Congestion control of robotic swarms in the common target problem: theory and algorithms

Yuri Tavares dos Passos, BSc, MSc
School of Computing and Communications
Lancaster University

A thesis submitted for the degree of
Doctor of Philosophy

November, 2023

In dedication to my wife, my supervisor, my family and friends.

Declaration

I declare that the work presented in this thesis is, to the best of my knowledge and belief, original and my own work. The material has not been submitted, either in whole or in part, for a degree at this, or any other university. This thesis does not exceed the maximum permitted word length of 80,000 words including appendices and footnotes, but excluding the bibliography. A rough estimate of the word count is 56.222.

Yuri Tavares dos Passos

Congestion control of robotic swarms in the common target problem: theory and algorithms

Yuri Tavares dos Passos, BSc, MSc.
School of Computing and Communications, Lancaster University
A thesis submitted for the degree of *Doctor of Philosophy*. November, 2023

Abstract

The common target problem in robotic swarms occurs when every robot must visit the same target region. This may happen when a swarm of robots has to collect a leaked toxic substance, for instance. To minimise congestion in such a situation, coordination algorithms for traffic control are required. This thesis presents a mathematical study about measures for evaluating the access efficiency of a common target area as the number of robots in the swarm rises: the target area throughput and the asymptotic throughput. Based on them, two algorithms were developed: the Single Queue Former (SQF) and the Touch and Run Vector Fields (TRVF). I also present approaches for ad hoc robots (i.e., those unaware of the coordination algorithm used by others) without having to learn about them: the Ad Hoc Follower (AHF) and the Mixed Teams (MT). Surprisingly, between the proposed approaches, simply executing an alternative algorithm is better than following another robot perceived by local sensors. In the tests performed in the Stage simulator, the other robots can use SQF or TRVF. These algorithms are tested as alternative algorithms in addition to no coordination (NC), where the robots only go directly to the target and avoid bumping each other. SQF significantly outperforms all algorithms for a higher number of robots or when the circular target region radius is small. Estimations of the expected task completion time for the presented algorithms are also studied. These estimations consider not only the number of robots as input but also environmental and algorithmic global variables, such as the common target area size, the average speed and the average distance between the robots. This work is a fundamental first step to start a discussion on how better approximations can be achieved and which mathematical theories about local-to-global analysis are better suited to this problem.

Acknowledgements

I express my gratitude to my supervisor, Leandro Soriano Marcolino, for his support and friendly guidance. His expertise helped me to communicate with clarity my ideas in this thesis.

I am grateful to the Faculty of Science and Technology at Lancaster University for providing me with the opportunity to conduct my research in the United Kingdom through studentship and for all of the resources and support they provided.

I would also like to thank Xavier Duquesne for his brilliant work in the initial phase of my research, Gao Peng and Abdulrahman Kerim for the review of the earlier drafts, and Ronghui Mu, Elnaz Shafipour Yourdshahi, Mario Escarce Junior, Lukasz Pelcner and Washington Luis de Souza Ramos for support and discussions.

I would also like to thank my friends and family for their love and assistance.

Publications

Parts of Chapter 1 and Section 3.1, the whole Chapter 4 and Section 4.6 were developed during the PhD course and published in the paper below.

Yuri Tavares dos Passos, Xavier Duquesne, and Leandro Soriano Marcolino (2022). "On the Throughput of the Common Target Area for Robotic Swarm Strategies". In: *Mathematics* 10.14, p. 2482. ISSN: 2227-7390. DOI: 10.3390/math10142482

In addition, an article composed of fragments of Chapter 1 and Section 3.1, the whole Sections 5.1, 5.2, 5.4.1.1 and 5.4.1.2 was accepted on the journal Robotics and Autonomous Systems:

Yuri Tavares dos Passos, Xavier Duquesne, and Leandro Soriano Marcolino (2023). "Congestion control algorithms for robotic swarms with a common target based on the throughput of the target area". In: *Robotics and Autonomous Systems* 159, p. 104284. ISSN: 0921-8890. DOI: 10.1016/j.robot.2022.104284

Finally, a paper was submitted to the Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS) about the task completion time estimations in Section 3.2, parts of Section 5.4.1.3 and Chapter 6. The sections about the AHF algorithm and MT (that is, Sections 3.3, 5.3, 6.1.4 and parts of Section 5.4.1.3) will be submitted for a conference yet to be determined.

