | 29         |
| 2-4 Conclusion . . . . .                                                                         | 31         |

|          |                                                                                |           |
|----------|--------------------------------------------------------------------------------|-----------|
| <b>3</b> | <b>Distributed Approach</b>                                                    | <b>33</b> |
| 3-1      | Refuel Event Selection . . . . .                                               | 33        |
| 3-1-1    | Fixed Threshold . . . . .                                                      | 33        |
| 3-1-2    | Adaptive Threshold . . . . .                                                   | 34        |
| 3-2      | Fuelling Station Allocation . . . . .                                          | 35        |
| 3-3      | Mobile Fuelling Stations . . . . .                                             | 36        |
| 3-3-1    | Rendezvous Strategy . . . . .                                                  | 36        |
| 3-3-2    | Reformulation of the Distributed Approach . . . . .                            | 37        |
| 3-4      | Simulation Results and Discussion . . . . .                                    | 39        |
| 3-4-1    | Simulation Environment . . . . .                                               | 39        |
| 3-4-2    | Simulation Setup . . . . .                                                     | 40        |
| 3-5      | Conclusion . . . . .                                                           | 44        |
| <b>4</b> | <b>Hierarchical Approach</b>                                                   | <b>47</b> |
| 4-1      | Fuelling Station Allocation . . . . .                                          | 47        |
| 4-1-1    | Maximum Bipartite Matching . . . . .                                           | 48        |
| 4-1-2    | Queue Ordering . . . . .                                                       | 51        |
| 4-2      | Conclusion . . . . .                                                           | 54        |
| <b>5</b> | <b>Comparison between Centralized, Distributed and Hierarchical Approaches</b> | <b>55</b> |
| 5-1      | Simulation Results and Discussion . . . . .                                    | 56        |
| 5-2      | Conclusion . . . . .                                                           | 60        |
| <b>6</b> | <b>Conclusions</b>                                                             | <b>63</b> |
| 6-1      | Summary . . . . .                                                              | 63        |
| 6-2      | Future Work . . . . .                                                          | 65        |
|          | <b>Bibliography</b>                                                            | <b>67</b> |
|          | <b>Glossary</b>                                                                | <b>71</b> |
|          | List of Acronyms . . . . .                                                     | 71        |
|          | List of Symbols . . . . .                                                      | 72        |

---

## List of Figures

|     |                                                                                                                                                                                                                                                                                                                                     |    |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1-1 | Example of a mission where 4 MRs have to visit $N$ waypoints in a pre-determined order, and have the possibility of refuelling at a fixed, and mobile FS. . . . .                                                                                                                                                                   | 4  |
| 2-1 | Illustration of an optimal path for the heterogeneous vehicle. $\tau_i$ and $l_j$ denote the points where the fast vehicle will leave and enter the carrier respectively. The solid line shows the trajectory of the carrier and the dashed lines show the trajectory of the fast vehicle - Adapted from Klauco et al. [1]. . . . . | 10 |
| 2-2 | Scenario with 1 FS, and 2 MRs, which both have to visit 2 waypoints. . . . .                                                                                                                                                                                                                                                        | 16 |
| 2-3 | A scenario with 2 FSs, and 2MRs, which both have to visit 2 waypoints. . . . .                                                                                                                                                                                                                                                      | 21 |
| 2-4 | Scenario with 1 FS, and 1 MR which is located at an arbitrary initial position, and has to visit 2 waypoints. . . . .                                                                                                                                                                                                               | 25 |
| 2-5 | Patrol mission, environment with 2 FSs, and 3 MRs which all have to visit 2 waypoints, placed at a fixed distance from each other. . . . .                                                                                                                                                                                          | 27 |
| 3-1 | Total mission time given for a scenario with 2 FSs, and 6 MRs, which have to visit randomly placed waypoints, using a fixed threshold strategy. . . . .                                                                                                                                                                             | 41 |
| 3-2 | Total travel distance of all MRs given for a scenario with 2 FSs, and 6 MRs, which have to visit randomly placed waypoints, using a fixed threshold strategy. . . . .                                                                                                                                                               | 41 |
| 3-3 | Total travel distance of FS <sub>2</sub> given for a scenario with 2 FSs, and 6 MRs, which have to visit randomly placed waypoints, using a fixed threshold strategy. . . . .                                                                                                                                                       | 42 |
| 3-4 | Total mission time given for a scenario with 2 FSs, and 6 MRs, which have to visit randomly placed waypoints, using a fixed threshold strategy. The waiting times are taken into account. . . . .                                                                                                                                   | 42 |
| 3-5 | Simulation results for the adaptive threshold strategy, for a scenario with 2 FSs, and 6 MRs, which have to visit randomly placed waypoints. . . . .                                                                                                                                                                                | 43 |
| 3-6 | Comparison between the fixed, and adaptive threshold strategy, for a scenario with 2 FSs, and 6 MRs, which have to visit randomly placed waypoints. . . . .                                                                                                                                                                         | 44 |
| 4-1 | Bipartite graph with 5 applicants and 4 jobs, which have to be assigned to the applicants. . . . .                                                                                                                                                                                                                                  | 49 |
| 4-2 | Example of a scenario where 3 MRs are waiting in the queue of FS <sub>1</sub> . . . . .                                                                                                                                                                                                                                             | 53 |

|     |                                                                                                                                                        |    |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 5-1 | Comparison between distributed, and hierarchical approach, for scenarios with 2 FSs, and 6 MRs, which have to visit randomly placed waypoints. . . . . | 59 |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------|----|

---

## List of Tables

|      |                                                                                                                                                                       |    |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2-1  | Example of the set $\mathcal{Q}$ , which contains all possible combinations between FSs where a MR can start and end a fuel cycle, for a scenario with 3 FSs. . . . . | 13 |
| 2-2  | The set $\mathcal{Q}$ , which contains all combinations between FSs where a MR can start and end a fuel cycle, for a scenario with 2 FSs. . . . .                     | 14 |
| 2-3  | All possible refuel orderings for a scenario of 2 MRs which have to visit 2 waypoints in a pre-determined order. . . . .                                              | 16 |
| 2-4  | All refuel orderings taken into account when using Heuristic 1, for a scenario of 2 MRs which have to visit 2 waypoints in a pre-determined order. . . . .            | 23 |
| 2-5  | Average mission times of all centralized approaches, for the patrol mission. . . .                                                                                    | 27 |
| 2-6  | Comparison of the refuel ordering, chosen by the optimal approach, Heuristic 1, and the receding horizon principle. . . . .                                           | 28 |
| 2-7  | Comparison of the routes, found by the optimal approach, Heuristic 1, and the receding horizon principle. . . . .                                                     | 29 |
| 2-8  | Average mission times of all centralized approaches, for the mission with randomly placed waypoints. . . . .                                                          | 29 |
| 2-9  | Average mission times of all centralized approaches, calculated over the patrol mission, and mission with randomly placed waypoints. . . . .                          | 30 |
| 2-10 | Comparison of the computation times, between Heuristic 1, and the optimal approach, for a scenario with 2 FSs. . . . .                                                | 31 |
| 4-1  | Matrix that contains the total times of a refuel event for all combinations between $\{\text{MR}_1, \text{MR}_2\}$ , and $\{\text{FS}_1, \text{FS}_2\}$ . . . . .     | 50 |
| 4-2  | Matrix that contains the total times of a refuel event, for all combinations between common interest MRs, and all FSs. . . . .                                        | 51 |
| 4-3  | Values of $T_{p_{k,\text{next}}}$ , calculated for $\text{MR}_1$ , and $\text{MR}_2$ , for each possible refuel ordering. . . . .                                     | 53 |
| 4-4  | Values of $T_{p_{k,\text{next}}}$ , calculated for $\text{MR}_1$ , $\text{MR}_2$ , and $\text{MR}_3$ for the existing optimal schedule. . . . .                       | 54 |
| 5-1  | Average mission times of all approaches, for the patrol mission, and the mission 2 with randomly placed waypoints for each MR. . . . .                                | 56 |

|     |                                                                                                                                                                                               |    |
|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 5-2 | Mean errors of all approaches w.r.t. the optimal approach, calculated over the average mission times of the patrol mission, and mission with 2 randomly placed waypoints for each MR. . . . . | 57 |
| 5-3 | Average mission times for the mission with 7 randomly placed waypoints for each MR. . . . .                                                                                                   | 58 |
| 5-4 | Mean errors w.r.t. Heuristic 1, calculated over the average mission times of the mission with 7 randomly placed waypoints for each MR. . . . .                                                | 58 |
| 5-5 | Average mission times for the distributed, and hierarchical approaches, for scenarios with 2 FSs, and 6 MRs, which have to visit 10-50 randomly placed waypoints. . .                         | 60 |
| 5-6 | Comparison between scenarios with 2 fixed FSs, and 1 fixed and 1 mobile FS, for scenarios with 6 MRs, which have to visit 10-50 randomly placed waypoints. . . .                              | 60 |

---

# Acknowledgements

First of all I would like to thank my supervisor dr.ir. T. Keviczky for guiding me throughout the entire thesis project. Thanks for always taking time to discuss my updates, help if necessary, and providing helpful feedback which made me improve every time. I really enjoyed working with you.

Further I would like to thank dr.ir. C.J.M. Verhoeven for being my thesis advisor. Thanks for the ideas about swarm robotics, which helped me to think of methods that can really be beneficial in practice. I am curious what robotic swarms will bring in the future.

I also would like to thank dr.ir. T.J.J. van den Boom for taking his time to sit down with me to discuss max-plus algebra, max-plus linear systems, and how this could be useful during my thesis project.

My fellow student Paolo Forni, thank you for helping me with a combinatorial problem. I appreciate your help buddy.

Off course I want to thank my girlfriend Atousa for always motivating me, and her understanding. You were the best support.

At last but not least I would like to thank my parents for always believing in me, and their infinite support. I could not have done it without you.

Delft, University of Technology  
October 20, 2014

R. Huisman



“There are an endless number of things to discover about robotics. A lot of it is just too fantastic for people to believe.”

— *Daniel H. Wilson*



---

# Chapter 1

---

## Introduction

Since autonomous mobile robots (MRs) evolved, they have an increasing role in modern society. Autonomous MRs are well suited for missions in hazardous environments or environments where it is impossible for human beings to survive, e.g. toxic areas, deep sea, and space. The operation duration of MRs is limited by their energy capacity. With the evolvement of autonomous MRs, the need for self-sufficiency increased. MRs are self-sufficient if they are able to autonomously refuel themselves when necessary. Self-sufficient MRs can be deployed to fulfil all kinds of long-term missions, e.g. persistent labour, patrol missions, or area exploration. Self-sufficient MRs are already part of the society, some examples are: the museum tour guide ATLAS [2], the vacuum cleaning iRobot Roomba [3], and dog-like Sony Aibo [4].

The first MRs capable of autonomous refuelling, were the tortoises of W. Grey Walter built in 1950 [5]. Walter's tortoises were able to move themselves into a lighted recharging hutch. In 1960 J. Hopkins introduced the Hopkins beast, which was capable of plugging itself into a wall outlet. Steels [6] built a robotic ecosystem in 1994, where MRs had to compete for energy with competitors. This work was continued in 1997 by Belpaeme and Birk [7], who introduced different robotic species into the ecosystem. Besides competing for energy, the different species had to cooperate to survive. The last two decades self-sufficient MRs gained interest, and research began to focus on improving the efficiency of individual MRs and robotic teams. The aim was to maximize the time the MRs spent working [8, 9]. Extension of the operation time and range of the MRs, has been realized by incorporating mobile fuelling stations (MFSs) [10, 11]. Strategies have been proposed for coordinating the refuelling activities, in order to improve the efficiency of MR teams. Kim et al. [12] presented a centralized architecture to coordinate the refuelling of unmanned aerial vehicles (UAVs), for long-term persistent mission fulfilment. Hierarchical architectures in the form of a market-based mechanism have been introduced by Marmol et al. [10], Leonard et al. [13].

A self-sufficient MR performs the basic cycles of work, find fuel, and refuel [14]. In order to find fuel a MR has to be capable of locating a fuelling station (FS) and navigating towards it, also called homing behaviour. Several methods can be applied for this purpose, such as infra-red homing system [15], vision and artificial landmarks [16], virtual pheromones [17], environment mapping [18], and motion capture systems [19]. Most of the MRs are powered by

batteries. These MRs can be resupplied by either replacing [20], or recharging the batteries. Recharging can be done either via a direct connection [21], or contactless [22]. The battery voltage slope is often assumed to be linear, in practice this is not the case. Birk [23] proposed to take the non-linearities into account, by using the voltage slope to determine when a MR should go for recharge. It is even possible to recharge a MR during operation via a laser beam [24], although this is not energy efficient. It is beyond the scope of this thesis to elaborate on these strategies, but it has to be noted that these are important basics of self-sufficient MRs.

This thesis focuses on two major decisions that should be taken, in order for a MR to be self-sufficient: refuel event selection, and allocation of the FSs as a shared resource. The major problem is to schedule the refuelling activities of multiple MRs, across multiple FSs, such that the overall mission time is minimized. The goal of this thesis is to develop and compare different methods to solve this problem, with the restriction that a FS can only refuel a single MR at a time. This implies that the MRs should share the available FSs efficiently, in order to prevent depletion, and accomplish a mission in a minimum amount of time.

## 1-1 Problem Statement

Consider an environment that contains: a set  $\mathcal{M}$  of  $M$  FSs (indexed by  $m$ ), and a set  $\mathcal{K}$  of  $K$  MRs (indexed by  $k$ ). The FSs are assumed to have an unlimited energy capacity, and ability to move at a constant velocity  $v_m$ . The MRs are assumed to move at a constant velocity  $v_k$ . Furthermore it is assumed that the MRs only consume energy while moving, at a constant rate  $\dot{E}_k$ . The FSs can only refuel a single MR at a time, and with a constant energy donation rate  $\dot{E}_m$ . The FSs can be heterogeneous in the sense that they can have different maximum velocities  $v_{m,\max}$ , and energy donation rates  $\dot{E}_m$ . The MRs can be heterogeneous in the sense that they can have different maximum velocities  $v_{k,\max}$ , energy consumption rates  $\dot{E}_k$ , and energy capacities  $E_{k,\max}$ .

Each MR starts the mission at its initial location  $p_{k0}$  and has to visit  $N$  waypoints (indexed by  $n$ ), in a pre-determined order. The mission of a single MR is fulfilled when it has visited all waypoints, and is completely refuelled again. Each FS has an initial location  $p_{m0}$ . The FSs can either be mobile, or situated at a fixed location (from now on denoted as: fixed FS), determined by their maximum velocity  $v_{m,\max}$ . A FS is fixed if  $v_{m,\max} = 0$ , and mobile if  $v_{m,\max} > 0$ . A MR can be refuelled by a MFS at a rendezvous location  $f_{mlk}$ ,  $m \in \mathcal{M}, k \in \mathcal{K}$ . The index  $l$  denotes the  $l^{th}$  refuelling task of a FS. The rendezvous locations are not known a priori, and have to be determined in such a way that the time the MRs spend on a refuelling activity is minimized. A single fuel cycle of a MR consists of: visiting waypoints, transitioning to a fuel location, and refuelling. Each MR has a maximum initial energy,  $E_k = E_{k,\max}$ . When refuelling, a MR leaves the FS when its energy level reaches its maximum. If a MR is completely refuelled, it is assumed to be able to visit at least a single waypoint. This implies that the maximum number of fuel cycles a MR can have, is equal to the number of waypoints. The mission of the robotic team is accomplished when all MRs reached their final waypoint, and are completely refuelled. The objective is to minimize the mission completion time  $T_{\text{mission}}$ .

Formally we are interested in solving the following problem:

**Problem 1.** Given:

- $M$  FSs at initial positions  $p_{m0}$ , with: maximum velocity  $v_{m,\max}$ , an unlimited energy capacity, and possibility to refuel MRs at a rate  $\dot{E}_m$ .
- $K$  MRs at initial positions  $p_{k0}$ , with: on-board energy  $E_k$ , energy capacity  $E_{k,\max}$ , energy consumption rate  $\dot{E}_k$ , and maximum velocity  $v_{k,\max}$ .
- $KN$  waypoints  $p_{kn} \in \mathbb{R}^2, k \in \mathcal{K}, n = 1, \dots, N$ , to be visited in a consecutive order by the corresponding  $\text{MR}_k$ .

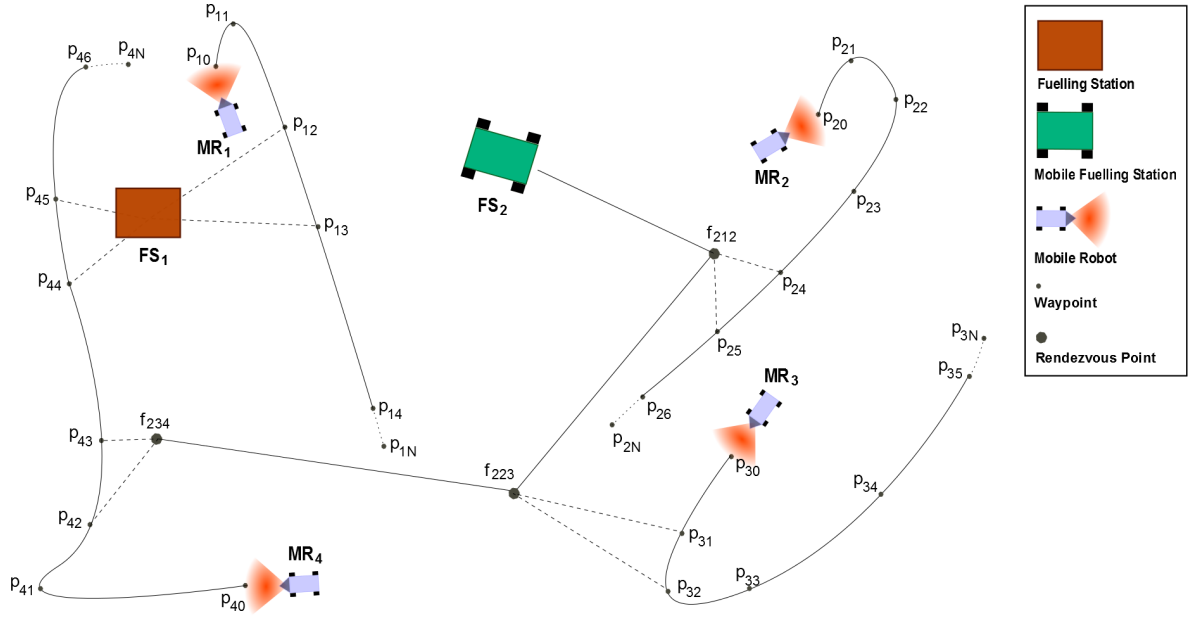
Determine for each:

- MR a set  $\Omega_r$  of refuel points  $\{p_{kn}\}$ , such that:  $\text{MR}_k$  goes for refuel after visiting waypoint  $p_{kn}$ , and visits waypoint  $p_{k(n+1)}$  afterwards.
- MR a set  $\Omega_f$  of rendezvous locations  $f_{mlk}$ , such that:  $\text{MR}_k$  goes for refuel at the rendezvous location of  $\text{FS}_m$ , during the  $l^{\text{th}}$  refuelling task of  $\text{FS}_m$ . In case the FS is mobile, the rendezvous location becomes a decision variable. Otherwise the rendezvous location is equal to the location of the fixed FS.
- MR the ordering in which it is allowed to enter a FS (from now on denoted as: refuel ordering), in case multiple MRs want to enter the same FS during the same timeslot.

Such that:

- $T_{\text{mission}}$  is minimized.
- $E_k > 0, \forall k \in \mathcal{K}$ , at all times.

An example of a mission is illustrated in Figure 1-1. Here the environment consists of 1 fixed FS, 1 MFS, and 4 MRs. Each MR is assigned  $N$  waypoints which have to be visited in a pre-determined order. The direct routes of the MRs are indicated by the solid lines, connecting the waypoints. During operation the MRs have to refuel from time to time, because their energy capacity is insufficient to visit all waypoints at once. The refuelling activities are indicated by the dashed lines. In this situation,  $\text{MR}_1$  will refuel at  $\text{FS}_1$  after visiting  $\{p_{11}, p_{12}\}$ , to be able to visit  $\{p_{13}, \dots, p_{1N}\}$  afterwards.  $\text{MR}_2$  will refuel at  $\text{FS}_2$  (which is a MFS) at rendezvous location  $f_{212}$ , after visiting  $\{p_{21}, \dots, p_{24}\}$ .  $\text{MR}_3$  will refuel at  $\text{FS}_2$ , at the rendezvous location  $f_{223}$  after visiting  $p_{31}$ .  $\text{MR}_4$  will first refuel at  $\text{FS}_2$  at rendezvous location  $f_{234}$  after visiting  $\{p_{41}, p_{42}\}$ , and refuel a second time at  $\text{FS}_1$  after visiting  $\{p_{43}, p_{44}\}$ . In this example it is clear that  $\text{FS}_2$  will first refuel  $\text{MR}_2$ , then  $\text{MR}_3$ , and finally  $\text{MR}_4$ . However it is undetermined in which order  $\text{FS}_1$  is going to refuel  $\text{MR}_1$ , and  $\text{MR}_4$ .  $\text{FS}_1$  can refuel these MRs in the order  $\{\text{MR}_1, \text{MR}_4\}$ , or  $\{\text{MR}_4, \text{MR}_1\}$ , where the MR which is listed first is the robot that is allowed to enter the FS as first. This indicates an important aspect of the problem, that besides selecting the refuel events, and allocation of the FSs, also the refuel ordering has to be taken into account.



**Figure 1-1:** Example of a mission where 4 MRs have to visit  $N$  waypoints in a pre-determined order, and have the possibility of refuelling at a fixed, and mobile FS.

The mathematical formulation of the problem taken under consideration is given below.

### Notations

$\mathcal{J}$  - Set of  $J$  possible refuel orderings (indexed by  $j$ ).

$\mathcal{L}$  - Set of  $L$  refuel tasks for each FS (indexed by  $l$ ). The value of  $L$  is not known a priori, but is coupled to the decision variable  $X$ , which is among others used to select how often a MR will refuel.

$T_{imk}^r$  - Time it takes MR $_k$  to get completely refuelled by FS $_m$  during its  $i^{\text{th}}$  fuel cycle.

$T_{imkj}^w$  - Time MR $_k$  has to wait before it is allowed to enter FS $_m$  during its  $i^{\text{th}}$  fuel cycle, corresponding to refuel ordering  $j$ .

$\tau_{ik}$  - Total time MR $_k$  is moving during its  $i^{\text{th}}$  fuel cycle.

$D_{ik}$  - Distance travelled by MR $_k$  during its  $i^{\text{th}}$  fuel cycle.

$$R_{lmk} = \begin{cases} 1 & \text{if FS}_m \text{ is refuelling MR}_k, \text{ during its } l^{\text{th}} \text{ refuelling task.} \\ 0 & \text{otherwise.} \end{cases}$$

### Decision Variables

A binary decision variable  $X$  can be used for the selection of the refuel events, and the allocation of the FSs, as follows:

$$X_{inmk} = \begin{cases} 1 & \text{if MR}_k \text{ goes for refuel at FS}_m, \text{ after visiting waypoint } p_{kn}, \text{ during its } i^{\text{th}} \text{ fuel cycle.} \\ 0 & \text{otherwise.} \end{cases} \quad (1-1)$$

A binary decision variable  $U$  can be defined, in order to select from all possible refuel orderings the ordering that results in a minimum total mission time.  $U$  can be defined as follows:

$$U_j = \begin{cases} 1 & \text{if } j \text{ corresponds to the chosen refuel ordering.} \\ 0 & \text{otherwise.} \end{cases} \quad (1-2)$$

The rendezvous locations of the MFS, have to be determined such that the total time the MRs spend on a refuelling activity is minimized. Because of this the rendezvous location  $f_{mlk}$ ,  $m \in \mathcal{M}, l \in \mathcal{L}, k \in \mathcal{K}$  is a continuous decision variable.

### Constraints

The constraint that guarantees that the amount of energy each MR spends moving in a fuel cycle is less than its maximum energy, is given as follows:

$$D_{ik}\dot{E}_k < E_{k,\max}, \quad i = 1, \dots, N, k \in \mathcal{K} \quad (1-3)$$

In order to guarantee that each MR travels to a FS after it visited its last waypoint, the following constraint is given:

$$\sum_{m=1}^M X_{inmk} = 1, \quad i \in \{1, \dots, N\}, n = N, k \in \mathcal{K} \quad (1-4)$$

Since a FS can only refuel a single MR at a time, the constraint given in (1-5) makes sure that a FS cannot be used by multiple MRs simultaneously.

$$\sum_{k=1}^K R_{lmk} = 1, \quad l \in \mathcal{L}, m \in \mathcal{M} \quad (1-5)$$

The following constraint guarantees that from all possible refuel orderings, exactly one is selected.

$$\sum_{j=1}^J U_j = 1 \quad (1-6)$$

### Cost Function

The cost function is the total mission time, which can be determined as follows. For each MR: sum up all travel times from the initial position to the final waypoint; subtract the travel times between the waypoints that the MR is travelling via a FS; add the travel times from the waypoints after which is chosen to refuel, to the chosen FSs; add the times the MR spends waiting, add the times the MR spends refuelling; and add the travel times from the chosen FSs to the waypoints that has to be visited after the refuel event. The cost function is given by (1-7).

$$\begin{aligned} T_{\text{mission}} = & \sum_{k \in \mathcal{K}} \sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} \sum_{i=1}^N \sum_{j \in \mathcal{J}} \left( \frac{1}{v_k} [\|p_{kn} - p_{k(n-1)}\| + \right. \\ & + (\|f_{mlk} - p_{kn}\| - \|p_{k(n+1)} - p_{kn}\| + \|p_{k(n+1)} - f_{mlk}\|) X_{inmk}] + \\ & \left. + (T_{imk}^r + T_{imkj}^w U_j) X_{inmk} \right) \end{aligned} \quad (1-7)$$

## Optimization problem formulation

$$\min_{X,f,U} T_{\text{mission}} \quad (1-8a)$$

$$\text{s.t. (1-3)-(1-6)} \quad (1-8b)$$

## 1-2 Problem Analysis

The problem can be divided in the three subproblems listed below, which have to be solved in order to minimize the overall mission time.

1. Find a set of waypoints after which the MRs should go for refuel, and determine to which FSs the MRs should go for refuel;
2. Determine the refuel ordering;
3. Find a set of rendezvous locations.

The first subproblem is a path planning problem. If a single vehicle, and a distance-based metric are considered, the problem is well studied and usually boils down to solve a travelling salesman problem (TSP). A description of the TSP is among others given by Laporte [25]. In general the TSP describes the problem of finding the shortest possible route for a travelling salesman, who has to visit a number of cities exactly once. A generalization of the TSP is the vehicle routing problem (VRP). The VRP describes the problem of finding the shortest possible routes for a number of vehicles, which have to visit a number of customers exactly once. Each route starts and ends at the same location, which is called the depot. A variant of the VRP is the capacity constrained vehicle routing problem (CVRP), which takes a maximum capacity of the vehicles into account. A description of the VRP, and CVRP is among others given by Laporte et al. [26], Laporte and Nobert [27]. Another variant of the VRP is the multi-depot capacity constrained vehicle routing problem (MDCVRP). This is the CVRP, with the extension that there are multiple depots. A restriction is that each route has to start and end at the same depot. The first subproblem is related with the MDCVRP, but has some major differences. Comparing the first subproblem with the MDCVRP: the MRs, FSs, and waypoints can be seen as vehicles, depots, and customers respectively. The differences with the MDCVRP are:

- A MR does not have to start and end a route at the same FS;
- The waypoints are already assigned to each MR, and has to be visited in a pre-determined order;
- The total mission time is minimized, instead of the total travel distance.

The second subproblem is a combinatorial optimization problem, which involves finding all permutations between all possible fuel cycles of all MRs, with the constraint that for each MR, the first fuel cycle should occur before the second fuel cycle, the second fuel cycle should occur before the third fuel cycle, etc.

The third subproblem is an optimization problem, where the rendezvous locations are the decision variables. The objective is to minimize the time the MRs spend during a refuel activity. If the time a MR has to spend refuelling is relative long compared to the time the MR can be moving in a fuel cycle, it is beneficial if the rendezvous location is close to the current location of the MR. This prevents the MR from spending a lot of energy to travel to a rendezvous location, and so decreases the refuel time. On the other hand if the MR can move a long time during a fuel cycle, and little time is needed to be completely refuelled, it can be beneficial if the rendezvous location is closer to the current location of the MFS. Another option is to determine the rendezvous location while taking the locations of a cluster of MRs into account. In that case the rendezvous location should be chosen such that the overall time the MRs in the cluster spend on a refuelling activity is minimized.

The objective function of (1-8) is non-linear because of the multiplication between the decision variables  $f$  and  $X$ , and  $U$  and  $X$ . All constraints are linear. The decision variables consists of continuous ( $f$ ), and binary variables ( $X$ ,  $U$ ). This makes this problem a mixed integer nonlinear program (MINLP). MINLP problems are NP-hard, and generally very complex to solve [28, 29]. Some algorithms to solve a MINLP are: outer approximation, branch-and-bound, extended cutting plane methods, and generalized Bender's decomposition [30]. There are a number of optimization solvers which can be used in MATLAB<sup>®</sup> to solve these class of problems. Some of these solvers are listed below.

- The TOMLAB<sup>®</sup> solver minlpBB solves large, sparse or dense non-linear programming problems. minlpBB implements a branch-and-bound algorithm, searching a tree whose nodes correspond to continuous non-linearly constrained optimization problems.
- The MIDACO solver is suitable for problems with up to several hundreds to some thousands optimization variables. MIDACO implements a derivative-free, heuristic algorithm that treats the problem as black-box.
- The NOMAD solver from the MATLAB opti toolbox, is a derivative free, global MINLP solver.

The solutions of the algorithms to solve a MINLP are suboptimal, and it is hard to verify how far the solutions are from a global optimum.

The goal of this thesis is to develop, and compare several approaches to solve the problem defined in Section 1-1. Since the problem is NP-hard, the computational cost will grow very quickly when the problem size increases. In this thesis methods have been developed, which make it possible to solve this problem for large scale systems, by reducing the computation time at the cost of a suboptimal solution. Unless the well known P=NP problem [31] is solved, it is not possible to find a global optimal solution in polynomial runtime for the original problem definition. We are not striving at solving the P=NP problem. We strive to solve the problem defined in Section 1-1, and related problems using a (sub)optimal approach, which solution quality is acceptable compared to the computation time. A global optimal solution has to be found, which can serve as a benchmark to verify the solution quality of the different approaches. The problem stated in (1-8) is simplified, by only taking fixed FSs into account. This combined with the use of max-plus algebra made it possible to formulate the problem as a mixed integer linear program (MILP). For a MILP problem it is possible

to obtain a global optimum. A MILP belongs to the class of NP-complete problems, which computation time grows very quickly as well when the problem size increases.

There is a wide variety of optimization solvers, which can be used in MATLAB to solve MILP problems. Some of them are listed below.

- CPLEX<sup>®</sup>, an optimization solver from IBM, for integer programming problems, solves very large [32] linear programming problems using either primal or dual variants of the simplex method or the barrier interior point method, convex and non-convex quadratic programming problems, and convex quadratically constrained problems.
- GUROBI<sup>®</sup> is a solver for linear programming, quadratic programming, quadratically constrained programming, mixed integer linear programming, mixed-integer quadratic programming, and mixed-integer quadratically constrained programming.
- SEDUMI is a toolbox for MATLAB, for solving optimization problems with linear, quadratic, and semi definiteness constraints [33].

YALMIP is a modelling language for advanced modelling of convex and non-convex optimization problems [34]. YALMIP is implemented as a MATLAB toolbox and can be used to model convex and non-convex optimization problems. For the computation YALMIP relies on existing optimization solvers. A large variety of optimization solvers can be used in combination with YALMIP, among others the three solvers listed above.

## 1-3 Outline

This thesis discusses three different approaches to schedule the refuelling activities of multiple heterogeneous MRs. Chapter 2 discusses a centralized approach, where a MILP is formulated to solve a simplified version of the problem stated in (1-8). A MILP is formulated for both a distance and a time-based metric. A heuristic is discussed, which is proposed to speed up the computation time. A receding horizon principle is used to solve small scale problems repeatedly, in order to solve a large scale problem where the MRs have to visit a large number of waypoints. The performance of the different methods is analysed by performing some numerical experiments, which will conclude this chapter. Chapter 3 discusses a distributed approach to solve the problem defined in Section 1-1. This approach is distributed in the sense that the decisions are made by each MR individually, based on local knowledge. Two different refuel event selection methods are discussed, which performances are evaluated by simulation experiments at the end of the chapter. A hierarchical approach to schedule the refuelling activities of multiple MRs is presented in Chapter 4. This approach is called hierarchical because the decisions are made for individuals or clusters of MRs, based on local knowledge. In Chapter 5 a comparison is made between the centralized, distributed and hierarchical approaches. The performance of the different approaches is evaluated in terms of total mission time. Furthermore the influence of MFSs on the total mission time is investigated. Finally, in Chapter 6 the conclusions, and proposals for future research will be given.

---

## Chapter 2

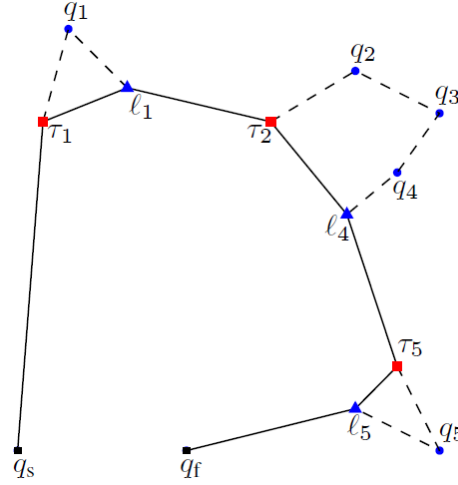
---

# Centralized Approach

This chapter presents a centralized method in order to solve the problem stated in Section 1-1. The approach is called centralized, because a central node performs all calculations, and decision making for all mobile robots (MRs), based on global knowledge. The advantage of a centralized architecture is that global optimal plans can be formulated which lead to global optimal solutions. Drawbacks of this architecture are that it is not robust to dynamic environments, communication failures, and other uncertainties [35]. Furthermore centralized architectures are highly vulnerable because they have a central point of failure [35, 36]. Centralized approaches are most suited for problems involving small robot teams [35, 36], and static environments [36].

Since the problem of Equation (1-8) is a mixed integer nonlinear program (MINLP), the solutions are suboptimal, and it is hard to verify how far the solution is from a global optimum. In order to find a global optimal solution, the problem is simplified by assuming that all fuelling stations (FSs) are situated at a fixed location. This changes the cost function of Equation (1-7), such that the rendezvous location  $f$  is a constant and not a decision variable anymore. By using max-plus algebra, the multiplication in (1-7) between  $X$  and  $U$ , could be transformed into a linear operation. The basics of max-plus algebra, and the exact operations that were done, are explained in detail in Section 2-2. These changes made it possible to transform the cost function, such that it consists of all linear terms. Since the constraints of (1-8) are linear as well, the reformulated problem is a mixed integer linear program (MILP). A solution for this problem is presented for both a distance, and a time-based metric as shown in the following sections. The distance, time, and energy values used in this thesis are expressed in general terms as distance, time, and energy units. One can interpret these units as desired. If one for instance is interested in distances of e.g. metres, or kilometres, the distance units can be interpreted in metres, or kilometres respectively.

In recent literature there are some approaches which schedule the refuelling activities of robotic teams, using a centralized approach. A MILP is formulated by Kim et al. [12], in order to schedule the refuelling activities of unmanned aerial vehicles (UAVs). Klauco et al. [1] formulated a mixed integer second order cone program (MISOCP) for a path planning problem of heterogeneous multi-vehicle systems. The problem they considered is that of a



**Figure 2-1:** Illustration of an optimal path for the heterogeneous vehicle.  $\tau_i$  and  $l_j$  denote the points where the fast vehicle will leave and enter the carrier respectively. The solid line shows the trajectory of the carrier and the dashed lines show the trajectory of the fast vehicle - Adapted from Klauco et al. [1].

vehicle combination, which consists of a slow carrier vehicle with a large operation range, and a fast vehicle with a short operation range. The combined vehicle starts at an initial position and has to visit a number of  $N$  waypoints in a consecutive order. The objective is to minimize the mission completion time, while taking the limited operation range of the fast vehicle into account. Refuelling is assumed to happen instantaneously. The fast vehicle can refuel itself by rendezvousing with the slow carrier. This means the fast vehicle can leave the slow carrier to visit some waypoints, return to refuel, and visit waypoints again. This is illustrated in Figure 2-1. This figure shows that the fast vehicle sometimes leaves the carrier to quickly visit some waypoints and returns to the carrier again to refuel. Our MILP formulation is based on the formulation of Klauco et al. [1]. The problem they consider can be translated in the problem defined in Section 1-1, while only taking fixed FSs into account. The FSs can be considered as fixed carriers, and the MRs as fast vehicles. The major differences are that in our case the FSs are fixed, there are multiple FSs and MRs, there is not necessarily an equal number of FSs and MRs, and the MRs do not necessarily have to return to the same FS.

To the best of our knowledge, we are the first to formulate a MILP in order to schedule the refuelling activities of multiple MRs, across multiple FSs, using a time-based metric, while taking the duration that each MR spends refuelling (from now on denoted as: refuel time) and the refuel ordering into account. Klauco et al. [1] use a time-based metric as well, but assume that the MR can refuel instantaneously. Due to this assumption, the refuel times do not have to be taken into account. Furthermore the system they consider consists of a single mobile fuelling station (MFS) and a single MR. Kim et al. [12] consider a problem where the refuelling activities of UAVs have to be scheduled for persistent mission fulfilment. A distance-based metric is used, which makes it unnecessary to take the refuel times and refuel ordering into account.

## 2-1 Distance-Based Metric

This section discusses the formulation of a MILP to schedule the refuelling activities of multiple heterogeneous MRs, such that the MRs perform a mission while the total travel distance is minimized. This formulation forms the basis for the final MILP formulation, where the total mission time is minimized.

### 2-1-1 MILP Formulation

When using a distance-based metric, and assuming that the FSs are situated at a fixed location, two decisions have to be made. In order to find an optimal schedule, there has to be decided after which waypoints, and to which FSs the MRs go for refuel. These decisions can be made by solving a MILP. The formulation of the MILP is discussed in the rest of this section for a single FS, and a multiple FSs scenario.

#### Single Fuelling Station Scenario

A binary decision variable  $X$  is defined, in order to determine after which waypoints the MRs should go for refuel.  $X$  is defined as follows:

$$X_{ink} = \begin{cases} 1 & \text{if MR}_k \text{ goes for refuel after visiting waypoint } p_{kn}, \text{ during} \\ & \text{its } i^{\text{th}} \text{ fuel cycle.} \\ 0 & \text{otherwise.} \end{cases} \quad (2-1)$$

Based on the MISOCP formulation by Klauco et al. [1], the decision variable is a binary matrix  $X \in \{0, 1\}^{N \times N \times K}$ .  $X_{ink}$  is interpreted as follows: during its  $i^{\text{th}}$  fuel cycle,  $\text{MR}_k$  starts fully refuelled at the FS, visits waypoints  $p_i, \dots, p_n$  without refuelling, and ends the fuel cycle completely refuelled at the FS again. There are two important restrictions. The first is that the MRs start and end a mission at the FS. The second restriction is that when completely refuelled, a MR should at least be able to visit a single waypoint. This implies that each MR can have a maximum of  $N$  fuel cycles.

For a single MR, which has to visit three waypoints, the structure of  $X$  can be given as follows:

$$X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

In this example, the first element in the first row is 1, which means the MR starts its first fuel cycle fully refuelled at the FS, visits waypoint  $p_1$ , and returns to the FS for refuel. The last element of the second row is also 1, which means that during the second fuel cycle, the MR visits waypoints  $\{p_2, p_3\}$  in a consecutive order, and then returns to the FS again. For scenarios with  $K$  MRs,  $X$  is a three-dimensional binary decision matrix, such that the refuel events can be determined for each MR individually.

In order to calculate the distances travelled by each MR, a matrix  $D \in \mathbb{R}^{N \times N \times K}$  is created.  $D$  contains all the euclidean distances, the MRs have to cross to visit the waypoints during

each fuel cycle. An example of a distance matrix for a single MR is given as follows:

$$D = \begin{bmatrix} 6 & 12 & 16 \\ 0 & 8 & 10 \\ 0 & 0 & 8 \end{bmatrix}$$

$D$  has the same structure as  $X$ . The nonzero values of  $D$  represent the distances of the routes that start and end at the FS. During the first fuel cycle the MR has three options, visit: only  $p_1$  (distance is 6),  $p_1$  and  $p_2$  (distance is 12), or  $p_1, p_2$ , and  $p_3$  (distance is 16). During the second fuel cycle the MR can visit: only  $p_2$  (distance is 8), or  $p_2$ , and  $p_3$  (distance is 10). During the third fuel cycle the MR can only visit  $p_3$  (distance is 8).

For scenarios with  $K$  MRs,  $D$  is a three-dimensional matrix which contains all possible travel distances for each MR.

In a similar fashion as the distance matrix, an energy consumption matrix  $E^c \in \mathbb{R}^{N \times N \times K}$  can be constructed, which contains all amounts of energy that each MR will consume during all possible routes. It is assumed that a MR only consumes energy while moving, and at a constant rate. The energy consumption matrix can be calculated by multiplying the distance matrix with the energy consumption rate, as given by (2-2).

$$E_{ink}^c = D_{ink} \dot{E}_k, \quad i = 1, \dots, N, n \in \mathcal{N}, k \in \mathcal{K} \quad (2-2)$$

The energy consumption matrix will be used to guarantee that all MRs do not travel routes of longer distances than their energy capacity allows.

The objective is to minimize the accumulated travel distance  $D_{\text{total}}$ , which can be calculated by the total sum of the distance matrix multiplied with  $X$ . The cost function is given by (2-3).

$$D_{\text{total}} = \sum_{k=1}^K \sum_{n=1}^N \sum_{i=1}^N D_{ink} X_{ink} \quad (2-3)$$

A constraint is formulated in (2-4) to guarantee that the MRs do not travel routes that exceed their energy capacity,

$$E_{ink}^c X_{ink} \leq E_{k,\text{max}}, \quad i = 1, \dots, N, n \in \mathcal{N}, k \in \mathcal{K} \quad (2-4)$$

Another constraint is defined, in order to guarantee that each MR visits each waypoint exactly once. This constraint is given as follows:

$$\sum_{i=1}^w \sum_{n=w}^N X_{ink} = 1, \quad k \in \mathcal{K}, w = 1, \dots, N \quad (2-5)$$

The optimization problem for a single FS scenario, using a distance-based metric can be formulated as follows:

$$\min_X D_{\text{total}} \quad (2-6a)$$

$$\text{s.t. (2-4)-(2-5)} \quad (2-6b)$$

This problem is a linear program (LP) due to the fact that the objective function and constraints are linear. Because the decision variable  $X$  is binary, this problem is a MILP.

### Multiple Fuelling Stations Scenario

For scenarios with multiple FSs, the MRs have the option to start and end the mission at one of the FSs. A MR can end a fuel cycle at a different FS, than where it started the fuel cycle. The MILP formulation defined for the single fuelling station scenario can be extended to a multiple FSs scenario. When the number of FSs increases, the number of possible FSs where a MR can start and end a fuel cycle increases quadratically with the number of FSs.

**Theorem 2.1.** *The number of combinations between locations where a mobile robot can start and end a fuel cycle, increases quadratically with the number of fuelling stations.*

*Proof.* Consider the set  $\mathcal{M} = \{1, 2, \dots, M\}$  of  $M$  FSs. For each  $\text{FS}_i, i \in \mathcal{M}$ , it is possible that a MR starts a fuel cycle at  $\text{FS}_i$ , and ends the fuel cycle at  $\text{FS}_j, j \in \mathcal{M}$ . Since a MR can start a fuel cycle at  $M$  FSs, and end a fuel cycle at  $M$  FSs, the total number of combinations between the locations where a MR can start and end a fuel cycle is  $M^2$ .  $\square$

Based on Theorem 2.1 a set  $\Omega_M = \{s_1, s_2 \mid s_1, s_2 \in \mathcal{M}\}$  is defined, which can contain any possible combination between two elements in  $\mathcal{M}$ . A second set  $\mathcal{Q} = \{c_1, c_2, \dots, c_{M^2} \mid c_1, c_2, \dots, c_{M^2} \in \Omega_M\}$  is defined, which contains all possible combinations between the elements in  $\mathcal{M}$ . The elements in  $\mathcal{Q}$  are indexed by  $q$ .

For a scenario with three fuelling stations, the set  $\mathcal{Q}$  and the corresponding FSs can be defined as given in Table 2-1.

**Table 2-1:** Example of the set  $\mathcal{Q}$ , which contains all possible combinations between FSs where a MR can start and end a fuel cycle, for a scenario with 3 FSs.

| $q$         | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|-------------|---|---|---|---|---|---|---|---|---|
| Start at FS | 3 | 2 | 1 | 1 | 1 | 2 | 2 | 3 | 3 |
| End at FS   | 3 | 2 | 1 | 3 | 2 | 1 | 3 | 2 | 1 |

A binary decision variable  $X$  is defined, in order to determine after which waypoints, and to which FSs the MRs should go for refuel.  $X$  is given as follows:

$$X_{inqk} = \begin{cases} 1 & \text{if MR}_k \text{ starts and ends its } i^{\text{th}} \text{ fuel cycle at the FSs corresponding to } q, \text{ after visiting waypoint } p_{kn}. \\ 0 & \text{otherwise.} \end{cases} \quad (2-7)$$

$X_{inqk}$  is interpreted as follows: during its  $i^{\text{th}}$  fuel cycle,  $\text{MR}_k$  starts fully refuelled at the FS corresponding to  $q$ , visits waypoints  $p_i, \dots, p_n$  without refuelling, and ends the fuel cycle completely refuelled at the FS corresponding to  $q$  again.  $X$  becomes a four-dimensional binary decision matrix  $X \in \{0, 1\}^{N \times N \times M^2 \times K}$ , because in the multiple fuelling stations scenario the allocation of the FSs forms an extra decision in comparison to the single FS scenario.

The distance matrix also becomes a four-dimensional matrix  $D \in \mathbb{R}^{N \times N \times M^2 \times K}$ . Again  $D$  has the same structure as  $X$  and contains the distances of all possible routes, for all MRs, during each fuel cycle.

The energy consumption matrix becomes a four-dimensional matrix as well  $E^c \in \mathbb{R}^{N \times N \times M^2 \times K}$ . This matrix can be calculated again by multiplying the distance matrix with the energy consumption rate, as given by (2-8).

$$E_{inqk}^c = D_{inqk} \dot{E}_k, \quad i = 1, \dots, N, n \in \mathcal{N}, q \in \mathcal{Q}, k \in \mathcal{K} \quad (2-8)$$

For a multiple FSs scenario the cost function can be calculated as follows:

$$D_{\text{total}} = \sum_{k=1}^K \sum_{q=1}^{M^2} \sum_{n=1}^N \sum_{i=1}^N D_{inqk} X_{inqk} \quad (2-9)$$

The constraints of the single FS scenario are applicable to the multiple FSs scenario as well, when the extended decision, distance, and energy consumption matrices are taken into account. The constraint that prevents that the amount of energy consumed by a MR during a fuel cycle is larger than its maximum energy capacity allows, is given as follows:

$$E_{inqk}^c X_{inqk} \leq E_{k,\max}, \quad i = 1, \dots, N, n \in \mathcal{N}, q \in \mathcal{Q}, k \in \mathcal{K} \quad (2-10)$$

The constraint that guarantees that each waypoint is visited exactly once becomes:

$$\sum_{q=1}^{M^2} \sum_{i=1}^w \sum_{n=w}^N X_{inqk} = 1, \quad k \in \mathcal{K}, w = 1, \dots, N \quad (2-11)$$

A third constraint has to be formulated, in order to guarantee that if a MR ends a fuel cycle at a certain FS, it departs the next fuel cycle from the same FS. This is illustrated by the following example. Consider the scenario of a single MR, which has to visit 3 waypoints, and has the option of refuelling at two FSs. Consider which the decision matrix  $X$  to be as given below.

$$X_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, X_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, X_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, X_4 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

Here the index of  $X$  represents the value of  $q$ , the index of the set  $\mathcal{Q}$ . For this example  $\mathcal{Q}$  and the corresponding FSs where a MR can start and end a fuel cycle are defined as given in Table 2-2. In this example the first element in the first row of  $X_3$  is 1. Since the first element

**Table 2-2:** The set  $\mathcal{Q}$ , which contains all combinations between FSs where a MR can start and end a fuel cycle, for a scenario with 2 FSs.

| $q$         | 1 | 2 | 3 | 4 |
|-------------|---|---|---|---|
| Start at FS | 2 | 1 | 1 | 2 |
| End at FS   | 2 | 1 | 2 | 1 |

in the first row is 1, this indicates that the MR visits only the first waypoint during its first fuel cycle. From Table 2-2 it can be seen that because an element in  $X_3$  is 1, the MR starts this fuel cycle at FS<sub>1</sub> and ends the fuel cycle at FS<sub>2</sub>. Because the MR ends this fuel cycle at FS<sub>2</sub>, it has to depart the next fuel cycle from FS<sub>2</sub>. In the MILP formulation a constraint is implemented which guarantees that in this case the second row of both  $X_2$ , and  $X_3$  are zero

rows. Since according to Table 2-2 if one of these rows contained a 1, would mean that the MR starts this fuel cycle at  $FS_1$ . This is however impossible, since the MR just got refuelled by  $FS_2$ . In this example, the third element of the second row of  $X_4$  is 1, which means that the MR departs the second fuel cycle from  $FS_2$ , visits the second, and third waypoint, and ends the fuel cycle at  $FS_1$ .