Contents

List of Figures

List of Tables

List of Symbols

α Central angle region of the touch and run strategy. 50

α_l Angle of the α-area to switch to *locked* state for PCC and PCC-EE algorithms. 18

α_w Angle of the α-area to switch to *waiting* state for PCC and PCC-EE algorithms. 16

δ Radius of the α-area for PCC and PCC-EE algorithms. 16

δ_{comm} Communication radius for PCC and PCC-EE algorithms. 16

γ_{ad} maximum angle on the left and right of the target direction to follow another robot in the AHF algorithm. 95

ω Angle of the entry region for EE and PCC-EE algorithms. 18

ρ_I Probability of impatience for PCC and PCC-EE algorithms. 18

θ Hexagonal packing angle. 41

ξ Current robot orientation. 84

Alg Alternative algorithm employed by the robots using AHF when no other robot is detected. 95

$C_{MTAw}(p)$ Constant for the time estimation of an MT respective to the influence of the aware robots for a given ratio p of ad hoc robots. 162

$C_{MTNC}(p)$ Constant for the time estimation of an MT respective to NC influence for a given ratio p of ad hoc robots executing it. 162

C_{NC1} Constant for the time estimation of NC in the linear case when the number of robots is less or equal to the target area capacity twofold. 135

C_{NC2} Constant for the time estimation of NC in the case when the number of robots is more than the target area capacity twofold. 135

C_{SQF1} Constant for the time estimation of the SQF algorithm in the linear case when the number of robots is less or equal to the entry capacity. 146

C_{SQF2} Constant for the time estimation of the SQF algorithm in the squared case when there are more robots than entry capacity. 146

C_{TRVF} Constant for the time estimation of the TRVF algorithm. 154

d Minimum distance between the robots. 32

$\bar{d}$ The mean distance between the centre of mass of a robot and the others inside the influence radius I_d for all robots. 133

d_r Distance from the target centre for a robot to begin the turning on the curved path in the touch and run strategy. 50

E Maximum distance from the boundary of the working area for randomly positioning robots at the start of an experiment. 132

$f_h(t, \theta)$ Throughput of the hexagonal packing strategy for a given time and a hexagonal packing angle. 42

$f_h(\theta)$ Asymptotic throughput of the hexagonal packing strategy for a given hexagonal packing angle. 44

$f_h(t, u)$ Throughput of the hexagonal packing strategy for a given time and ratio u for the hexagonal packing angle θ^* which maximises $f_h(t, \theta^*, u)$. 57

f_p Asymptotic throughput of the parallel lanes strategy. 39

$f_p(t)$ Throughput of the parallel lanes strategy for a given time. 39

$f_p(t, u)$ Throughput of the parallel lanes strategy for a given time and a ratio u. 57

$f_t(K)$ Asymptotic throughput of the touch and run strategy for a given number of lanes. 54

$f_t(K, t)$ Throughput of the touch and run strategy for a given number of lanes and a time. 53

$f_t(K, t, u)$ The throughput of the touch and run strategy for a given number of lanes, a time and a ratio u. 58

$\mathbf{G} = (G_x, G_y)$ Current target centre position. 11

I_d The default influence radius. 12

I_{min} The minimum influence radius for the SQF algorithm. 81

K Number of lanes in the touch and run strategy and the TRVF algorithm. 51

K_a Constant magnitude of a force produced by an algorithm. 133

K_{AHF} Constant force magnitude for the AHF algorithm. 95

K_{dp} Control constant multiplied to the current robot velocity. 133

K_{EE} Constant force magnitude for the EE algorithm. 186

k_o Constant for exponentiation in the orbit following vector field algorithm. 86

k_ω Constant for proportional angular speed controller, also used as parameter in the straight-line and orbit following vector field algorithms. 84

K_{PCC} Constant force magnitude for the PCC algorithm. 186

$K_{PCC\text{-}EE}$ Constant force magnitude for the PCC-EE algorithm. 186

K_r Multiplicative constant for the repulsive force field. 133

K_{res} Constant force magnitude multiplied after the normalised vector addition of an algorithmic attractive force and a repulsive force. 133

k_s Constant for exponentiation in the straight-line vector field algorithm. 84

K_{SQF} Constant force magnitude for the SQF algorithm. 81

K_{TRVF} Constant force magnitude for the TRVF algorithm. 84

M Number of ad hoc robots. 91

N Number of robots. 31

p ratio of ad hoc robots by the total number of robots, that is, $\frac{M}{N}$. 162

$\mathbf{p} = (p_x, p_y)$ Current robot position. 11

r Turning radius of a lane in touch and run strategy. 50

r_γ Outer radius of the *danger* region for PCC, EE and PCC-EE algorithms. 16

r_σ Radius of the *free* region for PCC and PCC-EE algorithms. 16

s Radius of the common circular target area. 31

T_{comm} Number of iterations before sending a message for PCC and PCC-EE algorithms. 16

T_η Number of iterations for testing if a *waiting* robot will change state for PCC and PCC-EE algorithms. 18

u The ratio between the radius of the target region and the minimum distance between the robots. 56

v Constant maximum velocity. 32

$\bar{v}$ The mean linear speed of all robots. 133

Chapter 1

Introduction

1.1 Motivation

An agent is a software or hardware that acts autonomously in the environment (Russell and Norvig, 2020). The decision-making of a single agent considers only its environment with many distinct properties to take action. An autonomous vacuum cleaner choosing the next direction to clean and an intelligent non-player character used as an opponent in games for one person are examples of single agents.