Lets introduce the variables  $Y_{ik}^s$ , and  $Y_{ik}^e$ , which contains for each  $i^{\text{th}}$  fuel cycle the indices of the FSs, where  $MR_k$  starts and ends the fuel cycle respectively. With these variables the constraint of (2-12) can be formulated. This constraint guarantees that if a MR ends a fuel cycle at a certain FS, it departs the next fuel cycle from the same FS.

$$Y_{ik}^s = Y_{(i-1)k}^e, \quad i = 2, \dots, N, k \in \mathcal{K} \quad (2-12)$$

With the defined cost function and constraints, the optimization problem for a multiple FSs scenario, using a distance-based metric can be formulated as follows:

$$\min_X D_{\text{total}} \quad (2-13a)$$

$$\text{s.t. (2-10)-(2-12)} \quad (2-13b)$$

## 2-2 Time-Based Metric

When using a distance-based metric it is possible that MRs will cluster around one FS, while other FSs are free. This can be prevented by considering a time-based metric. If the objective is to minimize the total mission time, MRs are more likely to go to other FSs if the closest FS is occupied. This can increase the total travel distance of the MRs, but results in a more equal distribution of MRs across the FSs, and faster accomplished missions. Using a time-based metric introduces extra difficulties. Since a FS can only refuel a single MR at a time, the duration that a MR actually spends refuelling, and the order in which the robots are allowed to enter a FS should be taken into account. If a FS is currently occupied, and another MR wants to refuel at the same FS, it has to wait until the FS comes free again before it can start refuelling. This implies that the times MRs are waiting until a FS comes free, should also be taken into account. Due to the fact that the FSs can refuel a single MR at a time, each refuel event of an individual MR can influence all future refuel events of all MRs. This implies that all possible actions of all MRs should be considered, in order to obtain a global optimal solution.

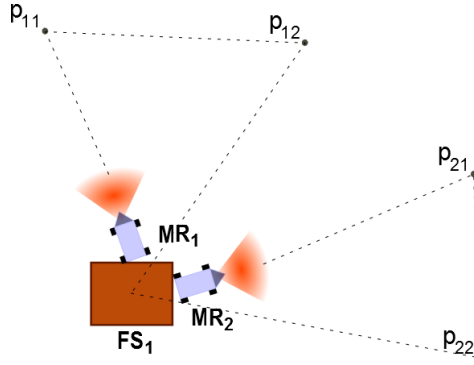
### 2-2-1 MILP Formulation

The MILP formulation using a time-based metric is build on the basis of the MILP using a distance-based metric, discussed in Section 2-1. This formulation is extended with: the translation from travel distances to travel times, refuel times, refuel ordering selection, and waiting times.

#### Refuel Ordering Selection

If multiple MRs want to enter the same FS during the same timeslot, the refuel ordering has to be determined. Consider the scenario of Figure 2-2, where two MRs start completely

refuelled at the FS, and each MR has to visit two waypoints in a pre-determined order. In this figure for each MR the shortest possible route is indicated by a dashed line. When using a distance-based metric, these routes would be the optimal routes if the energy capacity of each MR is sufficient to visit both waypoints during one fuel cycle. However when using a time-based metric it is possible that alternative routes will result in a shorter total mission time. For  $MR_k$ ,  $k = 1, 2$  there are two possible routes:



**Figure 2-2:** Scenario with 1 FS, and 2 MRs, which both have to visit 2 waypoints.

1. Visit  $p_{k1}$ , go to  $FS_1$  for refuel, visit  $p_{k2}$ , return to  $FS_1$  for refuel;
2. Visit  $p_{k1}$ , and  $p_{k2}$  in one fuel cycle, return to  $FS_1$  for refuel.

Each MR can have a maximum of two fuel cycles. All possible refuel orderings should be taken into account, in order to find a global optimal solution. The total number of refuel orderings is denoted by  $J$ , and a single refuel ordering is indexed by  $j$ . For this example there are 6 possible refuel orderings, given in Table 2-3. In this table each row represents a refuel ordering.  $MR_{ki}$  is interpreted as follows:  $MR_k$  is allowed to refuel during its  $i^{\text{th}}$  fuel cycle if the preceding MR in the row is done refuelling. The first row describes the following refuel ordering: During its first fuel cycle  $MR_1$  is allowed to refuel as soon as it reached the FS, during its second fuel cycle  $MR_1$  is again allowed to refuel as soon as it reaches the FS (because the preceding MR in the row is  $MR_1$  as well), during its first fuel cycle  $MR_2$  is allowed to refuel when  $MR_1$  is done refuelling at the end of the second fuel cycle, during its second fuel cycle  $MR_2$  is allowed to refuel as soon as it reaches the FS (because the preceding MR in the row is  $MR_2$  as well). In order to find a global optimal solution, the total mission

**Table 2-3:** All possible refuel orderings for a scenario of 2 MRs which have to visit 2 waypoints in a pre-determined order.

|           |           |           |           |
|-----------|-----------|-----------|-----------|
| $MR_{11}$ | $MR_{12}$ | $MR_{21}$ | $MR_{22}$ |
| $MR_{11}$ | $MR_{21}$ | $MR_{12}$ | $MR_{22}$ |
| $MR_{11}$ | $MR_{21}$ | $MR_{22}$ | $MR_{12}$ |
| $MR_{21}$ | $MR_{11}$ | $MR_{12}$ | $MR_{22}$ |
| $MR_{21}$ | $MR_{11}$ | $MR_{21}$ | $MR_{12}$ |
| $MR_{21}$ | $MR_{22}$ | $MR_{11}$ | $MR_{12}$ |

time should be calculated for each refuel ordering. The global minimum, can than be found by selecting the refuel ordering that corresponds to the shortest total mission time.

In order to determine all valid refuel orderings, the MATLAB function PERMS could be used. This function returns all permutations of the elements in the input vector. By creating an input vector which contains all possible refuelling activities  $MR_{ki}$ ,  $i = 1, \dots, N, k = 1, \dots, K$ , the PERMS function will calculate all permutations. A drawback of using PERMS for this purpose is that the function also returns all invalid permutations, e.g. a refuel ordering where  $MR_{12}$  occurs before  $MR_{11}$ . This means all invalid combinations should be removed from the outcome afterwards. Another drawback from the PERMS function is that it can only handle up to 10 inputs, because the output for 11 inputs takes over 3 GB. In example for a scenario with 2 MRs, this means that the PERMS function can only calculate the refuel orderings up to 5 waypoints for each MR.

A recursive function SEQUENCES had been implemented, in order to calculate the refuel orderings for larger scenarios and prevent unnecessarily calculated permutations. SEQUENCES only returns the valid refuel orderings. This function requires an input vector  $U$  which contains: all possible refuelling activities of all MRs, the number of MRs  $K$ , the number of waypoints  $N$ , and a set  $S = \emptyset$ .  $S$  is used to store the refuel ordering. The algorithm of SEQUENCES is explained by Algorithm 1. The major difference with the PERMS function is

---

**Algorithm 1** SEQUENCES

---

```

1: function SEQUENCES( $KN - 1, U, S$ )
2:   for  $i = 1$  to  $length(U)$  do
3:     insert element  $i$  from  $U$  in  $S$ 
4:     remove element  $i$  from  $U$ 
5:     if  $S$  is valid then
6:       SEQUENCES( $KN - 1, U, S$ )
7:     else
8:       reinsert element  $i$  from  $S$  in  $U$ 
9:       remove element  $i$  from  $S$ 
10:    end if
11:  end for
12: end function

```

---

that after inserting an element in  $S$ , SEQUENCES immediately checks if the refuel ordering is valid. If this is the case SEQUENCES will continue with inserting another element in  $S$ , else all orderings that begin with this invalid combination of elements will be left out. As a comparison, the output of PERMS for 10 inputs takes over 300 MB, and the output of the SEQUENCES function for a scenario with 5 MRs, which all have to visit 2 waypoints (this is the worst case scenario where the number of inputs is 10, and the solution can still be calculated in reasonable time) the output takes 9 MB. Calculating all valid refuel orderings can be very computationally complex, because the total number of possible refuel orderings scales badly when the number of MRs, or waypoints increases.

**Theorem 2.2.** *All valid combinations of refuel activities for a scenario with  $K$  mobile robots which have a maximum of  $N$  fuel cycles, can be calculated in  $O((KN)!) time.$*

*Proof.* The computation time of finding all permutations of  $KN$  elements is  $O((KN)!) [37]$ . The number of valid combinations of refuel activities is smaller than the number of all permutations, this implies that the computation time of finding all valid combinations of refuel activities is also  $O((KN)!)$ .  $\square$

The  $O()$  notation is used to indicate the computational complexity of a problem. This notation represents the worst case computation time. If for example a problem with an input  $n$  is defined as  $O(n!)$ , it means that in the worst case scenario the problem scales with  $n!$ .

The possible refuel orderings are always the same, for the same scenarios. Because of this, the refuel orderings can be pre-calculated and stored in a lookup table. When solving the MILP for a specific scenario, the possible refuel orderings for this scenario can be loaded from the lookup table and do not have to be calculated. This can significantly reduce the total computation time, since the computational complexity of finding all possible refuel orderings is  $O((KN)!)$ .

### Max-plus algebra

Max-plus algebra is used to select from all possible refuel orderings, the refuel ordering that results in the minimum total mission time. Here some parts of the basic definition of max-plus algebra are given as defined by van den Boom et al. [38].

Define:

$$\epsilon = -\infty$$

$$\mathbb{R}_\epsilon = \mathbb{R} \cup \{\epsilon\}$$

A binary max-plus decision variable  $u$  is defined as:

$$u \in \mathbb{B}_\epsilon = \{0, \epsilon\}$$

The adjoint variable of  $u$  is  $\bar{u}$ , for which counts:

$$\bar{u} = \begin{cases} 0 & \text{if } u = \epsilon \\ \epsilon & \text{if } u = 0 \end{cases}$$

As the name already suggests, in max-plus algebra there are two possible operations: either taking the maximum, or perform an addition.

### Optimization Problem Formulation

The same decision variable  $X$  as defined in (2-7) is used, in order to determine after which waypoints the MRs, and to which FSs the MRs should go for refuel.

A binary max-plus decision vector  $U \in \mathbb{B}_\epsilon^R$  is defined as follows:

$$U_j = \begin{cases} 0 & \text{if } j \text{ is the selected refuel ordering.} \\ \epsilon & \text{otherwise.} \end{cases} \quad (2-14)$$

In combination with a constraint that guarantees that a single element in  $U$  is equal to 0, and all other elements are equal to  $\epsilon$ ,  $U$  is used as a selection vector to select the refuel ordering that results in the minimum mission completion time. This will be explained in more detail later on in this section.

A matrix  $T \in \mathbb{R}^{N \times N \times M^2 \times K}$  is created, which contains all possible travel times of all MRs.  $T$  has the same structure as  $D$ , and can be calculated by dividing the distance matrix by the corresponding maximum velocity of each MR. The calculation of  $T$  is given as follows:

$$T_{inqk} = \frac{D_{inqk}}{v_{k,\max}}, \quad i = 1, \dots, N, n \in \mathcal{N}, q \in \mathcal{Q}, k \in \mathcal{K} \quad (2-15)$$

A refuel time matrix  $\Gamma \in \mathbb{R}^{N \times N \times M^2 \times K}$  is created, in order to determine the amount of time each MR has to refuel during each fuel cycle.  $\Gamma$  has the same structure as  $T$ , and is calculated as follows:

$$\Gamma_{inqk} = \frac{E_{k,\max} - D_{inqk} \dot{E}_k}{\dot{E}_q}, \quad i = 1, \dots, N, n \in \mathcal{N}, q \in \mathcal{Q}, k \in \mathcal{K} \quad (2-16)$$

Here  $\dot{E}_q$  is the energy donation rate of the FS where a MR can end a fuel cycle, corresponding to  $q$ , index of the set  $\mathcal{Q}$ .

A refuel event matrix  $F \in \mathbb{R}^{KN \times 2 \times M \times J}$  is created, which contains the actual times that the FSs will start and end a refuelling task. Since the mission of a MR is completed when it is completely refuelled after visiting its last waypoint, the maximum value in the refuel event matrix will be equal to the total mission time.  $F$  is a four dimensional matrix. The first two dimensions are used to describe the start, and end times of all refuel events at a single FS. In order to describe the start, and end times of all refuel events for all FSs, a third dimension is applied. The fourth dimension is used to describe the start, and end times of all refuel events, for all refuel orderings. With the use of the decision vector  $U$ , the refuel ordering that results in the shortest total mission time is selected. Consider a scenario of 2MRs, which both have to visit 2 waypoints, and let the refuel ordering be:  $\{\text{MR}_{11}, \text{MR}_{21}, \text{MR}_{12}, \text{MR}_{22}\}$ . For this scenario the refuel event matrix of  $\text{FS}_m$ , is calculated as follows:

$$F_m = \begin{bmatrix} \max(X_{1nq1}T_{1nq1}) & F_{m(1,1)} + \max(X_{1nq1}\Gamma_{1nq1}) \\ \max(F_{m(1,2)}, X_{1nq2}T_{1nq2}) & F_{m(2,1)} + \max(X_{1nq2}\Gamma_{1nq2}) \\ \max(F_{m(2,2)}, F_{m(1,2)} + X_{2nq1}T_{2nq1}) & F_{m(3,1)} + \max(X_{2nq1}\Gamma_{2nq1}) \\ \max(F_{m(3,2)}, F_{m(2,2)} + X_{2nq2}T_{2nq2}) & F_{m(4,1)} + \max(X_{2nq2}\Gamma_{2nq2}) \end{bmatrix}, \quad n = 1, \dots, N, q \in \mathcal{Q} \quad (2-17)$$

Each row of  $F_m$  indicates a refuel event at  $\text{FS}_m$ . The left and right column correspond to the start, and end times of a refuel event respectively. The decision variable  $X_{inqk}$ , travel

time matrix  $T_{inqk}$ , and refuel time matrix  $\Gamma_{inqk}$  correspond to the  $i^{\text{th}}$  fuel cycle of  $\text{MR}_k$ , for  $n = 1, \dots, N$ , and for all  $q \in \mathcal{Q}$  that represent ending a fuel cycle at  $\text{FS}_m$ . It can be seen that each row of  $F_m$  represents the refuel event of the corresponding MR in the refuel ordering. The first, and second row contain the start and end times of a refuel event during the first fuel cycle of  $\text{MR}_1$ , and  $\text{MR}_2$  respectively. The third, and fourth row contain the start and end times of a possible refuel event during the second fuel cycle of  $\text{MR}_1$ , and  $\text{MR}_2$  respectively. If a MR visits both waypoints in a single fuel cycle, the start and end times in  $F_m$  that correspond to a refuel event during its second fuel cycle are both equal to the value in the second column of the overlying row. Also if a MR goes for refuel at a different FS, the start and end times in  $F_m$  that correspond to a refuelling activity of this MR will be equal to the value in the second column of the overlying row.

The cost function is the total mission time  $T_{\text{mission}}$ , which is given by (2-18).

$$T_{\text{mission}} = \max(\max(F) + U_j), \quad j = 1, \dots, J \quad (2-18)$$

Here the  $\max()$  operator is interpreted as the operator that finds the maximum value in a matrix. Since  $F$  contains all refuel events at all FSs, the maximum value in  $F$  corresponds to the time that the last MR is done refuelling during its final fuel cycle. Thus the maximum value in  $F$  corresponds to the mission completion time.

All constraints defined in Section 2-1 for the multiple FSs scenario using a distance-based metric, are still applicable. For a time-based metric, another constraints has to be defined to guarantee that from all possible refuel orderings, a single refuel ordering is selected. This constraint is given by (2-19).

$$\sum_{j=1}^J \frac{\bar{U}_j}{\epsilon} = 1. \quad (2-19)$$

Here  $J$  is the total number of all possible refuel orderings.

The optimization problem for a multiple FSs scenario using a time-based metric can be formulated as follows:

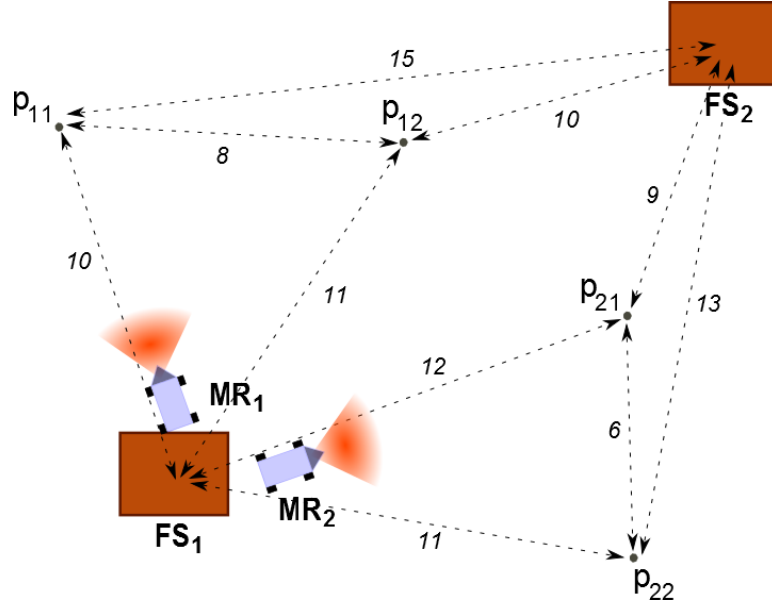
$$\min_{X, U} T_{\text{mission}} \quad (2-20a)$$

$$\text{s.t. (2-10)-(2-12), and (2-19)} \quad (2-20b)$$

Solving this MILP formulation results in a global optimum. Since all possible refuel orderings are taken into account, the computation time increases extremely quickly when the problem scale increases. A heuristic and the use of a receding horizon principle are proposed in the next sections, which make it possible to solve this problem for larger scale scenarios.

## Numerical Example

The principle of the MILP formulation is illustrated by a numerical example. Consider the scenario of Figure 2-3. The illustrated environment contains 2 FSs, and 2 MRs which both have to visit 2 waypoints. In this figure the paths the MRs can travel are indicated by the dashed arrows. The distance of each path is given in *italics* next to the middle of each arrow. Let  $v_{k,\max} = 1$ ,  $E_{k,\max} = 45$ ,  $\dot{E}_k = 1.5$ ,  $\forall k \in \mathcal{K}$ , and  $\dot{E}_m = 2$ ,  $\forall m \in \mathcal{M}$ . This means that the



**Figure 2-3:** A scenario with 2 FSs, and 2MRs, which both have to visit 2 waypoints.

MRs: travel with a velocity of 1 distance unit per time unit, and when completely refuelled the MRs can travel  $45/1.5 = 30$  distance units before their energy level becomes zero. When refuelling, the energy level of a MR increases with 2 energy units per time unit.

For this example  $q$  is defined as given in Table 2-2. The travel time matrix has the following values:

$$T_{11} = \begin{bmatrix} 30 & 33 \\ 0 & 20 \end{bmatrix}, T_{21} = \begin{bmatrix} 20 & 28 \\ 0 & 22 \end{bmatrix}, T_{31} = \begin{bmatrix} 25 & 28 \\ 0 & 21 \end{bmatrix}, T_{41} = \begin{bmatrix} 25 & 34 \\ 0 & 21 \end{bmatrix};$$

$$T_{12} = \begin{bmatrix} 18 & 28 \\ 0 & 26 \end{bmatrix}, T_{22} = \begin{bmatrix} 24 & 28 \\ 0 & 22 \end{bmatrix}, T_{32} = \begin{bmatrix} 21 & 31 \\ 0 & 24 \end{bmatrix}, T_{42} = \begin{bmatrix} 21 & 27 \\ 0 & 24 \end{bmatrix}.$$

The refuel time matrix contains the following values:

$$\Gamma_{11} = \begin{bmatrix} 22.5 & 24.75 \\ 0 & 15 \end{bmatrix}, \Gamma_{21} = \begin{bmatrix} 15 & 21 \\ 0 & 16.5 \end{bmatrix}, \Gamma_{31} = \begin{bmatrix} 18.75 & 21 \\ 0 & 15.75 \end{bmatrix}, \Gamma_{41} = \begin{bmatrix} 18.75 & 25.5 \\ 0 & 15.75 \end{bmatrix};$$

$$\Gamma_{12} = \begin{bmatrix} 13.5 & 21 \\ 0 & 19.5 \end{bmatrix}, \Gamma_{22} = \begin{bmatrix} 18 & 21 \\ 0 & 16.5 \end{bmatrix}, \Gamma_{32} = \begin{bmatrix} 15.75 & 23.25 \\ 0 & 18 \end{bmatrix}, \Gamma_{42} = \begin{bmatrix} 15.75 & 20.25 \\ 0 & 18 \end{bmatrix}.$$

The energy consumption matrix has the following values:

$$E_{11}^c = \begin{bmatrix} 45 & 49.5 \\ 0 & 30 \end{bmatrix}, E_{21}^c = \begin{bmatrix} 30 & 42 \\ 0 & 33 \end{bmatrix}, E_{31}^c = \begin{bmatrix} 37.5 & 42 \\ 0 & 31.5 \end{bmatrix}, E_{41}^c = \begin{bmatrix} 37.5 & 51 \\ 0 & 31.5 \end{bmatrix};$$

$$E_{12}^c = \begin{bmatrix} 27 & 42 \\ 0 & 39 \end{bmatrix}, E_{22}^c = \begin{bmatrix} 36 & 42 \\ 0 & 33 \end{bmatrix}, E_{32}^c = \begin{bmatrix} 31.5 & 46.5 \\ 0 & 36 \end{bmatrix}, E_{42}^c = \begin{bmatrix} 31.5 & 40.5 \\ 0 & 36 \end{bmatrix}.$$

Consider the decision matrix  $X$  to be:

$$X_{11} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, X_{21} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, X_{31} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, X_{41} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix};$$

$$X_{12} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, X_{22} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, X_{32} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, X_{42} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

The first element in the first row of  $X_{21}$ , and the second element in the second row of  $X_{31}$  are 1. This means that  $MR_1$  will: start its mission at  $FS_1$ , visit  $p_1$ , go for refuel at  $FS_1$ , visit  $p_2$ , and end its mission by refuelling at  $FS_2$ . Since the second element in the first row of  $X_{22}$  is 1,  $MR_2$  will: start its mission at  $FS_1$ , visit  $\{p_1, p_2\}$ , and end its mission by refuelling at  $FS_1$ .

For this scenario, there are 6 possible refuel orderings, which are defined in the same way as in Table 2-3. Let the max-plus binary decision variable  $U$  be:

$$U = \begin{bmatrix} \epsilon & 0 & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix}$$

Since the second element of  $U$  is 0, the selected refuel ordering is  $\{MR_{11}, MR_{21}, MR_{12}, MR_{22}\}$ , corresponding to the second row of Table 2-3.

For these  $X$ , and  $U$ , the refuel event matrix  $F$  will be:

$$F_1 = \begin{bmatrix} 20 & 35 \\ 35 & 56 \\ 56 & 56 \\ 56 & 56 \end{bmatrix}, F_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 59 & 77 \\ 77 & 77 \end{bmatrix}.$$

where  $F_1$ , and  $F_2$  contain the actual times of a refuel event at  $FS_1$ , and  $FS_2$  respectively. Observing  $F$  and the refuel ordering, the following can be noticed.  $MR_2$  finished its mission after 56 time units in a single fuel cycle, after refuelling at  $FS_1$ , and  $MR_1$  finished its mission after 77 time units in two fuel cycles, after refuelling at  $FS_2$ . The total mission time is 77 time units, which is the maximum of the individual mission completion times.

### 2-2-2 Heuristic 1: Fixed Refuel Orderings

This heuristic is proposed to speed up the computation time. The total number of all possible routes and refuel orderings, scales very bad when the problem size increases. Based on insights gathered from simulations, and results from literature, a heuristic is proposed that reduces the total number of refuel orderings.

While solving the MILP for different scenarios, and during simulations using the distributed, and hierarchical approach which will be discussed in the next chapters, a useful insight is gained. It is observed that refuel orderings such as  $\{MR_{11}, MR_{12}, MR_{21}, MR_{22}\}$ , where one MR is refuelling twice before another MR is refuelling, rarely occur. Sempé et al. [39] show that if homogeneous MRs are sharing a single FS without any communication, the MRs will alternate at the FS in a fixed order.

Based on these insights the following heuristic is defined:

#### **Heuristic 1.** *Fixed Refuel Orderings*

Instead of taking all possible refuel orderings into account, only take the refuel orderings into account where the refuel events of the MRs occur in a fixed order.

Applying this heuristic, significantly reduces the total number of refuel orderings. This leads to a significant reduction of: decision variables in  $U$ , and the size of the refuel event matrix  $F$ . For the scenario of 2 MRs, which both have to visit 2 waypoints the refuel orderings that have to be taken into account are given in Table 2-4. Comparing Table 2-4 with Table 2-3, it can be seen that the number of refuel orderings that are taken into account reduced from 6 to 2. According to Theorem 2.2 the computation time of all possible refuel orderings

**Table 2-4:** All refuel orderings taken into account when using Heuristic 1, for a scenario of 2 MRs which have to visit 2 waypoints in a pre-determined order.

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| MR <sub>11</sub> | MR <sub>21</sub> | MR <sub>12</sub> | MR <sub>22</sub> |
| MR <sub>21</sub> | MR <sub>11</sub> | MR <sub>12</sub> | MR <sub>22</sub> |

is  $O((KN)!)$ . When computing all refuel orderings taken into account by Heuristic 1, the number of waypoints is not a limiting factor anymore.

**Theorem 2.3.** *For a scenario with  $K$  MRs which have a maximum of  $N$  fuel cycles, the refuel orderings where the refuel events of the MRs alternate in a fixed order, can be calculated in  $O(K!)$  time.*

*Proof.* Since the refuel events of the MRs alternate in a fixed order, a repetitive pattern occurs. The refuel events of all MRs during the  $(i + 1)^{\text{th}}$  fuel cycle, occur in the same order as the refuel events during the  $i^{\text{th}}$  fuel cycle. This implies that if the refuel ordering for the first fuel cycle of all MRs is determined, the refuel ordering for all  $N$  fuel cycles is determined. Calculating all possible refuel orderings for the first fuel cycle is the same as finding all permutations between  $K$  robots, which can be done in  $O(K!)$  time [37].  $\square$

Since the optimization problem still belongs to the class NP-complete, the computation time still increases very quickly as the problem scale increases. However with this heuristic the computation time reduced significantly, and it is possible to find a (sub)optimal solution for larger scale scenarios, compared to the approach that takes all possible refuel orderings into account (from now on referred to as: optimal approach). In Section 2-3, the computation time of the optimal approach and Heuristic 1 is compared.

### 2-2-3 Solving the Problem Using a Receding Horizon Principle

The problem scale for which the optimal approach can be solved in reasonable time is very limited. For scenarios with 2 FSs, the maximum number of waypoints for which the optimal approach can find a solution in reasonable time: is 5, and 2 waypoints, for scenarios with 2, and 3 MRs respectively. When comparing Theorem 2.1 with Theorem 2.2, it can be observed that the problem scale increases faster when the total number of MRs, or waypoints increases than when the total number of FSs increases. Since the total number of MRs, and waypoints seems to be the bottlenecks, it is proposed to solve the MILP using a receding horizon principle. This principle solves subproblems of the optimization problem repeatedly for a prediction horizon of  $W$  waypoints, such that eventually problems can be solved for an arbitrary number of  $N$  waypoints. The basic algorithm behind the receding horizon principle works as described in Algorithm 2. Shifting the prediction horizon can be done periodically,

**Algorithm 2**


---

```

1: while Mission is not completed do
2:   Solve optimization problem for  $W$  a subset of  $N$  waypoints
3:   Check  $X$ , and statuses of MRs, to determine which event will occur next
4:   Corresponding to the time of the next event, update:
       • Statuses, positions, and energy levels of all MRs
       •  $T_{\text{mission}}$ 
5:   if Event = “waypoint reached” then
6:     Shift the prediction horizon for the corresponding MR
7:   end if
8: end while

```

---

or event driven. For this type of problems it makes no sense to shift the prediction horizon periodically. If for example the MRs just started a mission, the solution to the optimization problem will probably not change until an event happens, i.e. a MR reached a waypoint. Since the solution is more likely to change after an event has happened, here the prediction horizon is shifted on an event driven basis. The following events trigger a new calculation of the optimization problem:

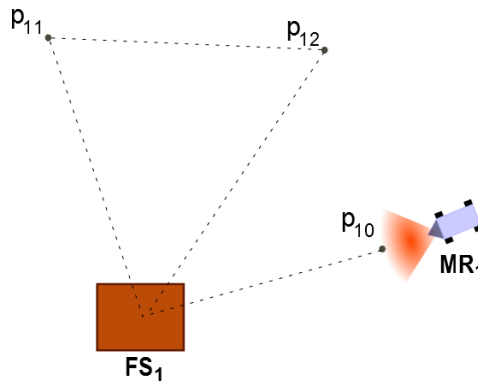
- A MR reached a waypoint;
- A MR reached a FS;
- A MR is done refuelling.

Consider the prediction horizon to be 2 waypoints. Initially the optimization problem will be solved for all MRs, for  $\{p_1, p_2\}$ . After solving the optimization problem, it is checked which MR will reach the next event as first (denoted as  $\text{MR}_{\text{next}}$ ). The positions, and energy levels of all MRs, and total mission time are updated, such that they correspond to the time of the next event. If the next event is that  $\text{MR}_{\text{next}}$  reaches a waypoint, the prediction horizon for  $\text{MR}_{\text{next}}$  is shifted with 1. Which means the next time the optimization problem will be solved for:  $\text{MR}_{\text{next}}$  for  $\{p_2, p_3\}$ , and for the other MRs for  $\{p_1, p_2\}$  again. When the next event is that  $\text{MR}_{\text{next}}$  reaches a FS, it is checked if the FS is occupied, or if another MR is on its way to the same FS and is allowed to refuel first. If this is not the case,  $\text{MR}_{\text{next}}$  starts refuelling. Otherwise the MR will wait in the queue. When the next event is that  $\text{MR}_{\text{next}}$  is done refuelling, it is checked if  $\text{MR}_{\text{next}}$  has visited all its waypoints. When this is the case,  $\text{MR}_{\text{next}}$  completed its mission. Otherwise  $\text{MR}_{\text{next}}$  will continue visiting waypoints. These steps are repeated, until all MRs accomplished their mission.

It has to be noted that the previous assumption that each MR starts a mission at a FS, does not hold for the receding horizon principle. Since events can happen when some MRs are not located at a FS (e.g. in between two waypoints), the optimization problem should be solvable with the MRs having arbitrary initial positions. Also the constraint of (2-10) does not hold anymore, since the MRs are not always completely refuelled when the optimization problem is solved. The MILP formulation of the optimal approach needs some adjustments, before it can be used in a receding horizon fashion.

The first row of the travel distance matrix  $D$  as given in Section 2-1, contains all euclidean distances a MR can travel during the first fuel cycle. Originally the first fuel cycle consisted of: starting at a FS, visiting waypoint(s), end at a FS. When introducing that a MR can start at any arbitrary initial position, the first row of  $D$  contains all euclidean distances of a fuel cycle that consists of: starting at an initial position, visiting waypoint(s), end at a FS. Since the energy consumption, travel time, and refuel time matrices are all build from  $D$ , also the values in the first row of these matrices will change. A drawback from this adjustment is that a problem that could previously be solved for each MR visiting  $N$  waypoints, can now be solved for  $N - 1$  waypoints. This is illustrated by the following example.

Consider a scenario of 1 FS, and 1 MR which has to visit 2 waypoints in the order  $\{p_{11}, p_{12}\}$ . This scenario is illustrated in Figure 2-4. The initial position of the MR is indicated with  $p_{10}$ , and a possible route is indicated by the dashed lines. When using the optimal approach, the MR would start at the FS and the decision matrix  $X$  would be  $N \times N$ . Now if the MR can have any initial position, an extra decision has to be made. When starting at the initial position, it has to be determined if the MR visits the next waypoint directly, or travels via the FS (see the indicated route in Figure 2-4). This causes that the decision matrix  $X$  becomes  $(N + 1) \times (N + 1)$ . Because of this extra decision the number of waypoints for which a solution can be found in reasonable time will decrease with 1. Since 1 decision variable is



**Figure 2-4:** Scenario with 1 FS, and 1 MR which is located at an arbitrary initial position, and has to visit 2 waypoints.

needed to determine if the MR goes directly to the next waypoint or via the FS, this implies that the prediction horizon should be at least 2 waypoints. If the prediction horizon is set to 1 waypoint,  $D$  contains a single element, which is the travel distance from the MR's current position to the FS.  $X$  will be a single decision variable, and due to the constraint of (2-11) this variable will always be 1. This means that during the first fuel cycle, the MR will always travel towards the FS. Now for a second fuel cycle the same scenario arises, only now the MR is located at the FS. Since  $X$  will be 1, the decision will be that the MR should go to the FS. This means that the MR will be stuck at the FS after the first fuel cycle. This can be prevented by choosing the prediction horizon to be at least 2 waypoints. In that case the MR always has the choice of either going to the FS, or to the next waypoint. Furthermore to prevent that a MR stays at a FS when being completely refuelled, a constraint is defined in

(2-21), that guarantees that a MR always travels to the next waypoint after a refuel event.

$$\sum_{q=1}^{M^2} X_{inqk} = 0, \quad i = 1, n = 1, \forall k \in \mathcal{K} \mid E_k = E_{k,\max} \quad (2-21)$$

Since the optimization problem is solved each time an event occurs, the MRs do not always start with a maximum energy level. The constraint of (2-10) has to be adjusted such that the amount of energy consumed by a MR during a fuel cycle is not allowed to be larger than its current energy level, given by (2-22).

$$E_{inqk}^c X_{inqk} \leq E_k, \quad i = 1, \dots, N, n \in \mathcal{N}, q \in \mathcal{Q}, k \in \mathcal{K} \quad (2-22)$$

Here  $E_k$  is the current energy level of  $\text{MR}_k$ .

The following optimization problem is solved, when using the receding horizon principle:

$$\min_{X,U} T_{\text{mission}} \quad (2-23a)$$

$$\text{s.t. (2-11), (2-12), (2-19), (2-21), and (2-22)} \quad (2-23b)$$

In the next section simulation results are presented, and a comparison is made between the optimal approach, Heuristic 1, and the receding horizon principle.

## 2-3 Simulation Results and Discussion

The problems were solved using GUROBI on a laptop computer running on a Windows® 8.1 Enterprise 64 bits operating system, with Intel(R)Core(TM)i7 CPU Q720, 1.6 GHz and 4.00 GB RAM.

### 2-3-1 Simulation Setup

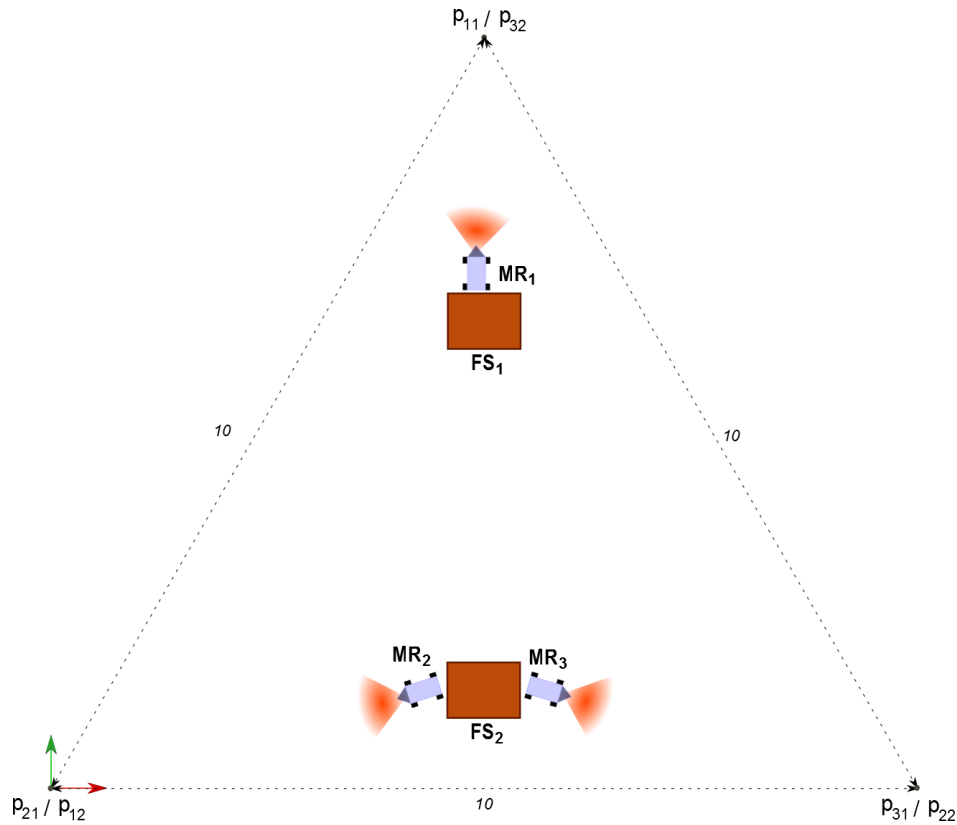
Two different environments were created. In the first environment the waypoints are chosen such that the mission can be seen as a patrol mission, where the MRs have to patrol a certain area. The second environment contains randomly placed waypoints.

Both environments contained:

- 2 FSs:  $\{\text{FS}_1, \text{FS}_2\}$ , located at  $(5, 5)$ , and  $(5, 1)$  respectively. The properties of the FSs were:
  - $v_{m,\max} = 0, \forall m \in \mathcal{M}$ ;
  - $\dot{E}_m \in \{1, 2\}, m \in \mathcal{M}$ .
- 3 MRs, with the following properties:
  - $E_{k,\max} \in \{15, 20, 25, 30\}, k \in \mathcal{K}$ ;
  - $v_{k,\max} \in \{1, 2, 3\}, k \in \mathcal{K}$ ;
  - $\dot{E}_k = 1, \forall k \in \mathcal{K}$ .

### 2-3-2 Simulation 1: Patrol Mission

The first mission that is simulated can be seen as a patrol mission, where the environment contains 2 FSs, and 3 MRs which have to visit two waypoints in a pre-determined order, such that a certain area will be patrolled. The patrol mission is illustrated in Figure 2-5. The distances between two waypoints is 10 distance units and are shown in italics. In total 30



**Figure 2-5:** Patrol mission, environment with 2 FSs, and 3 MRs which all have to visit 2 waypoints, placed at a fixed distance from each other.

trials were performed, where the energy levels of the MRs varied between 15-30 energy units in intervals of 5. The velocities of the MRs were varied between 1-3 distance units per time unit, in intervals of 1. The energy donation rates of the FSs were varied between 1, and 2 energy units per time unit. The prediction horizon of the receding horizon principle was set to 2 waypoints. Table 2-5 shows the average total mission time for the optimal approach, Heuristic 1, and receding horizon principle. From this table it can be observed that Heuristic 1

**Table 2-5:** Average mission times of all centralized approaches, for the patrol mission.

|                      | Optimal approach | Heuristic 1 | Receding horizon |
|----------------------|------------------|-------------|------------------|
| $T_{\text{mission}}$ | 37.64            | 38.04       | 40.43            |

performs very well during the patrol mission. On average Heuristic 1 takes 0.4 times units more than the optimal approach, which is equal to an increase of 1.1 %. The total mission time of the receding horizon principle takes on average 2.79 time units more than the optimal

approach, which is an increase of 7.4%. That Heuristic 1 outperforms the receding horizon principle, is not surprising. Heuristic 1 schedules the refuelling activities of all MRs, for the whole mission. The receding horizon principle schedules the refuelling activities for all MRs, up to the prediction horizon. Since the prediction horizon was set to 2 waypoints, it means that the refuel activities are scheduled while looking one waypoint ahead. Since the receding horizon principle schedules the refuel activities based on less knowledge compared to Heuristic 1, it makes sense that its solution quality is worse.

One trial is highlighted to illustrate which actions of Heuristic 1, and the receding horizon principle can lead to a suboptimal solution. For the scenario with  $E_{k,\max} = \{20, 25, 15\}$ , and  $v_{k,\max} = \{3, 1, 2\}$ , for  $MR_k$ ,  $k = \{1, 2, 3\}$  respectively, and  $\dot{E}_m = \{2, 1\}$ , for  $FS_m$ ,  $m = \{1, 2\}$  respectively, the total mission time of all three approaches were different. The refuel orderings chosen by the approaches are given in Table 2-6. It can be observed that the refuel ordering of each approach is different, which possibly influences the total mission time. Table 2-7 shows all

**Table 2-6:** Comparison of the refuel ordering, chosen by the optimal approach, Heuristic 1, and the receding horizon principle.

|                  | Refuel ordering                                        |
|------------------|--------------------------------------------------------|
| Optimal approach | $MR_{31}, MR_{32}, MR_{21}, MR_{22}, MR_{11}, MR_{12}$ |
| Heuristic 1      | $MR_{11}, MR_{31}, MR_{21}, MR_{12}, MR_{32}, MR_{22}$ |
| Receding horizon | $MR_{11}, MR_{31}, MR_{12}, MR_{32}, MR_{21}, MR_{22}$ |

routes, and arrival, wait, and end times of each fuel cycle, of all MRs, for all three approaches. This table shows that  $MR_2$  takes the same route for all approaches. Each approach found a different route for  $MR_1$ . The routes of  $MR_3$  are the same for the optimal approach, and the receding horizon principle. When observing Tables 2-6 and 2-7, the following conclusions can be drawn. The route found by the optimal approach, is chosen in such a way that none of the MRs has to wait when reaching a FS. Every time a MR reaches a FS, the MR that was previously using it already left. The routes found by Heuristic 1 makes  $MR_3$  wait twice, because  $MR_1$  is already using the FS. The first fuel cycle  $MR_3$  only has to wait 0.2 time units, the second fuel cycle however  $MR_3$  has to wait 4.38 time units. The reason why the receding horizon principle takes longer, is that the routes are chosen in such a way that only  $FS_1$  is used for refuelling. This causes that during its second fuel cycle  $MR_1$  has to wait 3.07 time units because  $MR_3$  is using  $FS_1$ . In turn  $MR_3$  has to wait 3.29 time units during its second fuel cycle, until  $MR_1$  is done refuelling. Eventually  $MR_2$  can start refuelling after waiting 2.82 time units on  $MR_3$ .

### 2-3-3 Simulation 2: Randomly Placed Waypoints

During the second simulation a mission is considered where the environment contains again two FSs, and 3 MRs which have to visit 2 waypoints. The waypoints are randomly placed in an area of  $10 \times 10$  distance units, with the restriction that the euclidean distance between two consecutive waypoints should be at least 7.5 distance units.  $FS_1$ , and  $FS_2$  are located at the coordinates (5,5), and (5,1) respectively. The same trials were performed as during Simulation 1. In total 30 trials were performed, where the energy levels of the MRs varied between 15-30 energy units in intervals of 5. The velocities of the MRs were varied between 1-3 distance units per time unit, with intervals of 1. The energy donation rates of the FSs

**Table 2-7:** Comparison of the routes, found by the optimal approach, Heuristic 1, and the receding horizon principle.

|                  | MR | Fuel cycle | Start location  | Visited waypoint(s) | End location    | Arrival time | Wait time | End time |
|------------------|----|------------|-----------------|---------------------|-----------------|--------------|-----------|----------|
| Optimal approach | 1  | 1          | FS <sub>1</sub> | $p_1, p_2$          | FS <sub>2</sub> | 6.35         | -         | 25.30    |
|                  |    | 2          | -               | -                   | -               | -            | -         | -        |
|                  | 2  | 1          | FS <sub>2</sub> | $p_1, p_2$          | FS <sub>1</sub> | 22.17        | -         | 33.26    |
| Heuristic 1      | 1  | 1          | FS <sub>1</sub> | $p_1$               | FS <sub>1</sub> | 2.50         | -         | 6.29     |
|                  |    | 2          | FS <sub>1</sub> | $p_2$               | FS <sub>2</sub> | 10.35        | -         | 22.52    |
|                  | 2  | 1          | FS <sub>2</sub> | $p_1, p_2$          | FS <sub>1</sub> | 22.17        | -         | 33.26    |
| Receding horizon | 1  | 1          | FS <sub>1</sub> | $p_1$               | FS <sub>1</sub> | 2.52         | -         | 6.29     |
|                  |    | 2          | FS <sub>1</sub> | $p_2$               | FS <sub>1</sub> | 11.07        | 3.07      | 21.21    |
|                  | 2  | 1          | FS <sub>2</sub> | $p_1, p_2$          | FS <sub>1</sub> | 22.17        | 2.82      | 36.07    |
|                  | 3  | 1          | FS <sub>1</sub> | $p_1$               | FS <sub>1</sub> | 7.07         | -         | 14.14    |
|                  |    | 2          | FS <sub>1</sub> | $p_2$               | FS <sub>1</sub> | 17.92        | 3.29      | 24.99    |

were varied between 1, and 2, energy units per time unit. Table 2-8 shows the average total mission time for the optimal approach, Heuristic 1, and receding horizon principle. From this table it can be observed that Heuristic 1 performs again very well. On average Heuristic 1 takes 0.14 times units more than the optimal approach, which is equal to an increase of 0.5%. The total mission time of the receding horizon principle takes on average 2.99 time units more than the optimal approach, which is an increase of 9.9%. Again Heuristic 1 outperforms the receding horizon principle.

**Table 2-8:** Average mission times of all centralized approaches, for the mission with randomly placed waypoints.

|                      | Optimal approach | Heuristic 1 | Receding horizon |
|----------------------|------------------|-------------|------------------|
| $T_{\text{mission}}$ | 30.18            | 30.32       | 33.17            |

#### 2-3-4 Comparison Between Optimal Approach, Heuristic 1, and Receding Horizon Principle

The average total mission times, taken over the patrol mission, and the mission with randomly placed waypoints are given in Table 2-9. The mean error of Heuristic 1, and the receding horizon principle with respect to the optimal approach is denoted as  $\mu$ , and given in percentage. From this table it can be observed that on average the solution found by Heuristic 1 is

0.8 % from the optimum. Compared to the receding horizon principle which solution is on average 8.4 % from the optimum, the solution quality of Heuristic 1 is very high.

**Table 2-9:** Average mission times of all centralized approaches, calculated over the patrol mission, and mission with randomly placed waypoints.

| Optimal approach     | Heuristic 1          |       | Receding horizon     |       |
|----------------------|----------------------|-------|----------------------|-------|
| $T_{\text{mission}}$ | $T_{\text{mission}}$ | $\mu$ | $T_{\text{mission}}$ | $\mu$ |
| 33.91                | 34.18                | 0.8 % | 36.76                | 8.4 % |

Simulations were performed with an increasing problem scale, in order to evaluate computation times, and the problem size that can be solved in reasonable time for both the optimal approach, and Heuristic 1. Here reasonable time is defined as two hours. In Table 2-10 the computation times are presented for a scenario with 2 FSs, and a changing number of MRs, and waypoints. It can be observed that the computation time of Heuristic 1 is significantly smaller compared to the optimal approach. A second important observation can be made about the size of the problems that can be solved in reasonable time. It can be seen that for a scenario with 2 MRs, problems can be solved up to 5, and 16 waypoints, by the optimal approach and Heuristic 1 respectively. For a scenario with 3 MRs, problems can be solved up to 2, and 8 waypoints, by the optimal approach and Heuristic 1 respectively. This is an enormous improvement of Heuristic 1. Of course there is always a trade-off, which is here computation time versus solution quality. Heuristic 1 is lots faster, compared to the optimal approach, but the solution is on average 0.8 % from the optimum as shown in Table 2-9.

**Table 2-10:** Comparison of the computation times, between Heuristic 1, and the optimal approach, for a scenario with 2 FSs.

| # MRs | # Waypoints | Optimal approach | Heuristic 1 |
|-------|-------------|------------------|-------------|
| 2     | 1           | 0.03 s           | 0.03 s      |
| 2     | 2           | 0.11 s           | 0.08 s      |
| 2     | 3           | 1.75 s           | 0.21 s      |
| 2     | 4           | 54.26 s          | 0.42 s      |
| 2     | 5           | 6068.3 s         | 1.72 s      |
| 2     | 6           | N/A              | 1.93 s      |
| 2     | 7           | N/A              | 4.36 s      |
| 2     | 8           | N/A              | 11.49 s     |
| 2     | 9           | N/A              | 11.17 s     |
| 2     | 10          | N/A              | 11.17 s     |
| 2     | 11          | N/A              | 80.75 s     |
| 2     | 12          | N/A              | 102.29 s    |
| 2     | 13          | N/A              | 85.29 s     |
| 2     | 14          | N/A              | 175.85 s    |
| 2     | 15          | N/A              | 653.94 s    |
| 2     | 16          | N/A              | 914.91 s    |
| 2     | 17          | N/A              | N/A         |
| 3     | 1           | 0.34 s           | 0.09 s      |
| 3     | 2           | 31.04 s          | 0.74 s      |
| 3     | 3           | N/A              | 5.85 s      |
| 3     | 4           | N/A              | 11.00 s     |
| 3     | 5           | N/A              | 23.12 s     |
| 3     | 6           | N/A              | 59.62 s     |
| 3     | 7           | N/A              | 242.81 s    |
| 3     | 8           | N/A              | 3560.6 s    |
| 3     | 9           | N/A              | N/A         |

The # sign stands for ‘total number of’, i.e. # waypoints means ‘total number of waypoints’.

## 2-4 Conclusion

In this chapter a centralized approach is discussed to schedule the refuelling activities of multiple heterogeneous MRs. In Section 2-1 a MILP was formulated for fixed FSs scenarios, using a distance-based metric. This formulation formed the basis for the final formulation where a time-based metric is considered.

Section 2-2 presents the formulation of the MILP for fixed FSs scenarios, using a time-based metric. By solving this MILP global optimal solutions were found. Since the computation time increases very quickly when the problem size increases, a solution can only be found in reasonable time for very small scale scenarios. A heuristic has been proposed, in order to speed up the computation time. A significant reduced computation time was observed, and the size of the problems that can be solved in reasonable time increased tremendously. On average

the solution of this heuristic turned out to be close to the optimal solution. The size of the problems that can be solved in reasonable time is among others limited by the total number of waypoints. In order to find a solution for problems with a large number of waypoints, a receding horizon principle was proposed. When using this principle, the optimization problem is repeatedly solved for a number of waypoints that is equal to the prediction horizon. The solution quality of the receding horizon principle is suboptimal, and is worse compared to the heuristic.

Simulation results are presented and discussed in Section 2-3. The performances of the centralized approaches were also compared in terms of the total mission time. The optimal approach, and the heuristic were compared based on the computation time, and the size of the problems that can be solved in reasonable time. Which method is best suited, depends on the scenario, and the performance requirements. In general there is a trade-off between the solution quality, and the computation time.

The main contributions of this chapter are: the formulation of a MILP for scheduling the refuelling activities of multiple MRs across multiple FSs using a time-based metric, and the implementation of the receding horizon principle. When using a time-based metric, the duration each MR spends refuelling and order in which the robots are allowed to enter a FS have to be taken into account. This is often neglected in literature. The next chapter will discuss a distributed approach to schedule the refuelling activities.

# Distributed Approach

In this chapter a distributed approach is discussed, in order to solve the problem formulated in Section 1-1. The presented approach is called distributed, because the mobile robots (MRs) make individual decisions based on local knowledge. There is only limited communication between the fuelling stations (FSs), and the MRs. FSs are not able to communicate with other FSs, and MRs are not able to communicate with other MRs. Advantages of a distributed architecture are that it is typically very fast [36], adaptive, and robust to failures [35, 36]. A disadvantage of this architecture is that the solutions are suboptimal [35, 36]. A distributed architecture is best suited for scenarios where large robotic teams, perform relative simple tasks, with no requirements for efficiency [36].

### 3-1 Refuel Event Selection

In recent literature there are several methods presented to select when a MR should go for refuel, when using a distributed architecture. The basic method makes use of a fixed threshold on the energy level [40, 41]. An extension of this strategy works with an adaptive energy threshold value [8]. Several other methods are proposed, such as self calibration [42], time discounted labour [9], opportunism [39], and motives and artificial emotions [43]. In this section the fixed, and adaptive threshold methods are discussed, which have been implemented for the proposed distributed approach.

#### 3-1-1 Fixed Threshold

This strategy makes use of a fixed threshold  $E_\gamma$  on the energy level of the MRs. Each MR will go for refuel when its energy level drops below the threshold:  $E < E_\gamma$ . The threshold value has to be chosen carefully, in order to guarantee that the MRs do not deplete. If the threshold value is chosen too high, the MRs can become inefficient because they spent a lot of time on travelling to and from the FSs. The threshold value should be chosen such that the MR can always cross the largest possible distance to a FS, if it has to be guaranteed that

the MRs do not run out of energy. Consider an environment  $A$ , with a single fixed fuelling station  $FS_m$ . The threshold value  $E_{\gamma_k}$  of  $MR_k$  for this example can be calculated as follows:

$$E_{\gamma_k} = \dot{E}_k \sqrt{(\max(x_A) - x_m)^2 + (\max(y_A) - y_m)^2}, \quad (3-1)$$

where  $\max(x_A)$ , and  $\max(y_A)$  determine the values of the  $x$ , and  $y$  locations respectively, corresponding to the point in  $A$  that is located at the farthest possible distance from  $FS_m$ . This implies that it is not desired to use one threshold value for all environments. The threshold value should be determined every time the environment changes, in order to guarantee that a MR does not run out of energy, and to obtain a certain efficiency level.

### 3-1-2 Adaptive Threshold

This method is an extension of the fixed threshold strategy. Wawerla and Vaughan [9] proposed this strategy for a single FS scenario as follows. Each time a MR reaches a waypoint, it calculates if its energy level is sufficient to reach the next waypoint, and the FS afterwards. If the energy level is sufficient, the MR will go to the next waypoint directly. Otherwise the MR will travel via the FS. Since the problem considered in this thesis consists of multiple FSs, this strategy can not be applied directly. Here an adjustment to this adaptive threshold strategy, such that it can be applied to scenarios with multiple FSs is proposed as follows. Each time a MR reaches a waypoint it calculates if its energy level is sufficient to reach the next waypoint, and the closest fixed FS afterwards. If the energy level is sufficient, the MR will go to the next waypoint directly. Otherwise the MR will travel via the closest FS. This strategy requires the locations of the fixed FSs to be known by the MRs. This can be realized by communication between the FSs, and the MRs. Here the locations of the fixed FSs are communicated as follows: every time a MR reaches a waypoint it broadcasts a “Waypoint” message, and the fixed FSs will respond by sending their positions.

When a  $MR_k$  reaches a new waypoint  $p_i$ , the energy threshold is calculated as given by (3-2).

$$E_{\gamma} = \dot{E}_k \left( \sqrt{(x_c - x_{p_i})^2 + (y_c - y_{p_i})^2} + \sqrt{(x_c - x_{p_{i+1}})^2 + (y_c - y_{p_{i+1}})^2} \right) \quad (3-2)$$

Here  $x_c, y_c, x_{p_i}$ , and  $y_{p_i}$  are the  $x$ , and  $y$  positions of the closest FS, and waypoint  $p_i$  respectively.

The benefits of the fixed, and adaptive threshold methods are that they are not computationally complex, and relative easy to implement. Furthermore when implemented carefully, these methods can be used to guarantee that the MRs do not deplete. The advantage of using an adaptive threshold instead of a fixed threshold, is that the efficiency (in terms of total travel distance) of the MR is larger. Wawerla and Vaughan [9] already showed this for a single FS, and single MR scenario. A drawback of the adaptive threshold method is that the communication demands are higher compared to the fixed threshold strategy. Simulations were performed to evaluate if the adaptive threshold strategy is also beneficial in terms of total mission time, compared to a fixed threshold, for scenarios with multiple FSs, and multiple MRs. The simulation results are presented in Section 3-4.

## 3-2 Fuelling Station Allocation

When a MR is in need of refuel, it has to be assigned to a FS. The FS should be chosen in such a way that eventually the total mission time is minimized. For a distributed architecture, the FS allocation principle is proposed as follows. When a MR is in need of refuel, it broadcasts an “Energy Low” message together with its: location, maximum velocity, next waypoint location, energy level, maximum energy level, and energy consumption rate. Based on this information a FSs receiving the message, will calculate how long it will take for the MR to reach the FS and the next waypoint afterwards. This travel time is indicated as  $T^t$  and is calculated for  $MR_k$ , and  $FS_m$  as follows:

$$T_{km}^t = \left( \sqrt{(x_m - x_k)^2 + (y_m - y_k)^2} + \sqrt{(x_m - x_{p_{k,next}})^2 + (y_m - y_{p_{k,next}})^2} \right) / v_{k,max}, \quad (3-3)$$

where  $x_m, y_m, x_k, y_k, x_{p_{k,next}}$ , and  $y_{p_{k,next}}$  are the x, and y positions of  $FS_m$ ,  $MR_k$ , and the next waypoint of  $MR_k$  respectively. Besides these values, the FS also calculates:

$E_{km}^f$  - The amount of energy  $MR_k$  needs to reach  $FS_m$ , from its current position;

$E_{km}^p$  - The amount of energy  $MR_k$  needs to reach the next waypoint, from the location of  $FS_m$ .

These energy values are calculated as given by (3-4), and (3-5).

$$E_{km}^f = \dot{E}_k \sqrt{(x_m - x_k)^2 + (y_m - y_k)^2} \quad (3-4)$$

$$E_{km}^p = \dot{E}_k \sqrt{(x_m - x_{p_{k,next}})^2 + (y_m - y_{p_{k,next}})^2} \quad (3-5)$$

The FSs send the value of  $T^t$  to the MR, which will compare all received travel times. Since the objective is to minimize the total mission time, the MR always travels to the FS corresponding to the shortest travel time. If this FS is occupied, the MR enters the queue and sends a “Wait” message to inform the FS that it is waiting for refuel. If multiple MRs are waiting, the robots are allowed to enter the FS on a first come first served basis. There are two scenarios in which it is undesired for  $MR_k$  to go for refuel at  $FS_m$ :

- $E_{km}^f > E_k$ , the current energy level of  $MR_k$  is insufficient to reach  $FS_m$ ;
- $E_{km}^p > E_{k,max}$ , the maximum energy level of  $MR_k$  is insufficient to reach the next waypoint, from the location of  $FS_m$ .

In the former case, the MR will deplete when it tries to reach the FS. In the latter case there is the risk of the MR travelling between the FS and the next waypoint without reaching it. If one of these scenarios arises, the travel time is set to infinity  $T^t = \infty$ . The FSs for which at least one of these scenarios occurs, are defined as unreachable FSs. Since the travel time towards the unreachable FSs is set to infinity, the MR will always travel to a FS for which none of the undesired scenarios occurred. This can cause situations where some FSs are free, while there are multiple MRs waiting for another FS to become free. At least depletion of the MRs is prevented. One could still think of problems where it can occur that all of the FSs,

are unreachable. Consider a single FS, single MR scenario, where the MR starts its mission at the FS, and the first waypoint cannot be reached by the MR with a maximum initial energy level. This will cause the MR to deplete, since it can never reach the first waypoint. This problem is infeasible, since there is no valid solution. If for a certain problem definition a situation occurs, where one of the MRs gets depleted, the problem is infeasible. In that case one should reconsider the threshold value (in case of using a fixed threshold), the placement of the FSs, the energy capacity of the MR, or the locations of the waypoints.

### 3-3 Mobile Fuelling Stations

The operation time and range of the MRs can be increased by using mobile fuelling stations (MFSs). When minimizing the total mission time, the use of MFSs has a number of advantages. Since the MFS can move towards the MRs, a MR can:

- Reach the refuel location faster;
- Save energy, such that it takes less time to be completely refuelled.

The only difference between a fixed FS, and a MFS is its maximum velocity. If  $v_{m,\max} = 0$ , the FS is fixed, and if  $v_{m,\max} > 0$  the FS is mobile.

#### 3-3-1 Rendezvous Strategy

In order for the MFS to be able to refuel a MR, they should meet at a rendezvous location. The rendezvous locations are defined in Chapter 1 as  $f_{mlk}$ ,  $m \in \mathcal{M}, k \in \mathcal{K}$ . Here  $l$  is the index of the  $l^{\text{th}}$  refuelling task of the FS. In recent literature there are several methods proposed for a MR to rendezvous with a MFS. Drenner and Papanikolopoulos [44] propose a method to relocate the position of a FS to maximize the longevity of a cluster of MRs. The position of the FSs is adjusted such that the average travel distance of all MRs in the cluster is minimized. Marmol et al. [10] propose a rendezvous strategy to minimize the total travel distance of an individual MR. If a MR is in need of refuel, the FS calculates a rendezvous location such that the travel distance of that single MR is minimized. The locations of the other MRs are not taken into account. This strategy has proven to be beneficial. The rendezvous strategy proposed in this section is inspired by the method of Marmol et al. [10]. Instead of choosing the rendezvous locations such that the travel distance of an individual MR is minimized, the rendezvous locations are determined in such a way that the total travel time of the MR is minimized. The rendezvous strategy is discussed below.

As soon as  $FS_m$  receives an “Energy Low” message from  $MR_k$  it determines a rendezvous location  $f_{mlk}$ , by minimizing  $T_{mlk}^t$ . This is the total time it takes before both the FSs and the MR will arrive at  $f_{mlk}$  plus the time the MR needs to travel to the next waypoint location afterwards.  $T_{mlk}^t$  is calculated as follows:

$$\begin{aligned}
 T_{mlk}^t = & \max \left( \sqrt{(x_m - x_{f_{mlk}})^2 + (y_m - y_{f_{mlk}})^2} / v_{m,\max}, \right. \\
 & \left. \sqrt{(x_k - x_{f_{mlk}})^2 + (y_k - y_{f_{mlk}})^2} / v_{k,\max} \right) + \\
 & + \sqrt{(x_{f_{mlk}} - x_{p_{k,\text{next}}})^2 + (y_{f_{mlk}} - y_{p_{k,\text{next}}})^2} / v_{k,\max},
 \end{aligned} \tag{3-6}$$

where  $x_m, y_m, x_k, y_k, x_{f_{mlk}}, y_{f_{mlk}}, x_{p_{k,next}},$  and  $y_{p_{k,next}}$  are the x, and y positions of FS<sub>m</sub>, MR<sub>k</sub>,  $f_{mlk}$ , and the next waypoint of MR<sub>k</sub> respectively.

The rendezvous location is determined by solving an optimization problem, such that  $T_{mlk}^t$  is minimized. The decision variables are the x and y positions of the rendezvous location:  $x_{f_{mlk}}$ , and  $y_{f_{mlk}}$ . With these decision variables, the optimization problem is formulated as follows:

$$\min_{x,y} T^t \quad (3-7)$$

This is an unconstrained convex optimization problem, since the objective function consists of only convex terms. There are several optimization solvers which can be used to solve these problems, for instance the state of the art solvers include CPLEX and GUROBI.

After determining the rendezvous location, the FS sends the value of  $T^t$  to the MR that had send the “Energy Low” message. The MR will compare all received travel times, and will go for refuel at the FS corresponding to the shortest travel time. The MR will send a “Coming” message to the chosen FS, and will travel towards the rendezvous location. If the FS is not occupied at the moment, it will travel towards the rendezvous location after receiving the “Coming” message. Refuelling will start as soon as both the FS and the MR reached the rendezvous location. In case the FS is occupied, it saves a list with all rendezvous locations of the MRs that had send a “Coming” message. In that case the FS will visit these rendezvous locations on a first comes first served basis.

### 3-3-2 Reformulation of the Distributed Approach

Simulation case studies involving a MFS, showed two remarkable results. It was observed that sometimes the total mission time is shorter, for scenarios with two fixed FSs, compared with scenarios with one fixed, and one mobile FS. Another observation was that the total mission time increased, when the velocity of the MFS increased. Both observations are counter intuitive. It was expected that the use of MFSs would decrease the total mission time. Since the MFSs are able to move towards the MRs, the MRs should: need less time to refuel, and reach a refuel location faster. During the simulations, a logical phenomena was observed. The faster a FS can travel, the closer the calculated rendezvous location will be to the current location of a MR. This implies the calculated rendezvous location is often closer to the MR’s current location, than the location of the fixed FS. Since the MRs always go to the FS that corresponds to the shortest travel time, the rendezvous location of the MFS is often picked as the best refuel option. This causes a lot of times a large number of MRs are waiting for the MFS to reach the rendezvous location, while it is currently refuelling another MR. In the meanwhile the fixed FS is often unused because the most of the MRs chose to refuel at the MFS. Because the MFS is dominating the fixed FS, the waiting times of the MRs increases, which results in a larger total mission time. A method to prevent that a MFS is dominant over the fixed FSs, is to take the following times into account:

- $T_{mlk}^r$  - Time MR<sub>k</sub> needs to be completely refuelled by FS<sub>m</sub>, at rendezvous location  $f_{mlk}$ ;
- $T_{mk}^o$  - Time FS<sub>m</sub> currently needs to completely refuel MR<sub>k</sub>;
- $T_{mk}^w$  - Time MR<sub>k</sub> currently has to wait in the queue, when choosing to refuel at FS<sub>m</sub>.

The calculation of  $T_{mlk}^r$  is given by (3-8).

$$T_{mlk}^r = \frac{E_{k,\max} - (E_k - \dot{E}_k \sqrt{(x_k - x_{f_{mlk}})^2 + (y_k - y_{f_{mlk}})^2})}{\dot{E}_m} \quad (3-8)$$

The optimization problem of the rendezvous strategy is extended with  $T^r$ . The rendezvous location now also depends on the energy donation rate of the FS. If the FS has a small energy donation rate, it is beneficial if the rendezvous location is chosen close to the current location of the MR, such that the MR saves energy, and needs less time to refuel. On the other hand if the FS has a large energy donation rate, but a slow maximum velocity, it is beneficial if the rendezvous location is chosen close to the current location of the FS. By including  $T^r$  in the optimization problem, the rendezvous location is optimized in terms of velocities of both the FS, and MR, and the energy donation, and consumption rates of the FS, and MR respectively. The optimization problem is given by (3-9). Again this is an unconstrained convex optimization problem, since the objective function consists of all convex terms.

$$\min_{x,y} T^t + T^r \quad (3-9)$$

The times  $T_{mk}^o$ , and  $T_{mk}^w$  do not depend on the MR that had just send the “Energy Low” message, but on the MR that is currently using the FS, and the MRs that are waiting in the queue respectively. The calculation of these times requires the current energy levels of the involved MRs have to be exchanged with the FS. The MRs in the queue have to communicate with the FS to send their energy levels. For the MR that is currently using the FS, this can either be done by communication, or by the FS measuring the energy level.  $T_{mk}^o$ , and  $T_{mk}^w$  are calculated as given in (3-10), and (3-11).

$$T_{mk}^o = \frac{(E_{k,\max} - E_k)}{\dot{E}_m}, \quad (3-10)$$

where the index  $k$  represents the MR that is currently using FS <sub>$m$</sub> .

$$T_{mk}^w = \sum_{k=1}^W \frac{(E_{k,\max} - E_k)}{\dot{E}_m}, \quad (3-11)$$

where  $W$  is the total number of MRs that are waiting, and the index  $k$  indicates the MRs in the queue.

Instead of only sending the travel time to the MR that had send the “Energy Low” message, the FSs now send the total time value  $\mathbb{T}$ , which is calculated as follows:

$$\mathbb{T}_{mlk} = T_{mlk}^t + T_{mlk}^r + T_{mk}^o + T_{mk}^w, \quad m \in \mathcal{M}, l \in \mathcal{L}, k \in \mathcal{K} \quad (3-12)$$

The MR will compare all received  $\mathbb{T}$  values, and goes for refuel at the FS corresponding to the lowest value. Since the waiting times are taken into account, the MRs will often choose to refuel at a different FS, if the waiting time of the MFS is too long. In this way it is prevented that MFSs dominate the fixed FSs, and the fixed FSs stay unused. The simulation results that initiated this reformulation are presented in the next section.

## 3-4 Simulation Results and Discussion

Several simulations were performed, in order to evaluate: the performance of the fixed, and adaptive threshold methods, and the benefits of MFSSs. Simulations were performed on a laptop computer running on a Windows® 8.1 Enterprise 64 bits operating system, with Intel(R)Core(TM)i7 CPU Q720, 1.6 GHz and 4.00 GB RAM. The optimization problem of (3-7) has been formulated using YALMIP and is solved by GUROBI. It takes GUROBI around 9.5 ms to solve this problem.

### 3-4-1 Simulation Environment

A simulation environment is created using object oriented programming in MATLAB. An object is created for each MR, and FS.

Each MR object has the following properties:

- Position (x, and y position);
- Maximum velocity;
- Energy capacity;
- Energy consumption constant (determines the amount of energy units the MR will consume per travel distance);
- Status (Visiting Waypoints, Going for Refuel, Waiting, Refuelling).

Each  $MR_k$  has the following inputs:

- Initial position (x, and y position);
- Set of  $N$  waypoints  $\{p_{k1}, p_{k2}, \dots, p_{kN}\}$ ;
- Communication input.

Each MR also has a communication output.

Each FS object has the following unique properties:

- Position (x, and y position);
- Maximum velocity;
- Energy donation rate;
- Status(Free, Occupied, Refuelling, Moving).

Each FS has the following inputs:

- Initial position (x, and y position);
- Communication input.

Each FS also has a communication output.

### 3-4-2 Simulation Setup

Five different environments were created, where 10, 20, 30, 40, and 50 waypoints were randomly placed in an area of  $10 \times 10$  distance units, with the restriction that the euclidean distance between two consecutive waypoints should be at least 7.5 distance units.

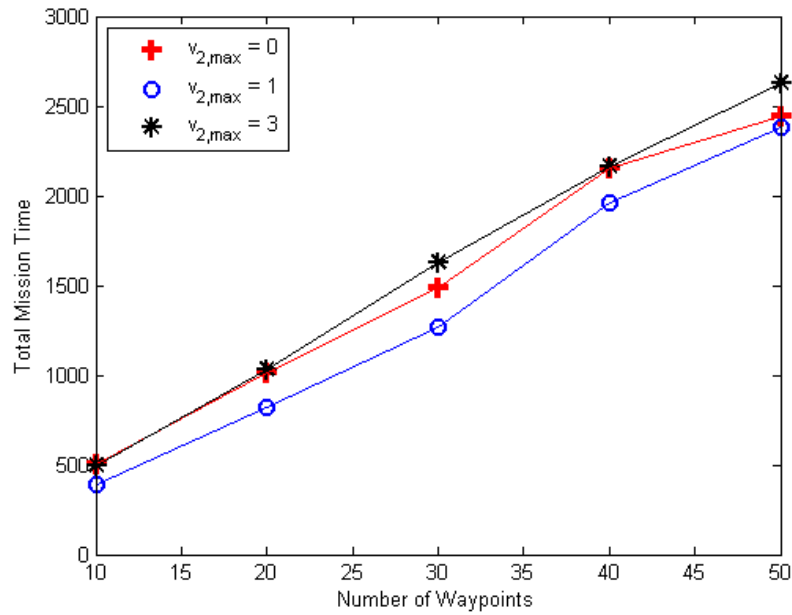
All environments contained:

- 2 FSs:  $\{FS_1, FS_2\}$ , initially located at  $(5, 5)$ , and  $(5, 1)$  respectively. The properties of the FSs were:
  - $v_{m,\max} = 0$ , for  $m = 1$ , and  $v_{m,\max} \in \{0, 1, 3\}$ , for  $m = 2$ ;
  - $\dot{E}_m = 1$ ,  $\forall m \in \mathcal{M}$ .
- 6 MRs:  $\{MR_1, MR_2, MR_3, MR_4, MR_5, MR_6\}$ , initially located at  $(5, 5)$ ,  $(5, 5)$ ,  $(5, 5)$ ,  $(5, 1)$ ,  $(5, 1)$ , and  $(5, 1)$  respectively. The properties of the MRs were:
  - $E_{\max} = \{20, 20, 25, 25, 30, 30\}$ , for  $\{MR_1, MR_2, MR_3, MR_4, MR_5, MR_6\}$  respectively;
  - $v_{\max} = \{1, 1, 2, 2, 3, 3\}$ , for  $\{MR_1, MR_2, MR_3, MR_4, MR_5, MR_6\}$  respectively;
  - $\dot{E}_k = 1$ ,  $\forall k \in \mathcal{K}$ .

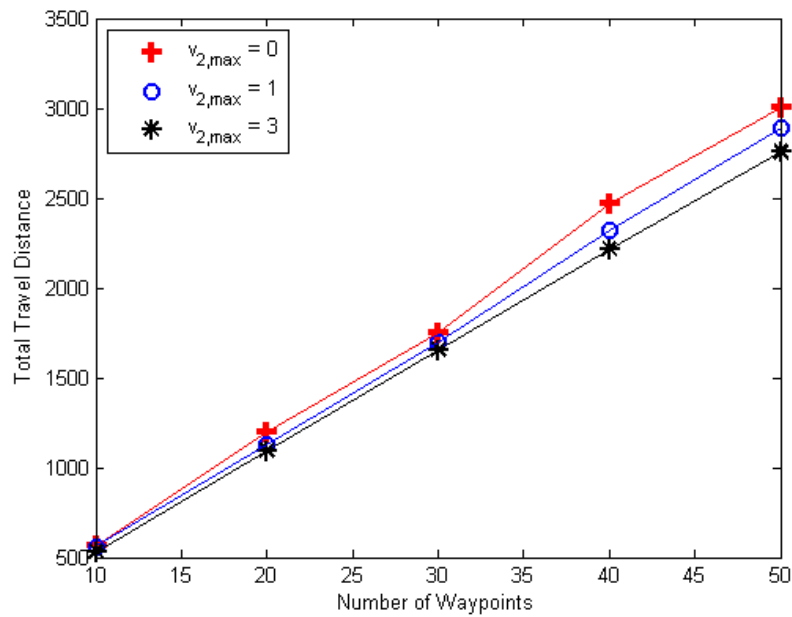
#### Simulation 1: Fixed Threshold

During this simulation, the performance of the fixed threshold strategy was evaluated in terms of total mission time. For all 5 environments, trials were run for three different velocities for  $FS_2$ :  $v_{2,\max} \in \{0, 1, 3\}$ . In total 15 trials were run. The threshold value was calculated using (3-1), and was set to 7.1 energy units for all MRs. This means that the MRs went for refuel as soon as their energy level reached 7.1 energy units. With this threshold value none of the MRs ran out of energy, as expected. The results of this simulation are presented in Figure 3-1. The total mission times are indicated for each trial, with a red plus, blue circle, and black asterisk, for the velocities of  $FS_2$ ,  $v_{2,\max} = 0, 1$ , and 3 respectively. For convenience, the markers are connected with a line in the corresponding color. It is not guaranteed that in between two markers the total mission time is equal to the time indicated by the line, since no simulations were run for those points. Nevertheless there can be observed a trend. Figure 3-1 shows the two remarkable results that are already discussed in Section 3-3. It can be observed that sometimes the total mission time is shorter, for scenarios with two fixed FSs, compared to scenarios where  $FS_2$  is mobile. Another observation is that the mission time of all trials is larger when  $v_{2,\max} = 3$ , compared to  $v_{2,\max} = 1$ . In order to get some more insight Figures 3-2 and 3-3 show the total travel distances during these trials of all MRs, and  $FS_2$  respectively. It can be observed that the previous noted positive effects of a MFS are still there: Since  $FS_2$  is travelling towards the MRs, the total travel distance of the MRs is reduced. This implies that the travel time, and refuel time is reduced. Because  $FS_2$  is dominating  $FS_1$ , the waiting times of the MRs increased which increases the total mission time. Apparently the waiting times increased more than the travel, and refuel times reduced. Because of this the total mission time increased, when a MFS was applied.

The adjustments to prevent that a MFS dominates the fixed FSs, are already discussed in Section 3-3. With these adjustments, the same trials were repeated. The total mission times

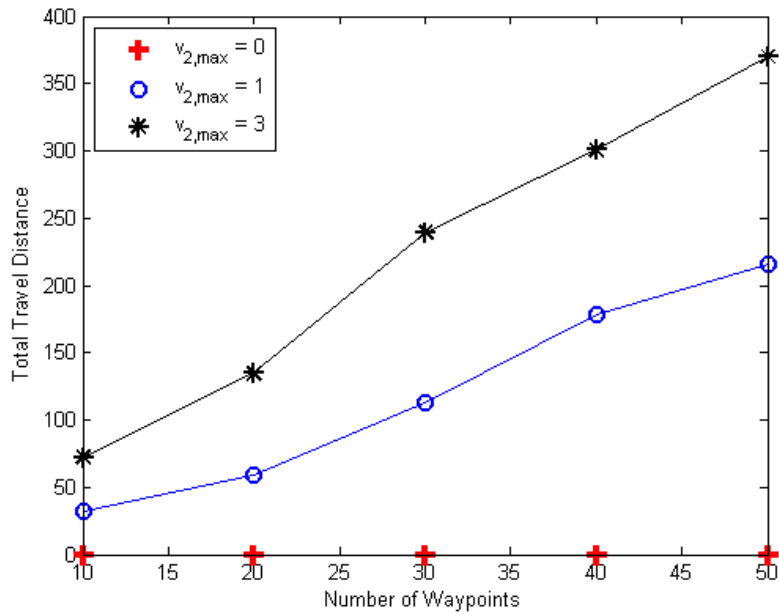


**Figure 3-1:** Total mission time given for a scenario with 2 FSs, and 6 MRs, which have to visit randomly placed waypoints, using a fixed threshold strategy.

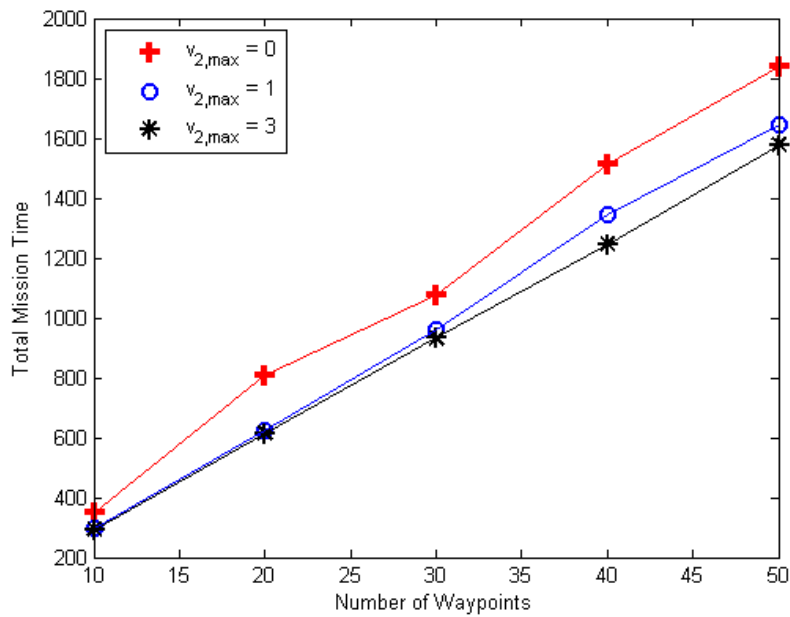


**Figure 3-2:** Total travel distance of all MRs given for a scenario with 2 FSs, and 6 MRs, which have to visit randomly placed waypoints, using a fixed threshold strategy.

of all trials are shown in Figure 3-4. From this figure it can be observed that the use of MFSs is beneficial in terms of total mission time, if the waiting times are taken into account. During all trials where FS<sub>2</sub> was mobile, the total mission time was shorter compared with scenarios



**Figure 3-3:** Total travel distance of  $FS_2$  given for a scenario with 2 FSs, and 6 MRs, which have to visit randomly placed waypoints, using a fixed threshold strategy.



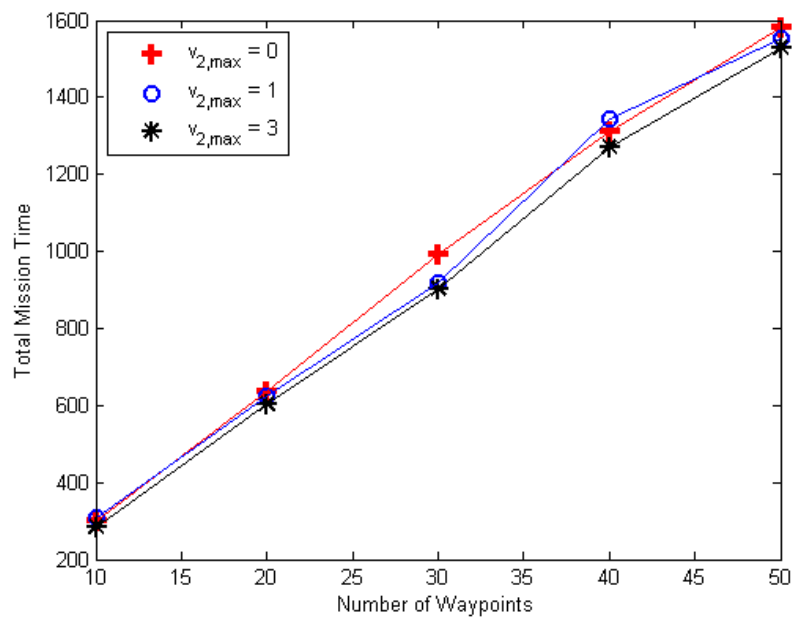
**Figure 3-4:** Total mission time given for a scenario with 2 FSs, and 6 MRs, which have to visit randomly placed waypoints, using a fixed threshold strategy. The waiting times are taken into account.

where both FSs were fixed. These simulation results clearly indicate the difference between the use of a distance, and a time-based metric. A reduction of the total travel distance does

not guarantee a reduced total mission time, and can even lead to an increased total mission time.

### Simulation 2: Adaptive Threshold

The same trials were run as discussed for the fixed threshold, only now for the adaptive threshold strategy. Since during Simulation 3-4-2 it turned out that the use of MFSs is only beneficial in terms of total mission time if the waiting times are taken into account, these are taken into account during these trials as well. The results are presented in Figure 3-5. This figure shows that it is often beneficial to make use of MFSs, except for the scenario

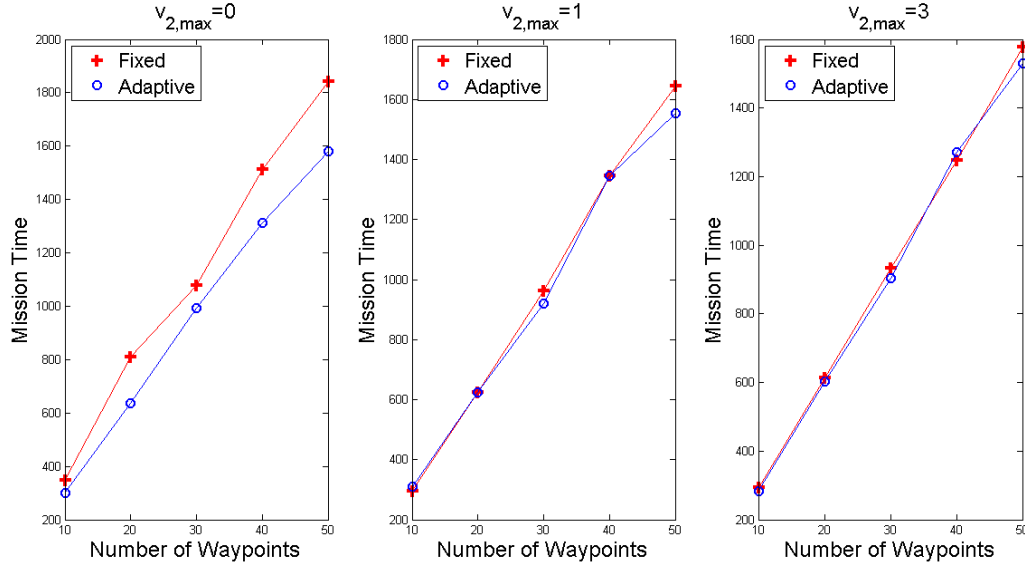


**Figure 3-5:** Simulation results for the adaptive threshold strategy, for a scenario with 2 FSs, and 6 MRs, which have to visit randomly placed waypoints.

with 40 randomly placed waypoints. It is interesting to observe that the use of a MFS is less beneficial for the adaptive threshold strategy compared to the fixed threshold. This can be caused by the property that a MR calculates if it is able to reach the next waypoint, and the closest fixed FS afterwards, when reaching a waypoint. When there are two fixed FSs there is more often a fixed FS available that can be reached after visiting the next waypoint, compared with scenarios with a single fixed FS. Another drawback of not taking the MFSs into account during the calculation of the threshold, is that the adaptive threshold cannot be used for scenarios with only MFSs. The decision to let the MRs only calculate if the closest fixed FS can be reached, is made to guarantee that the MRs do not deplete. If the MFSs are taken into account, it is possible that a MR determined that it can reach a MFS after visiting the next waypoint. While the MR is travelling towards the next waypoint, the MFS can move to a different location which corresponding rendezvous location cannot be reached by the MR. This can cause MRs running out of energy.

### Comparison Between Fixed, and Adaptive Threshold

A comparison between the total mission times of Figures 3-4 and 3-5 for each velocity of  $FS_2$  is shown in Figure 3-6. Wawerla and Vaughan [9], already showed that the adaptive threshold



**Figure 3-6:** Comparison between the fixed, and adaptive threshold strategy, for a scenario with 2 FSs, and 6 MRs, which have to visit randomly placed waypoints.

outperforms the fixed threshold strategy for scenarios with a single FS, using a distance-based metric. The results of Figure 3-6 indicate that the adaptive threshold also outperforms the fixed threshold strategy for scenarios with multiple fixed FSs, using a time-based metric. For the scenarios with one fixed, and one mobile FS the adaptive threshold seems to perform slightly better compared to the fixed threshold. During some trials, the fixed threshold even performed a little better. As discussed before this can occur because the rendezvous locations of the MFSs are not taken into account when calculating the adaptive threshold value. It is expected that the adaptive threshold will also significantly outperform the fixed threshold for scenarios with MFSs, if their rendezvous locations are taken into account.

## 3-5 Conclusion

This chapter discussed a distributed approach for scheduling the refuelling activities of multiple heterogeneous MRs. The approach is called distributed, because each MR makes individual decisions based on local knowledge.

In Section 3-1 two refuel event selection methods are discussed, which make use of a fixed, and adaptive threshold. These strategies have the advantages that they are not computationally complex, and relative easy to implement. When carefully implemented, these methods can guarantee that the MRs do not run out of energy.

Section 3-2 presents the fuelling station allocation strategy. The calculations of this approach

are distributed over the robotic team. This has the advantage that this strategy is not computational demanding.

Then Section 3-3 introduces the use of MFSs, in order to increase the operation time and range of the MRs. A rendezvous strategy has been discussed, which is used to find an optimal rendezvous location for a single MFS, and a single MR.

Finally in Section 3-4 simulation results are presented, and discussed. The performances of the refuel event selection methods were compared, based on total mission time. Also the benefits of MFSs were evaluated. The adaptive threshold method outperforms the fixed threshold, for scenarios with only fixed FSs. For scenarios with one fixed and one mobile FS, the performance of the adaptive threshold seems to be slightly better than the fixed threshold. This difference can be explained by the fact that the rendezvous locations of the MFSs are not taken into account by the adaptive threshold method. When these rendezvous locations are taken into account, it is expected that the total mission time will be reduced.

The main contributions of this chapter are: the implementation of the adaptive threshold for multiple FSs scenarios, and the implementation of MFSs. The combination between fixed and mobile FSs is seldom found in literature. Marmol et al. [10] address this combination, but in their approach each MR has an own fixed FS at a home base. This makes it unnecessary for the MRs to share the FSs. The next chapter discusses a hierarchical approach to schedule the refuelling activities.



# Hierarchical Approach

This chapter discusses a hierarchical approach, in order to solve the problem stated in Section 1-1. The approach is called hierarchical, because decisions are made for individuals or clusters of mobile robots (MRs), based on local knowledge. There is communication possible between the fuelling stations (FSs) and the MRs, and the MRs are able to communicate with each other. This architecture is in the middle of a centralized and distributed architecture. The solutions of a hierarchical approach are suboptimal, but in general better compared to distributed approaches. A hierarchical approach can distribute a lot of the planning and execution over the robotic team, thereby it retains the benefits of distributed approaches regarding to speed, flexibility, and robustness [35, 36]. Nevertheless hierarchical approaches also have their downsides. In scenarios where fully centralized methods are feasible, hierarchical approaches can produce poorer solutions and be more complex to implement. In scenarios where distributed approaches are sufficient, hierarchical approaches can be too complex to implement and have higher communication, and computation demands [36]. In literature hierarchical approaches in the form of a market based approach, are proposed for coordinating the refuelling activities of MRs by Leonard et al. [13], Marmol et al. [10].

The refuel events of the hierarchical approach are selected in the same way as in Chapter 3. A fixed, and adaptive threshold strategy were implemented. The hierarchical approach does not affect the refuel event selection. The main difference between the hierarchical, and distributed approach is that the FSs are reallocated, when multiple MRs want to refuel at the same FS.

### 4-1 Fuelling Station Allocation

The main principle of the FS allocation is the same as for the distributed approach. As soon as a MR becomes in need of refuel, it broadcasts an “Energy Low” message together with its: location, maximum velocity, next waypoint location, energy level, maximum energy level, and energy consumption rate. Based on this information the FSs that received the “Energy Low” message calculate the total time of the refuel event  $T$ , and send this value to the MR.  $T$  is calculated in the same way as given by (3-12). The mobile fuelling stations (MFSs) calculate

a rendezvous location after receiving the “Energy Low” message. This is done in the same way as discussed in Section 3-3. The optimization problem that is solved to determine the rendezvous location is given by (3-9). The FS allocation becomes different, when multiple MRs want to refuel at the same FS during the same timeslot. This is explained in the following subsections.

#### 4-1-1 Maximum Bipartite Matching

The hierarchical approach is mainly used to resolve conflicts arising when multiple MRs are willing to refuel at the same FSs at the same time. One such situation occurs when multiple MRs travel towards the same FS at the same time (from now on denoted as common interest MRs). This is illustrated by the following example.

Consider MR<sub>1</sub> is travelling towards FS<sub>1</sub>. Now consider that MR<sub>2</sub> reaches a low energy value and broadcasts the “Energy Low” message, before MR<sub>1</sub> has sent the “Close” message. FS<sub>1</sub> will respond to the “Energy Low” message of MR<sub>2</sub> by sending:

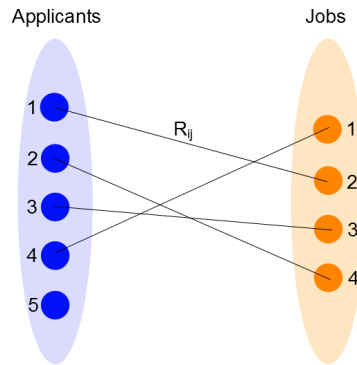
$$T_{12} = T_{12}^t + T_{12}^r + T_1^o + T_1^w$$

Here the indices 1, and 2 indicate FS<sub>1</sub>, and MR<sub>2</sub> respectively. When calculating  $T_{12}$ , FS<sub>1</sub> does not take into account the amount of time that it takes to refuel MR<sub>1</sub>. This time should be considered, since MR<sub>1</sub> is already travelling towards the FS. This implies that  $T_{12}$  is in principle too low, which can result in inefficient situations. MR<sub>2</sub> compares all received total times, and will travel towards the FS corresponding to the lowest value. Consider FS<sub>1</sub> has sent the lowest value. If no further action is taken, both MRs go to FS<sub>1</sub>, and at least one of the MR will end waiting in the queue. This can lead to a longer total mission time, in case there is another FS that is not occupied at the moment.

In practice, the time it takes to refuel a MR will often be much greater than the time it takes the MR to travel from its current location to a FS. This implies that in the scenario described above, in order to minimize the total mission time it often will be beneficial for MR<sub>1</sub>, or MR<sub>2</sub> to refuel at a different FS instead of refuelling at FS<sub>1</sub> and waiting for the other MR to be refuelled first. This issue has been addressed by Leonard et al. [13], who assign each common interest MR to a different FS, by using the auction algorithm of Bertsekas [45]. The principle of assigning each common interest MR to a different FS is applied here as well. A maximum bipartite matching algorithm is used instead of the auction algorithm of Bertsekas [45]. Maximum bipartite matching can be used to match uneven numbers of FSs, and MRs. When using the auction algorithm, only even numbers of FSs, and MRs can be matched. This algorithm can be used, but it has to be adjusted such that it can be used to match uneven numbers of FSs, and MRs. Since a maximum bipartite matching algorithm does not need any adjustments, such an algorithm is used here. The maximum bipartite matching principle is explained below.

The problem is to assign each common interest MR to a FS, such that the accumulated time spent on the refuelling activities is minimized. In graph theory this problem is known as the matching problem (TMP). In TMP a matching  $M$  of a graph  $G = (V, E)$  is a subset of the edges, such that none of the edges share the same node. TMP can also be considered for bipartite graphs, where the vertices can be divided into two independent sets  $U$  and  $V$ . Maximum bipartite matching is based on graph theory and involves matching the elements

in a bipartite graph, such that the total benefit is maximized. An example of a bipartite graph is shown in Figure 4-1. In this figure there are 5 applicants and 4 jobs which have to be assigned to the applicants. Each job can be assigned to a single applicant, which can lead to the result that some applicants end up with no job. An assignment between applicant  $i$  and job  $j$  has a reward  $R_{ij}$ . The objective is to assign all jobs to the applicants, such that the accumulated rewards are maximized. The BIPARTITE\_MATCHING function written



**Figure 4-1:** Bipartite graph with 5 applicants and 4 jobs, which have to be assigned to the applicants.

for MATLAB, by Gleich and Wang [46] is used. For the interested reader this algorithm and a detailed description about TMP can be found in [47]. According to Papadimitriou and Steiglitz [47], the computational complexity of the maximum bipartite matching algorithm is  $O(\min(|U|, |V|) \times |E|)$ . Here  $U$ , and  $V$  are the independent sets, and  $E$  is the set of edges of the bipartite graph. Here the ' $| \cdot |$ ' operator means: 'number of elements in the set'. The number of edges increases rapidly when the number of elements in  $U$ , or  $V$  increases. The total number of edges can be found by taking all possible combinations between the elements in  $U$ , and  $V$ . According to Theorem 2.2 this can be done in  $O((|U| \times |V|)!)^1$  time. In case of assigning FSs to MRs, the two independent sets of the bipartite graph are: all common interest MRs, and all FSs. When matching  $K$  MRs to  $M$  FSs, the computational complexity is  $O(\min(K, M) \times (KM)!)^1$ . Note that this is the same as  $O((KM)!)^1$ . This indicates TMP scales badly when the problem size increases. For the simulations performed during this thesis no scenarios with more than 6 MRs, and 2 FSs were considered. Since these are relative small numbers, the computation time of the maximum bipartite matching algorithm was not a limiting factor. For scenarios with a very large number of MRs, or FSs, this computation time can be a bottleneck. Computations on a laptop computer running on a Windows® 8.1 Enterprise 64 bits operating system, with Intel(R)Core(TM)i7 CPU Q720, 1.6 GHz and 4.00 GB RAM, were performed to check which problem size can be solved by the BIPARTITE\_MATCHING function in 1 s. It turned out that matching 325 MRs to 325 FSs took around 0.98 s. However in real life scenarios there will often be less FSs than MRs. When matching 725 MRs to 250 FSs, it took 0.99 s. This ratio between FSs and MRs seems feasible in practice [39]. It is very unlikely that all MRs in an environment are going to the same FS at the same time. Considering the worst case scenario it would mean that if the MRs have the same computational power as the laptop computer used here, the maximum bipartite matching algorithm could in practice be applied to a swarm of 750 MRs, if the computation time of 1 s is acceptable.

The BIPARTITE\_MATCHING function requires a matrix that contains the rewards of all possible matchings between the applicants, and the jobs as an input. The output of this function is the assignment of jobs to applicants, that maximizes the accumulated rewards. Translating the example of matching the jobs to applicants, to a matching between FSs and common interest MRs, the FSs can be seen as jobs, and the MRs as applicants. When using a maximum bipartite algorithm for assigning the FSs to MRs, the goal is to minimize the accumulated time spent on the refuelling activities. Going back to the initial example, this is realized as follows. When MR<sub>2</sub> decides to travel to FS<sub>1</sub>, it will broadcast a “Coming” message together with the unique ID number of FS<sub>1</sub>. If MR<sub>1</sub> receives this message, it will respond by sending its: position, maximum velocity, energy level, energy consumption rate, maximum energy capacity, and next waypoint location to MR<sub>2</sub>. Based on this information, and the information sent by the FSs, MR<sub>2</sub> constructs a matrix  $T_{\text{match}}$ . This matrix contains all total times of a refuel event for each possible combination between {MR<sub>1</sub>, MR<sub>2</sub>}, and {FS<sub>1</sub>, FS<sub>2</sub>}.  $T_{\text{match}}$  for this example has the form as given in Table 4-1.

**Table 4-1:** Matrix that contains the total times of a refuel event for all combinations between {MR<sub>1</sub>, MR<sub>2</sub>}, and {FS<sub>1</sub>, FS<sub>2</sub>}.

|                 | FS <sub>1</sub> | FS <sub>2</sub> |
|-----------------|-----------------|-----------------|
| MR <sub>1</sub> | $T_{11}$        | $T_{12}$        |
| MR <sub>2</sub> | $T_{21}$        | $T_{22}$        |

Here  $T_{11}$  means the the total time of the refuel event if MR<sub>1</sub> is assigned to refuel at FS<sub>1</sub>. The values of  $T$  are calculated as given by (3-12).

When matching MR<sub>k</sub>, and FS<sub>m</sub> the same two undesired scenarios can arise as discussed in Section 3-2:

- $E_{km}^f > E_k$ , the energy level of MR<sub>k</sub> is insufficient to reach FS<sub>m</sub>;
- $E_{km}^p > E_{k,\text{max}}$ , the maximum energy level of MR<sub>k</sub> is insufficient to reach the next waypoint from the location of FS<sub>m</sub>.

Again when one of these conditions holds,  $T_{mk} = \infty$ , in order to prevent these undesired situations.

The maximum bipartite matching algorithm maximizes the rewards for assigning FSs to MRs. In order to obtain a result that minimizes instead of maximizes the refuel time,  $T_{\text{match}}$  is transformed as follows:

$$\tilde{T}_{\text{match}} = \max(T_{\text{match}}) - T_{\text{match}} \quad (4-1)$$

This transformation makes the maximum value of  $T_{\text{match}}$  zero in  $\tilde{T}_{\text{match}}$ , and the smaller values of  $T_{\text{match}}$  positive values in  $\tilde{T}_{\text{match}}$ . The positive values can now be seen as positive rewards. The smaller the value in  $T_{\text{match}}$ , the larger this value will be in  $\tilde{T}_{\text{match}}$ . MR<sub>2</sub> uses  $\tilde{T}_{\text{match}}$  as an input to the BIPARTITE\_MATCHING algorithm. After running this algorithm, MR<sub>2</sub> sends MR<sub>1</sub> the location of the FS that is assigned to it. Each MR will travel towards the FS, determined by the maximum bipartite matching algorithm.

For situations where there are more than two common interest MRs, all MRs that are already travelling towards the FS respond to the “Coming” message of the last MR that decides to

travel to the FS. The last MR creates a  $T_{\text{match}}$  matrix that contains the total refuel times for all combinations, between all common interest MRs, and all FSs. This matrix has the form as indicated in Table 4-2. Where the index  $C$  indicates the total number of common interest MRs. Using this matrix as an input to the BIPARTITE\_MATCHING algorithm, the last MR determines a maximum matching between the FSs and all common interest MRs. After running the maximum bipartite matching algorithm, the last MR will broadcast which MR is assigned to which FS, and each MR will travel to its assigned FS.

**Table 4-2:** Matrix that contains the total times of a refuel event, for all combinations between common interest MRs, and all FSs.

|                 | FS <sub>1</sub> | FS <sub>2</sub> | ... | FS <sub>M</sub> |
|-----------------|-----------------|-----------------|-----|-----------------|
| MR <sub>1</sub> | T <sub>11</sub> | T <sub>12</sub> | ... | T <sub>1M</sub> |
| MR <sub>2</sub> | T <sub>21</sub> | T <sub>22</sub> | ... | T <sub>2M</sub> |
| ⋮               | ⋮               | ⋮               | ... | ⋮               |
| MR <sub>C</sub> | T <sub>C1</sub> | T <sub>C2</sub> | ... | T <sub>CM</sub> |

It is possible that there are more common interest MRs than FSs. In that case some of the MRs are not assigned to a FS when the maximum bipartite matching is performed. In order to guarantee that each common interest MR will eventually be assigned to a FS, every time the maximum bipartite matching algorithm is executed, it is checked if there are unassigned common interest MRs. If this is the case the previously assigned MRs are removed from the  $T_{\text{match}}$  matrix, and the algorithm is executed again. This sequence is repeated until all common interest MRs are assigned to a FS.

#### 4-1-2 Queue Ordering

A queue ordering algorithm has been developed, in order to reduce the time that the last MR of the queue will reach its next waypoint. In the distributed approach, the MRs in a queue are allowed to enter the FS on a first come first served basis. Sometimes it is beneficial if the MRs in the queue enter the FS in a different order than first come first served. The queue ordering algorithm proposed here is explained by the following example. Consider MR<sub>k</sub> to enter the queue of FS<sub>m</sub>. MR<sub>k</sub> sends a “Wait” message to the FS, together with its: location, maximum velocity, next waypoint location, energy level, and maximum energy level. Based on this information FS<sub>m</sub> calculates:

- $T_{mk}^r$  - The time it takes MR<sub>k</sub> to be completely refuelled by FS<sub>m</sub>.
- $T_{mk}^p$  - The time it takes MR<sub>k</sub> to travel from FS<sub>m</sub> to its next waypoint.

$T_{mk}^r$  is calculated as given by (3-8). The index  $l$  of  $T_{mlk}^r$  is omitted here, since the rendezvous locations are equal to the current positions of the MRs.  $T_{mk}^p$  is calculated as follows:

$$T_{mk}^p = \sqrt{(x_m - x_{p_{k,\text{next}}})^2 + (y_m - y_{p_{k,\text{next}}})^2} / v_{k,\text{max}} \quad (4-2)$$

When a second MR (let this be MR<sub>1</sub>) enters the queue, the same procedure is followed. Now the FS determines the refuel ordering of the MRs in the queue, by calculating the times that

both MRs will reach their next waypoint for all possible refuel orderings. In this example the possible refuel orderings are:  $\{\text{MR}_k, \text{MR}_1\}$ , and  $\{\text{MR}_1, \text{MR}_k\}$ .

The time that  $\text{MR}_k$  reaches its next waypoint, for the refuel ordering where  $\text{MR}_k$  is listed first can be calculated as follows:

$$T_{p_{k,\text{next}}} = T_{mk}^r + T_{mk}^p \quad (4-3)$$

For the refuel ordering where  $\text{MR}_k$  is listed last,  $T_{p_{k,\text{next}}}$ , can be calculated as follows:

$$T_{p_{k,\text{next}}} = T_{m1}^r + T_{mk}^r + T_{mk}^p \quad (4-4)$$

$T_{p_{k,\text{next}}}$  is calculated for each MR, in each possible refuel ordering. The refuel ordering that corresponds to the shortest time that the last MR will arrive at its next waypoint, is selected as queue order. This time value is denoted as  $T_{\text{order}}$ , and its calculation for the considered example is given by (4-5).

$$T_{\text{order}} = \min[\max(T_{mk}^r + T_{mk}^p, T_{mk}^r + T_{m1}^r + T_{m1}^p), \max(T_{m1}^r + T_{mk}^r + T_{mk}^p, T_{m1}^r + T_{m1}^p)] \quad (4-5)$$

It can occur that multiple refuel orderings, share the value of  $T_{\text{order}}$ . In these cases from these refuel orderings, the ordering is selected that leads to the minimum accumulated time it takes all MRs in the queue to reach the next waypoint.

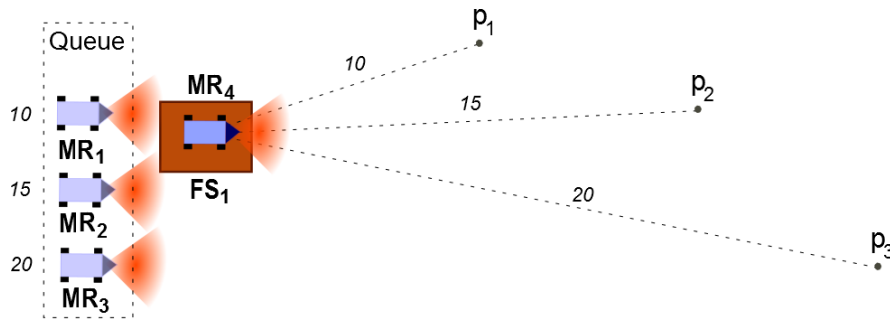
Now consider a third MR (let this be  $\text{MR}_2$ ) is entering the queue. The refuel ordering is determined in a similar way, as discussed before. Consider the previous determined refuel ordering between  $\text{MR}_k$  and  $\text{MR}_1$  to be  $\{\text{MR}_1, \text{MR}_k\}$ . Now the optimal ordering between  $\text{MR}_k$ ,  $\text{MR}_1$ , and  $\text{MR}_2$  is determined by inserting the  $T_{m2}^r$ , and  $T_{m2}^p$  values at all possible places in the existing ordering. This means the FS calculates the times all MRs reach their next waypoint for the following orderings:  $\{\text{MR}_2, \text{MR}_1, \text{MR}_k\}$ ,  $\{\text{MR}_1, \text{MR}_2, \text{MR}_k\}$ , and  $\{\text{MR}_1, \text{MR}_k, \text{MR}_2\}$ . From these refuel orderings, again the ordering that corresponds to  $T_{\text{order}}$  is selected. As soon as the FS has determined the queue ordering and it is not occupied anymore, it sends a message to the first MR in the queue order, to inform that this MR is now allowed to enter the FS.

Another scenario is possible. It can occur that multiple MRs are entering the queue simultaneously. In these cases the refuel ordering is determined by first determining the ordering between two MRs. Then the ordering is determined between three MRs, by inserting its  $T^r$ , and  $T^p$  values at all possible places in the existing ordering. This process is repeated until the ordering is determined for all MRs that entered the queue simultaneously. Since each MR is inserted in an optimal schedule, the final refuel ordering is optimal. The runtime of this approach is  $O(n^2)$  [10], where  $n$  stands for the total number of MRs in the queue. If the queue becomes very large, it could take a long time before the optimal ordering is determined. In practice this is however very unlikely, because the ratio between FSs and MRs has to be at least 0.33, in order to prevent all MRs from running out of energy [41]. This means that for every 3 MRs in a swarm, there should be at least 1 FS. This makes it unlikely that very large queues will arise, when the FSs are well distributed among the robotic swarm. Of course the ratio between FSs and MRs that should be considered to keep all MRs vivid, depends on the energy donation rates of the FSs, and the energy capacities and energy consumption rates of the MRs. This ratio should be reconsidered for each different setup.

The following aspect has to be noted about the queue ordering algorithm. Since in this thesis it is assumed that the MRs only consume energy while moving, changing the queue order does not cause MRs running out of energy. However when the MRs do consume energy while standing still, this should be taken into account in the queue ordering algorithm. This algorithm can cause a MR to be always last in the queue, every time a new MR enters the queue. In that case it is possible that MRs deplete, when they consume energy while waiting.

### Numerical Example

The queue ordering algorithm is explained by means of a numerical example. Consider the scenario of Figure 4-2. Here  $MR_1$ ,  $MR_2$ , and  $MR_3$  are waiting in the queue of  $FS_1$ , since it is currently occupied by  $MR_4$ . Next to each MR in the queue in italics  $T^r$  is shown.  $MR_1$ ,  $MR_2$ , and  $MR_3$  need 10, 15, and 20 time units respectively before being completely refuelled. Furthermore for each MR in the queue, the next waypoint is shown. The dashed lines represent the routes from the FS to the next waypoints. In italics  $T^p$  of each route is indicated.  $MR_1$ ,  $MR_2$ , and  $MR_3$  need 10, 15, and 20 time units respectively to travel from  $FS_1$  to their next waypoint. Consider that all MRs entered the queue simultaneously. Now  $FS_1$



**Figure 4-2:** Example of a scenario where 3 MRs are waiting in the queue of  $FS_1$ .

will first calculate the optimal refuel ordering between  $MR_1$ , and  $MR_2$ . Using Equations (4-3) and (4-4),  $T_{p_{k,next}}$  can be calculated for each MR. These values are given in Table 4-3.

**Table 4-3:** Values of  $T_{p_{k,next}}$ , calculated for  $MR_1$ , and  $MR_2$ , for each possible refuel ordering.

| Ordering     | $MR_1$ | $MR_2$ |
|--------------|--------|--------|
| $MR_1, MR_2$ | 20     | 40     |
| $MR_2, MR_1$ | 35     | 30     |

From Table 4-3 it can be observed that the time the last MR will arrive at its next waypoint is minimal for the refuel ordering  $\{MR_2, MR_1\}$ . Since this order results in a single minimum, this will be the order in which  $MR_1$ , and  $MR_2$  are allowed to enter  $FS_1$ . Now to determine when  $MR_3$  is allowed to enter the FS,  $MR_3$  is inserted in all possible positions in the existing optimal order. The possible refuel orderings and corresponding  $T_{p_{k,next}}$  of each MR are given in Table 4-4. From Table 4-4 it can be observed that the following orderings lead to a minimal time that the last MR will arrive at its next waypoint:  $\{MR_3, MR_2, MR_1\}$ , and  $\{MR_2, MR_3, MR_1\}$ . For both refuel orderings the last MR will arrive after 55 time units at its next waypoint. In this case the final refuel ordering is selected based on the accumulated

**Table 4-4:** Values of  $T_{p_{k,next}}$ , calculated for MR<sub>1</sub>, MR<sub>2</sub>, and MR<sub>3</sub> for the existing optimal schedule.

| Ordering                                          | MR <sub>1</sub> | MR <sub>2</sub> | MR <sub>3</sub> | Accumulated $T_{p_{k,next}}$ |
|---------------------------------------------------|-----------------|-----------------|-----------------|------------------------------|
| MR <sub>3</sub> ,MR <sub>2</sub> ,MR <sub>1</sub> | 55              | 50              | 40              | 145                          |
| MR <sub>2</sub> ,MR <sub>3</sub> ,MR <sub>1</sub> | 55              | 30              | 55              | 140                          |
| MR <sub>2</sub> ,MR <sub>1</sub> ,MR <sub>3</sub> | 45              | 30              | 65              | 140                          |

$T_{p_{k,next}}$ . Since the accumulated  $T_{p_{k,next}}$  has the lowest value for the ordering {MR<sub>2</sub>,MR<sub>3</sub>,MR<sub>1</sub>}, this will be the chosen queue order.

## 4-2 Conclusion

This chapter discussed a hierarchical approach to schedule the refuelling activities of multiple MRs. This approach is called hierarchical, since the decision making is done for individuals, or clusters of MRs, based on local knowledge. The refuel event selection methods are the same as implemented for the distributed approach.

Section 4-1 discusses the FS allocation principle. This is the part where the hierarchical approach differs from the distributed. The hierarchical approach reallocates the FSs, when multiple MRs want to refuel at the same FS during the same timeslot. A queue ordering algorithm has been proposed, in order to determine the ordering in which the MRs in a queue are allowed to enter a FS. The ordering is determined in such a way, that the time the last MR in the queue will arrive at its next waypoint is minimized. Since these decisions are made for clusters of robots, it is expected that the hierarchical approach will outperform the distributed approach.

The main contributions of this chapter are: the use of a bipartite matching algorithm to assign FSs to MRs with common refuelling interests, and the formulation of the queue ordering algorithm. The next chapter will discuss simulations that were performed to compare the centralized, distributed, and hierarchical approaches.

# Comparison between Centralized, Distributed and Hierarchical Approaches

This chapter describes a comparison between all approaches described in Chapters 2–4. The performance of the approaches is compared in terms of total mission time. For the comparison, the same two missions as discussed in Section 2-3 are used: the patrol mission, and the mission with two randomly placed waypoints for each mobile robot (MR). Furthermore the simulation results of the distributed, and hierarchical approaches are compared based on the simulations of the missions with 10-50 randomly placed waypoints, which are discussed in Section 3-4. A short recap on the different approaches is given below.

The centralized approach makes use of a central node to perform all calculations, and decision making for all MRs, based on global knowledge. The advantage of a generalized architecture is that global optimal plans can be formulated which lead to global optimal solutions. Drawbacks of this architecture are that it is not robust to dynamic environments, communication failures, and other uncertainties. Furthermore centralized architectures are highly vulnerable because they have a central point of failure. These approaches are most suited for problems involving small robotic teams and static environments.

The calculations and decision making of the distributed approach, are distributed over the robotic team. Each MR makes individual decisions based on local knowledge. There is only limited communication between the fuelling stations (FSs), and the MRs. FSs are not able to communicate with other FSs, and MRs are not able to communicate with other MRs. Advantages of a distributed architecture is that it is typically very fast, adaptive, and robust to failures. Due to the distributed nature, the solutions are suboptimal. A distributed architecture is best suited for scenarios where large robotic teams, perform relative simple tasks, with no requirements for efficiency.

The hierarchical approach also distributes the calculations and decision making over the robotic team. The difference with the distributed approach is that the MRs can communicate with each other, and decisions are made for individuals, or clusters of MRs, based on

local knowledge. This architecture is in the middle of a centralized and distributed architecture. The solutions are often suboptimal, but in general better compared with distributed approaches. Since the computations and decision making is distributed over the robotic team, hierarchical approaches retain the benefits of distributed approaches regarding to: speed, flexibility, and robustness.

Based on the properties of the different approaches, it is expected that the solution quality of the: centralized approach will be the best, hierarchical approach will be second best, and distributed approach will be the worst.

## 5-1 Simulation Results and Discussion

The patrol mission, and the mission with 2 FSs, and 3 MR which have to visit 2 randomly placed waypoints, which are described in Section 2-3 are used to evaluate the performance of all different approaches. For each mission, and each approach the same trials were performed as given in Section 2-3. In total for each approach, and each mission 30 trials were performed, where the energy levels of the MRs varied between 15-30 energy units in intervals of 5. The velocities of the MRs were varied between 1-3 distance units per time unit, in intervals of 1. The energy donation rates of the FSs were varied between 1, and 2 energy units per time unit. When using the fixed threshold method during the patrol mission, the threshold value was chosen to be 5.8 energy units. This value is calculated using Equation (3-1), such that the location with (x,y) coordinates (5,3) can be reached from the farthest possible waypoint. The location (5,3) is chosen, because this point is in the middle of the two FSs.

Table 5-1 shows the average total mission times, over 30 trials, for each mission, and all approaches. In all tables shown in this chapter the notations Fixed, and Adaptive or used to indicate the fixed, and adaptive threshold method respectively. In order to get a clearer

**Table 5-1:** Average mission times of all approaches, for the patrol mission, and the mission 2 with randomly placed waypoints for each MR.

| Mission | Optimal approach | Heuristic 1 | Receding horizon | Distributed |          | Hierarchical |          |
|---------|------------------|-------------|------------------|-------------|----------|--------------|----------|
|         |                  |             |                  | Fixed       | Adaptive | Fixed        | Adaptive |
| Patrol  | 37.64            | 38.04       | 40.43            | 46.17       | 44.91    | 48.39        | 45.33    |
| Random  | 30.18            | 30.32       | 33.17            | 43.10       | 36.79    | 43.06        | 38.30    |

view, the average total mission times over both missions are calculated. The mean error between the optimum, and the other approaches are calculated and given in Table 5-2 in percentages. The comparison between the optimal approach, Heuristic 1, and the receding horizon principle, is already given in Section 2-3 and is therefore omitted here. From Tables 5-1 and 5-2 some interesting results can be observed. A surprising result is that the distributed approach outperforms the hierarchical approach. During the simulations, the following was observed regarding the hierarchical approach. In the first fuel cycle of the MRs it often occurred that the three MRs were going for refuel at approximate the same moment. In the observed scenarios, two MRs were going to the same FS (these are common interest MRs), and the third went to the other FS. One of the common interest MRs ran the bipartite matching

**Table 5-2:** Mean errors of all approaches w.r.t. the optimal approach, calculated over the average mission times of the patrol mission, and mission with 2 randomly placed waypoints for each MR.

| Optimal approach | Heuristic 1 | Receding horizon | Distributed |          | Hierarchical |          |
|------------------|-------------|------------------|-------------|----------|--------------|----------|
|                  |             |                  | Fixed       | Adaptive | Fixed        | Adaptive |
| -                | 0.8%        | 8.4 %            | 31.6%       | 20.7%    | 34.8 %       | 23.6%    |

algorithm, and assigned one of the common interest MRs to the other FS. In this case this decision is inefficient, since another MR was already travelling to the other FS. Now still one MR ended up in the queue. The MR which FS got reassigned, travelled a longer distance. This increased its travel, and refuel time. So for these kind of scenarios, the choice to assign each common interest MR to a different FS can lead to an increase of the total mission time. After the first fuel cycle, the MRs alternate more regularly at the FSs. This was also shown by Sempé et al. [39]. Due to this effect, the MRs become more evenly distributed across the FSs. In that case it makes sense to assign each common interest MR to a different FS. Since the two missions considered here were relative short, the hierarchical approach had not enough time to compensate for the inefficient decision made in the beginning. Furthermore it can be noticed that the receding horizon principle outperforms the decentralized methods. This is not surprising, since the decisions made by the receding horizon principle are based on knowledge about all MRs, instead of individuals, or clusters of MRs. In general the decisions made by the receding horizon principle are based on more knowledge about the environment, compared to the decentralized approaches.

In order to verify if the previous drawn conclusions also hold for some larger scale scenarios, simulations were performed using Heuristic 1, the receding horizon principle, and the decentralized methods for scenarios with 7 randomly placed waypoints for each MR. The waypoints were randomly placed in an area of  $10 \times 10$  distance units, with the restriction that the euclidean distance between two consecutive waypoints should be at least 7.5 distance units. The setup was similar to the simulation with 2 randomly placed waypoints for each MR. The environment contained:

- 2 FSs:  $\{FS_1, FS_2\}$ , located at  $(5, 5)$ , and  $(5, 1)$  respectively. The properties of the FSs were:
  - $v_{m,\max} = 0, \forall m \in \mathcal{M}$ ;
  - $\dot{E}_m = 1$  for  $m = 1$ , and  $\dot{E}_m \in \{1, 2\}$  for  $m = 2$ .
- 3 MRs, which initial locations were chosen corresponding to the solution of Heuristic 1. The energy consumption rates of the MRs were:
  - $\dot{E}_k = 1, \forall k \in \mathcal{K}$ .

The maximum energy values of the MRs were varied between 15-30, in intervals of 5 energy units. The velocities of the MRs varied between 1-3, in intervals of 1 distance unit per time unit. When using a fixed threshold, the threshold value was set to 7.1 energy units, which is calculated in the same way as explained in Section 3-4. In total 10 trials were performed. The average mission times are presented in Table 5-3. From this table it can be observed

that the average total mission time is the smallest, when using Heuristic 1. Since the optimum cannot be calculated in reasonable time for this problem size, the mean errors of the other approaches with respect to Heuristic 1 were calculated. These values are shown in percentages in Table 5-4. Comparing this table to Table 5-2, some interesting observations can be made. First, the performance of the hierarchical approach improved compared to the distributed approach. When a fixed threshold was used the distributed approach performed slightly better, but when an adaptive threshold was applied, the hierarchical approach outperformed the distributed approach. This indicates that the conclusion can be correct that due to an inefficient decision made by the hierarchical approach during the first fuel cycle of the MRs, the distributed approach outperforms the hierarchical for scenarios with a very small number of waypoints. Secondly it can be observed that the mean error between the decentralized approaches using an adaptive threshold, and Heuristic 1 are much smaller. As a third observation, the average solution quality of the receding horizon principle became a little worse. These differences can be caused by the problem size considered in Table 5-2. It is possible, that scenarios where 2 waypoints are assigned to each MR are too small to draw conclusions about the suboptimality of the decentralized approaches, and the receding horizon principle. Unfortunately it is not possible to use Heuristic 1 for the same scenarios involving a much larger number of waypoints. Otherwise it would be interesting to compare the performances of the decentralized approaches, and the receding horizon principle with the heuristic for even larger problem sizes. It is expected that the mean error of the suboptimal approaches eventually will converge. It is surprising to see that on average the performance of the hierarchical approach using an adaptive threshold is better compared to the receding horizon principle for the mission with 7 randomly placed waypoints. The distributed approach using an adaptive threshold only performed 2.2 % worse compared to the receding horizon principle. Unfortunately due to the computational complexity of the receding horizon principle we did not perform simulations for scenarios with more than 7 waypoints for each MR. The trials described here took around 3-4 hours. It would be interesting to compare the performance of the receding horizon principle with the decentralized methods for scenarios with even larger number of waypoints.

**Table 5-3:** Average mission times for the mission with 7 randomly placed waypoints for each MR.

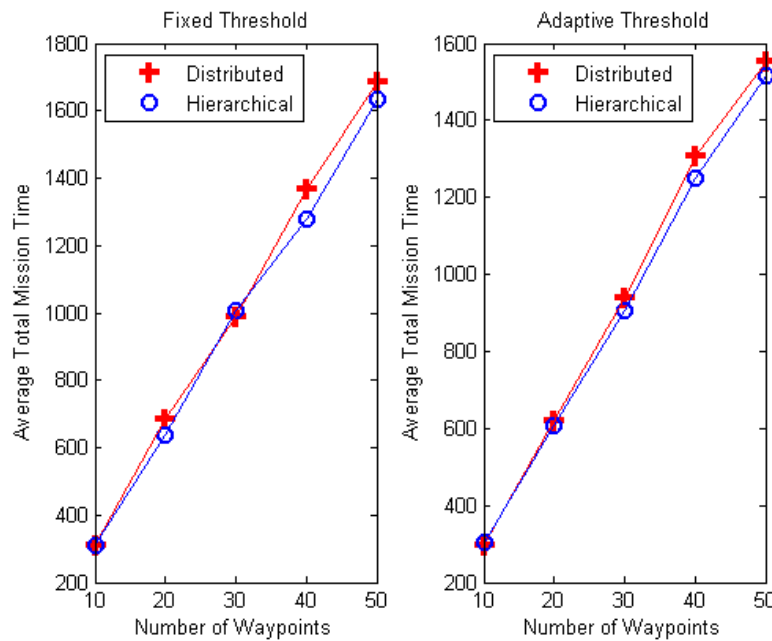
| Heuristic 1 | Receding horizon | Distributed |          | Hierarchical |          |
|-------------|------------------|-------------|----------|--------------|----------|
|             |                  | Fixed       | Adaptive | Fixed        | Adaptive |
| 121.3       | 134.7            | 160.8       | 137.5    | 161.0        | 134.6    |

**Table 5-4:** Mean errors w.r.t. Heuristic 1, calculated over the average mission times of the mission with 7 randomly placed waypoints for each MR.

| Heuristic 1 | Receding horizon | Distributed |          | Hierarchical |          |
|-------------|------------------|-------------|----------|--------------|----------|
|             |                  | Fixed       | Adaptive | Fixed        | Adaptive |
| -           | 11.1 %           | 32.6%       | 13.3%    | 32.7%        | 10.9%    |

The distributed and hierarchical approaches were also compared for larger simulations, up

to 50 waypoints. The same simulations were performed using the hierarchical approach as discussed in Section 3-4 using the distributed approach. The simulation setup was exactly the same. Five different environments were created, where 10, 20, 30, 40, and 50 waypoints were randomly placed in an area of  $10 \times 10$  distance units, with the restriction that the euclidean distance between two consecutive waypoints should be at least 7.5 distance units. The same trials were performed as discussed in Section 3-4, with varying velocities, and energy capacities of the MRs, and varying velocities, and energy donation rates of the FSs. Figure 5-1 shows the result for the distributed, and hierarchical approach, when a fixed, and adaptive threshold was applied. The fixed threshold value was chosen to be 7.1 energy units, equal to the threshold value of the simulations described in Section 3-4. This figure shows that most of the times the hierarchical approach performs better compared to the distributed approach. This was already expected, for scenarios with a larger number of waypoints. Since after the first fuel cycle the MRs alternate more regularly at the FSs, the MRs become more evenly distributed over the FSs. In that case the hierarchical approach turns out to be beneficial. Since the fuel cycles of the MRs are not synchronised, still situations can arise where it is inefficient to assign each common interest MR to a different FS. If these situation occur often, the hierarchical approach can perform worse compared to the distributed approach. This can explain why in the left figure of Figure 5-1, the distributed approach performed better compared to the hierarchical approach, for 30 waypoints.



**Figure 5-1:** Comparison between distributed, and hierarchical approach, for scenarios with 2 FSs, and 6 MRs, which have to visit randomly placed waypoints.

The average total mission times over the 10-50 waypoints are calculated, in order to draw an overall conclusion. These values are summarized in Table 5-5. The mean error of the distributed approach is calculated with respect to the hierarchical approach. These values are denoted by  $\mu$  and given in percentages. On average the hierarchical approach performed better compared to the distributed approach. These results indicate that the previous drawn

conclusion is correct. For scenarios with a small number of waypoints, the hierarchical approach often leads to an inefficient decision during the first fuel cycle. For small scale scenarios, the overall mission time is too short for the hierarchical approach to become beneficial again. However for larger scale scenarios it is indicated that on average the hierarchical approach performs better compared to the distributed approach.

**Table 5-5:** Average mission times for the distributed, and hierarchical approaches, for scenarios with 2 FSs, and 6 MRs, which have to visit 10-50 randomly placed waypoints.

|          | Distributed          |       | Hierarchical         |
|----------|----------------------|-------|----------------------|
|          | $T_{\text{mission}}$ | $\mu$ | $T_{\text{mission}}$ |
| Fixed    | 1008.3               | 3.6 % | 973.6                |
| Adaptive | 943.9                | 3.0%  | 916.2                |

As already discussed in Chapters 3 and 4, the benefits of having a mobile fuelling station (MFS) were larger when a fixed threshold was applied, instead of an adaptive threshold. Table 5-6 shows the average total mission times calculated over both decentralized methods, and over all trials, with FS<sub>2</sub> being fixed, and mobile. On average the total mission times of the scenarios with 2 fixed FSs was 14.3%, and 5.7% longer when using a fixed, and adaptive threshold respectively, compared with the scenarios with 1 fixed and 1 mobile FS. As explained before, the difference between the fixed, and adaptive threshold can be caused by not taking into account the rendezvous location of the MFSs during the calculation of the adaptive threshold value.

**Table 5-6:** Comparison between scenarios with 2 fixed FSs, and 1 fixed and 1 mobile FS, for scenarios with 6 MRs, which have to visit 10-50 randomly placed waypoints.

|          | FS <sub>2</sub> fixed |       | FS <sub>2</sub> mobile |
|----------|-----------------------|-------|------------------------|
|          | $T_{\text{mission}}$  | $\mu$ | $T_{\text{mission}}$   |
| Fixed    | 1080.9                | 14.3% | 945.9                  |
| Adaptive | 964.7                 | 5.7%  | 912.7                  |

## 5-2 Conclusion

This chapter presented the simulation results of all approaches presented in Chapters 2–4. The performance of the different approaches were compared based on total mission time. The solution quality of Heuristic 1 is on average 0.8% from the optimal value. It can be concluded that Heuristic 1 outperforms all other suboptimal approaches. Based on the experiment results the receding horizon principle performed better compared to the decentralized approaches, for scenarios with a small number of waypoints. It is likely that the problem size considered during these simulations were too small, to draw a conclusion about the suboptimality of these approaches. For scenarios with a larger number of waypoints the solution quality of the receding horizon principle decreased. Surprisingly the solution quality of the decentralized approaches using an adaptive threshold increases tremendously for these scenarios. On average for these scenarios the hierarchical approach using an adaptive threshold performed slightly better, and the distributed approach using an adaptive threshold performed little worse compared to the receding horizon principle. In general the solution quality of the decentralized

approaches is better when an adaptive threshold is applied compared to a fixed threshold. For use in practice the decentralized approaches with an adaptive threshold are recommended for scenarios with a large number of waypoints, instead of the receding horizon principle. This recommendation is based on the following two reasons: the computational complexity of the decentralized methods is much lower, and their solution quality does not differ a lot compared to the receding horizon principle. Furthermore there are general reasons why a decentralized approach is favoured above a centralized approach, regarding to: robustness, flexibility, and adaptability. For small scale scenarios the distributed approach outperforms the hierarchical approach. This is most likely due to an inefficient decision made by the hierarchical approach during the first fuel cycle of the MRs. For larger scale scenarios, on average the hierarchical approach performs better compared to the distributed approach. There is no overall best approach, which should be used in all kind of scenarios. Which approach one should use depends on several factors, such as problem scale, efficiency requirements, and communication possibilities. In general one can formulate the following recommendations. For small problem sizes with high efficiency requirements, an off-line schedule can be found using the optimal approach, or Heuristic 1. For dynamic environments or large scale problems, the distributed, or hierarchical approach, using an adaptive threshold leads to reasonable results.



---

## Chapter 6

---

# Conclusions

### 6-1 Summary

This thesis presented several methods to schedule the refuelling activities of multiple heterogeneous autonomous mobile robots (MRs). The objective was to schedule the refuelling activities in such a way that the overall mission time was minimized. The scheduling was focused on the selection of the refuel events, and the allocation of the fuelling stations (FSs) as a shared resource. A general contribution is that for all proposed approaches, heterogeneous MRs, and FSs were considered.

In particular a centralized, distributed, and hierarchical approach have been presented in order to schedule the refuelling activities. The centralized approach which is discussed in Chapter 2 solves a mixed integer linear program (MILP) that results in global optimal schedules. To the best of our knowledge we are the first to formulate a MILP in order to schedule the refuelling activities of multiple MRs, across multiple FSs, using a time-based metric. This is the greatest contribution of the presented centralized approach. All possible refuel events have to be taken into account, in order to obtain a global optimal solution. Since the number of refuel events grows very fast when the problem size increases, the computation time grows very quickly as well. A heuristic has been proposed which only takes fixed refuel orderings into account, in order to speed up the computation time. The solution quality of this heuristic had proven to be close to the optimal value. A significant reduction in computation time has been noticed, and the size of the problems that can be solved in reasonable time increased tremendously. One factor that leads to a fast increasing computation time, is the total number of waypoints. In order to solve problems with a large number of waypoints, the problem was solved using a receding horizon principle. This principle solves subproblems, with the number of waypoints being equal to the prediction horizon. By repeatedly solving these subproblems, and shifting the prediction horizon, larger scale problems can be solved. The main contributions of the centralized approach are: the formulation of a MILP to solve type of scheduling problems under consideration using a time-based-metric, and the implementation of the receding horizon principle.

Chapter 3 presented a distributed approach, where each MR made individual decisions based on local knowledge. Two refuel event selection methods have been discussed. These methods make use of a fixed, and adaptive threshold on the energy level of a MR. As expected the simulation results show that the adaptive threshold method outperforms the fixed threshold. In order to increase the operation range and time of the MRs, mobile fuelling stations (MFSs) were introduced. A rendezvous strategy has been proposed to determine a location where a MFS and a MR can rendezvous, such that the time the MR spends on the refuelling activity is minimized. During simulations it was confirmed that the use of MFSs leads to a reduced total mission time. The main contributions of the distributed approach presented here is the implementation of the MFSs, which need to be shared together with the fixed FSs in order to refuel the MRs in an efficient manner.

A hierarchical approach was proposed in Chapter 4. Here the decisions are made for individuals or clusters of MRs, based on local knowledge. This approach reallocates the FSs, in case multiple MRs want to refuel at the same FS during the same timeslot. For this purpose a maximum bipartite matching algorithm [46] has been used. In order to reduce the common time spent on refuelling activities by the MRs which are waiting in a queue of a FS, a queue ordering algorithm has been proposed. The contributions of the hierarchical approach are: the use of a bipartite matching algorithm in order to reallocate the FSs when multiple MRs have common refuelling interests, and the implementation of the queue ordering algorithm.

A comparison has been made between all presented approaches in Chapter 5. A variety of simulations were discussed, and the approaches were compared based on the total mission time. Since the problem size that can be solved using the centralized approaches is limited, the comparison with these approaches was only done for small scale systems. The largest problem scenario for which an optimal solution can be found in reasonable time is: 2 fixed FSs, and 3 MRs which all have to visit 2 waypoints. This indicates an important limitation of the optimal approach. The proposed heuristic improved the computation time significantly, but is still very computationally complex. The solvable problem size stays limited. In general for type of scheduling problems under consideration, the centralized approaches are not recommended for large scale scenarios. For scenarios with a large number of waypoints, the receding horizon principle could be used, although it takes a long time until a solution is found. The receding horizon principle outperforms the decentralized approaches for scenarios with a very small number of waypoints. However when the number of waypoints increases the performance of the decentralized approaches using an adaptive threshold, seems to improve compared to the receding horizon principle. For scenarios with 7 randomly placed waypoints for each MR, on average the hierarchical approach performed slightly better, and the distributed approach performed little worse compared to the receding horizon principle. Unfortunately the computational complexity of the receding horizon was too large, to perform simulations for scenarios with a much larger number of waypoints. For scenarios with a small number of waypoints the hierarchical approach seems to perform worse compared to the distributed approach. This is most likely caused by an inefficient decision made by the hierarchical approach, during the first fuel cycle of the MRs. The distributed and hierarchical approach were also compared for scenarios with 2FSs, and 6 MRs which had to visit up to 50 randomly placed waypoints. For these scenarios the solution quality of the hierarchical approach improved, compared to the distributed approach. When the number of waypoints increases, the hierarchical approach has the time to compensate for the inefficient decision made in the beginning. On average the hierarchical approach performed 3% to 3.6 % better

than the distributed approach, depending on the refuel event selection method. Since the decentralized approaches are less computationally complex, for large scale systems they are favoured above the centralized approaches. For some problem scales, a centralized approach cannot even find a solution in reasonable time. Each approach has its pros and cons, and therefore it is hard to determine which approach is the best. In general one can formulate the following recommendations. For small problem sizes, with high efficiency requirements, an off-line schedule can be found using the optimal approach, or the proposed heuristic. For dynamic environments or large scale problems, the distributed, or hierarchical approach, using an adaptive threshold can be used to obtain reasonable results in real time.

This work resulted in some contributions to the existing field of autonomous refuelling of MRs. The developed methods can be applied for scheduling the refuelling activities of individuals, or multiple MRs. The proposed scheduling methods can be used to refuel MRs during a large variety of missions, e.g. work in hazardous environments, persistent labour, or planetary exploration. Another application one could think of in the near future is scheduling the refuelling activities of self-driving cars.

## 6-2 Future Work

A major contribution can be made by reducing the computational complexity of the MILP formulation. This would make it possible to find optimal solutions in reasonable time for larger scale problems. A reformulation of the MILP might be needed, or possibly there are some relaxations that can be used. The bottlenecks of the MILP formulation are the number of MRs, and the number of waypoints.

The MILP formulation does not allow the FSs to be mobile. A reformulation can be considered to include this option. Klauco et al. [1] describe a mixed integer second order cone program (MISOCP) formulation for a heterogeneous multi-vehicle, which can be interpreted as a system that consists of a single MFS and a single MR. The decision of where the MFSs, and MRs rendezvous is part of the formulation. With the MILP formulation proposed in this thesis, combined with the MISOCP formulation of Klauco et al. [1], it should be possible to include the option of FSs to be mobile as well. It is important to note that this will increase the computation time.

Solving the centralized approach with the receding horizon principle, made it possible to solve refuelling problems for an arbitrary number of waypoints. Since the number of MRs is still a limiting factor, a contribution can be made here. A proposed method to realize that this approach can also be used for an arbitrary number of MRs is given as follows. Instead of solving the optimization problem for all MRs at once, solve the problem first for  $\{MR_1, MR_2\}$ . Subsequently solve the problem for  $\{MR_2, MR_3\}, \dots, \{MR_K, MR_1\}$ . Here  $K$  denotes the total number of MRs. From these pairwise solutions the optimal solution should be chosen. Note that this can be very complex, since each refuel event of one MR can influence the refuel events of all other MRs.

An open topic which is not addressed during this thesis is to determine the duration of a refuel event. The work presented here, assumed that a MR always stays at a FS, until it is completely refuelled. Possibly the total mission time can be reduced, if different refuel durations are taken into account. There is little literature that addresses the refuel duration

of MRs. Wawerla and Vaughan [9] propose a time discounted labour approach. The refuel duration with respect to the total energy of a group of MRs was considered by Michaud and Robichaud [48].

The problem under consideration entails an assignment of a unique set of waypoints to each MR. As future work problems can be considered, where the mission entails a set of waypoints assigned to the whole robotic team instead of individuals. In that case the waypoints should be assigned to the MRs, such that the overall mission time is minimized. These type of problems are also known as: task allocation for multi-robot exploration [49], a variant of the multiple travelling salesman problem (MTSP) [50].

It is expected that the performance of approaches where an adaptive threshold strategy is applied can be increased, when the rendezvous locations of the MFSs are taken into account during the calculation of the threshold value. This would be a nice contribution. A method to realize this is proposed as follows. Every time a MR reaches a waypoint, it has to send a message to the FSs, which calculate a rendezvous location when receiving this message. This rendezvous location has to be send back to the MR, which in turn calculates if the closest rendezvous location can be reached after visiting the next waypoint.

Another improvement of the hierarchical approach can be made by reconsidering the policy that if multiple MRs are travelling towards, the same FS, each MR is assigned to a different FS. During the simulations it turned out that this policy sometimes leads to an inefficient FS allocation. Instead of always reallocating the FSs, the outcome of the bipartite matching algorithm should be compared with the result of the initial FS allocation. In case the outcome of the bipartite matching algorithm is better, the FSs should be reallocated. Otherwise, it is beneficial to leave the FS allocation as it was originally.

In order to prevent MRs from becoming useless when depleted, a nice contribution would be if MFSs are able to revive MRs that ran out of energy. This can be realized as follows: When almost out of energy, and not able to reach a FS, MRs should be able to broadcast an “Emergency” signal. MFSs receiving this messages, have to decide which MFS will travel towards the MR to revive it. A similar principle is discussed in [51].

It is assumed that the FSs have an unlimited amount of energy. With this assumption the necessity for the MFSs to refuel themselves is neglected. As an extension of the current problem, one could take the energy capacity, and refuelling needs of the MFSs into account. The following items should be taken into consideration:

- The energy level of the MFSs should decrease when it is: moving, and refuelling a MR. In the latter case the energy level of the MFS should decrease with an amount equal to its energy donation rate (if energy losses are neglected).
- The MFSs should be able to refuel either at a fixed FS, or possibly at another MFS as well.
- The MFSs should share the available resources with the MRs.

Another possibility is to think of MFSs with different energy capacities, such that MFSs with smaller energy capacities can refuel at MFSs with larger energy capacities.

---

# Bibliography

- [1] M. Klaufic, S. Blazek, M. Kvasnica, and M. Fikar, “Mixed-integer socp formulation of the path planning problem for heterogeneous multi-vehicle systems,” in *Control Conference (ECC), European*. IEEE, 2014, pp. 1474–1479.
- [2] J. Shen and H. Hu, “Visual navigation of a museum guide robot,” in *Intelligent Control and Automation, WCICA. The Sixth World Congress on*, vol. 2. IEEE, 2006, pp. 9169–9173.
- [3] J. Forlizzi and C. DiSalvo, “Service robots in the domestic environment: a study of the roomba vacuum in the home,” in *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*. ACM, 2006, pp. 258–265.
- [4] J. E. Young, R. Hawkins, E. Sharlin, and T. Igarashi, “Toward acceptable domestic robots: Applying insights from social psychology,” *International Journal of Social Robotics*, vol. 1, no. 1, pp. 95–108, 2009.
- [5] W. G. Walter, “An imitation of life,” *Scientific American*, vol. 182(5), pp. 42–45, 1950.
- [6] L. Steels, “A case study in the behavior-oriented design of autonomous agents,” *From animals to animats*, vol. 3, pp. 445–452, 1994.
- [7] T. Belpaeme and A. Birk, “A real-world ecosystem featuring several robot species,” in *Proceedings, European Conference on Artificial Life, ECAL*, vol. 97, 1997.
- [8] J. Wawerla and R. T. Vaughan, “Near-optimal mobile robot recharging with the rate-maximizing forager,” in *Advances in Artificial Life*. Springer, 2007, pp. 776–785.
- [9] J. Wawerla and R. T. Vaughan, “Optimal robot recharging strategies for time discounted labour,” in *ALIFE*, 2008, pp. 670–677.
- [10] V. Marmol, B. Kannan, and M. B. Dias, “Market-based coordination of recharging robots,” 2012.
- [11] P. Zebrowski and R. T. Vaughan, “Recharging robot teams: A tanker approach,” in *Advanced Robotics. ICAR’05. Proceedings., 12th International Conference on*. IEEE, 2005, pp. 803–810.

- [12] J. Kim, B. D. Song, and J. R. Morrison, "On the scheduling of systems of uavs and fuel service stations for long-term mission fulfillment," *Journal of Intelligent & Robotic Systems*, vol. 70, no. 1-4, pp. 347–359, 2013.
- [13] J. Leonard, A. Savvaris, and A. Tsourdos, "Energy management in swarm of unmanned aerial vehicles," in *Unmanned Aircraft Systems (ICUAS), 2013 International Conference on*. IEEE, 2013, pp. 124–133.
- [14] D. McFarland and E. Spier, "Basic cycles, utility and opportunism in self-sufficient robots," *Robotics and Autonomous Systems*, vol. 20, no. 2, pp. 179–190, 1997.
- [15] K. Kim, H. Choi, S. Yoon, K. Lee, H. Ryu, C. Woo, and Y. Kwak, "Development of docking system for mobile robots using cheap infrared sensors," in *Proceedings of the 1st International Conference on Sensing Technology*, 2005, pp. 287–291.
- [16] R. C. Luo, C. T. Liao, K. L. Su, and K. C. Lin, "Automatic docking and recharging system for autonomous security robot," in *Intelligent Robots and Systems. IEEE/RSJ International Conference on*. IEEE, 2005, pp. 2953–2958.
- [17] K. Sugawara, T. Kazama, and T. Watanabe, "Foraging behavior of interacting robots with virtual pheromone," in *Intelligent Robots and Systems. Proceedings. IEEE/RSJ International Conference on*, vol. 3. IEEE, 2004, pp. 3074–3079.
- [18] M. Milford and G. Wyeth, "Persistent navigation and mapping using a biologically inspired slam system," *The International Journal of Robotics Research*, vol. 29, no. 9, pp. 1131–1153, 2010.
- [19] J. J. Troy, C. A. Erignac, and P. Murray, "Closed-loop motion capture feedback control of small-scale aerial vehicles," in *AIAA Infotech@ Aerospace 2007 Conference and Exhibit*, 2007, pp. 7–10.
- [20] M. Bonani, V. Longchamp, S. Magnenat, P. Rétornaz, D. Burnier, G. Roulet, F. Vausard, H. Bleuler, and F. Mondada, "The marxbot, a miniature mobile robot opening new perspectives for the collective-robotic research," in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*. IEEE, 2010, pp. 4187–4193.
- [21] D. Austin, L. Fletcher, and A. Zelinsky, "Mobile robotics in the long term-exploring the fourth dimension," in *Intelligent Robots and Systems. Proceedings. IEEE/RSJ International Conference on*, vol. 2. IEEE, 2001, pp. 613–618.
- [22] L. J. Chen, W. I. S. Tong, B. Meyer, A. Abdolkhani, and A. P. Hu, "A contactless charging platform for swarm robots," in *IECON 37th Annual Conference on IEEE Industrial Electronics Society*. IEEE, 2011, pp. 4088–4093.
- [23] A. Birk, "Autonomous recharging of mobile robots," in *Proceedings of the 30th International Symposium on Automotive Technology and Automation*. Isata Press. Citeseer, 1997.
- [24] T. Nugent, J. Kare, and L. LaserMotive, "Laser power for uavs," *Laser Motive White Paper-Power Beaming for UAVs*, NWEN, 2010.

- [25] G. Laporte, "The traveling salesman problem: An overview of exact and approximate algorithms," *European Journal of Operational Research*, vol. 59, no. 2, pp. 231–247, 1992.
- [26] G. Laporte, Y. Nobert, and S. Taillefer, "Solving a family of multi-depot vehicle routing and location-routing problems," *Transportation science*, vol. 22, no. 3, pp. 161–172, 1988.
- [27] G. Laporte and Y. Nobert, "A branch and bound algorithm for the capacitated vehicle routing problem," *Operations-Research-Spektrum*, vol. 5, no. 2, pp. 77–85, 1983.
- [28] G. Rochart, E. Monfroy, and N. Jussien, "Minlp problems and explanation-based constraint programming," in *4th Workshop on Cooperative Solvers in Constraint Programming (COSOLV'04)*, 2004.
- [29] R. Kannan and C. L. Monma, *On the computational complexity of integer programming problems*. Springer, 1978.
- [30] M. R. Bussieck and A. Pruessner, "Mixed-integer nonlinear programming," *SIAG/OPT Newsletter: Views & News*, vol. 14, no. 1, pp. 19–22, 2003.
- [31] T. Baker, J. Gill, and R. Solovay, "Relativizations of the  $p=?np$  question," *SIAM Journal on computing*, vol. 4, no. 4, pp. 431–442, 1975.
- [32] H. Mittelmann, "Recent benchmarks of optimization software," in *22nd European Conference on Operational Research (EURO XXII Prague, Czech Republic: Dept of Math and Stats Arizona State University)*, 2007.
- [33] J. F. Sturm, "Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones," *Optimization methods and software*, vol. 11, no. 1-4, pp. 625–653, 1999.
- [34] J. Löfberg, "Yalmip : A toolbox for modeling and optimization in MATLAB," in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. [Online]. Available: <http://users.isy.liu.se/johanl/yalmip>
- [35] Z. Yan, N. Jouandeau, and A. A. Cherif, "A survey and analysis of multi-robot coordination," *International Journal of Advanced Robotic Systems*, vol. 10, 2013.
- [36] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz, "Market-based multirobot coordination: A survey and analysis," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1257–1270, 2006.
- [37] C. H. Papadimitriou, *Computational complexity*. John Wiley and Sons Ltd., 2003.
- [38] T. J. van den Boom, G. D. Lopes, and B. De Schutter, "A modeling framework for model predictive scheduling using switching max-plus linear models," in *Decision and Control (CDC), IEEE 52nd Annual Conference on*. IEEE, 2013, pp. 5456–5461.
- [39] F. Sempé, A. Muñoz, and A. Drogoul, "Autonomous robots sharing a charging station with no communication: a case study," in *Distributed Autonomous Robotic Systems 5*. Springer, 2002, pp. 91–100.
- [40] M. C. Silverman, D. Nies, B. Jung, and G. S. Sukhatme, "Staying alive: A docking station for autonomous robot recharging," in *Robotics and Automation. Proceedings. ICRA. IEEE International Conference on*, vol. 1. IEEE, 2002, pp. 1050–1055.

- [41] H. Q. Raja and O. Scholz, "A case study on self-sufficiency of individual robotic modules in an arena with limited energy resources," in *ADAPTIVE, The Third International Conference on Adaptive and Self-Adaptive Systems and Applications*, 2011, pp. 29–35.
- [42] S. Jantz and K. L. Doty, "Learning to eat," in *Proc. 9th Conf. on Recent Advances in Robotics*. Citeseer, 1996, pp. 11–12.
- [43] F. Michaud, E. Robichaud, and J. Audet, "Using motives and artificial emotions for prolonged activity of a group of autonomous robots," in *Proceedings of the AAAI Fall Symposium on Emotions. Cape Code Massachussetts*, 2001.
- [44] A. Drenner and N. Papanikolopoulos, "Docking station relocation for maximizing longevity of distributed robotic teams," in *Robotics and Automation. ICRA. Proceedings IEEE International Conference on*. IEEE, 2006, pp. 2436–2441.
- [45] D. P. Bertsekas, "Auction algorithms." *Encyclopedia of optimization*, vol. 1, pp. 73–77, 2009.
- [46] D. F. Gleich and Y. Wang, "Bipartite\_matching," <http://www.mathworks.com/matlabcentral/fileexchange/24134-gaimc---graph-algorithms-in-matlab-code>, 2009.
- [47] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*. Courier Dover Publications, 1998.
- [48] F. Michaud and E. Robichaud, "Sharing charging stations for long-term activity of autonomous robots," in *Intelligent Robots and Systems. IEEE/RSJ International Conference on*, vol. 3. IEEE, 2002, pp. 2746–2751.
- [49] S. Sariel and T. R. Balch, "Efficient bids on task allocation for multi-robot exploration." in *FLAIRS Conference*, 2006, pp. 116–121.
- [50] T. Bektas, "The multiple traveling salesman problem: an overview of formulations and solution procedures," *Omega*, vol. 34, no. 3, pp. 209–219, 2006.
- [51] A. Muñoz and A. Drogoul, "Towards the design of self-vigilant robots using a principle of situated cooperation," in *Proceedings of the RoboCup Rescue Workshop*, 2000.

---

# Glossary

## List of Acronyms

|               |                                                          |
|---------------|----------------------------------------------------------|
| <b>CVRP</b>   | capacity constrained vehicle routing problem             |
| <b>FL</b>     | fuel location                                            |
| <b>FS</b>     | fuelling station                                         |
| <b>LP</b>     | linear program                                           |
| <b>MDCVRP</b> | multi-depot capacity constrained vehicle routing problem |
| <b>MFS</b>    | mobile fuelling station                                  |
| <b>MILP</b>   | mixed integer linear program                             |
| <b>MINLP</b>  | mixed integer nonlinear program                          |
| <b>MISOCP</b> | mixed integer second order cone program                  |
| <b>MR</b>     | mobile robot                                             |
| <b>MTSP</b>   | multiple travelling salesman problem                     |
| <b>RL</b>     | rendezvous location                                      |
| <b>TMP</b>    | the matching problem                                     |
| <b>TSP</b>    | travelling salesman problem                              |
| <b>UAV</b>    | unmanned aerial vehicle                                  |
| <b>VRP</b>    | vehicle routing problem                                  |
| <b>WP</b>     | waypoint                                                 |

## List of Symbols

|               |                                                                                      |
|---------------|--------------------------------------------------------------------------------------|
| $D$           | distance matrix, which contains all possible travel distances                        |
| $E$           | energy level                                                                         |
| $E^c$         | Energy consumption matrix                                                            |
| $\dot{E}$     | energy rate                                                                          |
| $\epsilon$    | $-\infty$                                                                            |
| $f$           | fuel location                                                                        |
| $F$           | refuelling event matrix                                                              |
| $\Gamma$      | refuel time matrix                                                                   |
| $\mathcal{J}$ | set of possible refuel orderings                                                     |
| $J$           | total number of possible refuel orderings                                            |
| $\mathcal{K}$ | set of mobile robots                                                                 |
| $K$           | total number of mobile robots                                                        |
| $\mathcal{L}$ | set of refuelling tasks for an individual fuelling station                           |
| $L$           | total number of refuelling tasks of an individual fuelling station                   |
| $\mathcal{M}$ | set of fuelling stations                                                             |
| $M$           | total number of fuelling stations                                                    |
| $N$           | total number of waypoints for an individual MR                                       |
| $\Omega_M$    | set that can contain any possible combination between two elements in $\mathcal{M}$  |
| $p$           | waypoint location                                                                    |
| $\mathcal{Q}$ | set that contains all possible combinations between the elements in $\mathcal{M}$    |
| $R$           | rendezvous location                                                                  |
| $\tau$        | total time a mobile robot is moving during a fuel cycle                              |
| $T$           | Total time of a refuelling event                                                     |
| $T$           | time matrix, which contains all possible travel times                                |
| $T^o$         | time a FS will be occupied until a MR is refuelled                                   |
| $T^w$         | time a MR has to wait in the queue of a FS                                           |
| $T^r$         | time it takes a mobile robot to be completely refuelled during a refuelling activity |
| $T^t$         | travel time                                                                          |
| $U$           | max-plus decision variable                                                           |
| $v$           | velocity                                                                             |
| $X$           | binary decision variable                                                             |
| $Y^e$         | set of locations where the mobile robots ended a fuel cycle                          |
| $Y^s$         | set of locations where the mobile robots started a fuel cycle                        |