However, there are situations where reasoning only with the information about the environment without regarding other agents may not work or be inefficient. Agents playing chess or multiplayer games have to consider other agents to succeed, for instance. Self-driving cars in a street need to consider other cars as agents, cooperating to minimise collisions and maximise the average speed on the road. With that in mind, a multi-agent system is composed of several agents that communicate with themselves explicitly or implicitly to solve a common task. Multi-agent systems have applications in many areas, such as in the prediction of cancers (Yong, 2013; Pourpanah et al., 2017; Tatari, Akbarzadeh-T, and Sabahi, 2012), students' performance (Abdullah, Malibari, and Alkhozae, 2014) and robotic intent (Demiris, 2007) or aiding decision-making in companies (Miley, 2018; Hanga and Kovalchuk, 2019; Caridi and Cavalieri, 2004).

In the literature, robots are often considered agents in the physical world. Agents can have the same characteristics, i.e., have the same hardware and software. In this case, the system is named homogeneous, otherwise heterogeneous. When a group of homogeneous or heterogeneous robots are used for a specific purpose, it is called a multi-robot system (for example, in robot soccer or a group of military autonomous drones). Such systems may be controlled by a single central processing unit (CPU), but if it fails, all robots in the system will stop functioning.

An alternative to avoid this situation is using different CPUs for controlling each

robot in the system distributedly. Accordingly, a multi-robot system with a large number of distributed homogeneous robots is known as a robotic swarm. Although this field was initially applied for validating biological models such as properties of insect colonies (Beni, 2005), the interest in engineering applications has attracted more researchers over the years (Dias et al., 2021). As it happens in insect swarms, no centralised coordination is performed for controlling each individual. Thus, in addition to a distributed control, local sensing is assumed in robotic swarms (Sahin, 2005; Beni and Wang, 1993).

In order to be more cost-effective, an increasing quantity of problems solved by robotic swarms are tackled using many simple robots rather than a few complex robots (Dudek et al., 1993). Counter-intuitively, some combinations of simple rules can create complex global behaviours (Mataric, 1995; Panait and Luke, 2003). Some robotic swarm research aim to create complex behaviours based on simple rules (Navarro and Matía, 2013; Garnier, Gautrais, and Theraulaz, 2007). Therefore, careful design can enable swarms of simple robots to perform complex tasks with robustness instead of using intricate and expensive robots.

Robotic swarms have the potential to be used in construction (Stewart and Russell, 2006), searching in regions harmful to humans (Arnold, Yamaguchi, and Tanaka, 2018), warehouse management (Wen, He, and Zhu, 2018; Yogi et al., 2020), microscopic robots locating cancer cells (Liu et al., 2015; Galstyan, Hogg, and Lerman, 2005; Bogdan, Wei, and Marculescu, 2012; Wang et al., 2022), collecting rubbish (Chaudhari, Patil, and Raut, 2019; Mărgăritescu et al., 2020) in remote areas such as the ocean and disaster response (Costa et al., 2022).

Having that in mind, this thesis focuses on a problem that may occur when robots of a swarm are moving to a common location. The next section introduces this problem, which is closely related to congestion problems in robotic swarms, discussed in Section 1.1.2. As congestion in Swarm Robotics studies can be further investigated, Section 1.1.3 elucidates why that topic should be theoretically analysed. Such a study not only broadens the understanding of congestion but inspires the creation of new algorithms for the common target problem. These algorithms may be used in another situation that needs the execution of different algorithms by different swarms. That situation is exemplified in Section 1.1.4. All algorithms approached here use artificial potential fields as local path planners. The analysis of the influence of a different potential field function on the overall swarm behaviour would benefit a swarm designer. Section 1.1.5 motivates the study of that subject.

1.1.1 The common target problem

The main problem treated in this thesis may happen when all robots have to access a shared place of any size. For instance, in a situation where robots are deployed to

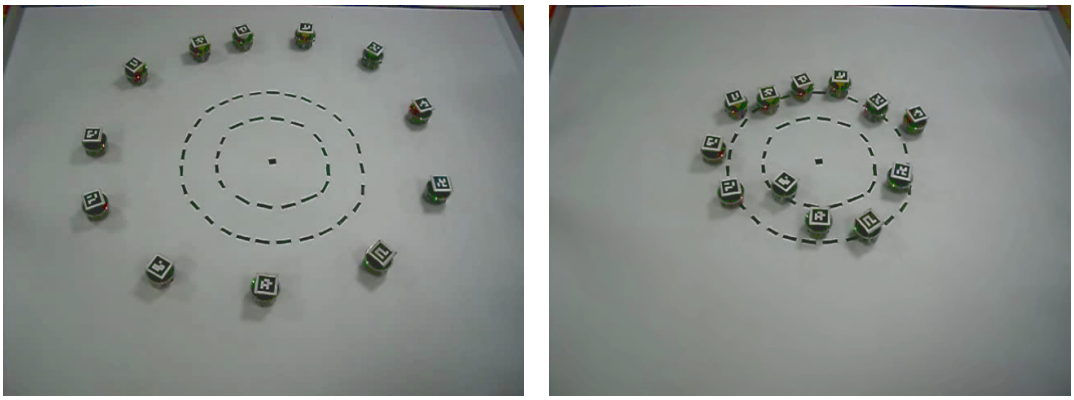
collect a leaked toxic substance, the robots need to coordinate themselves in order to get to that region as fast as possible while not interfering with the collection of the other robots. If the robots do not execute an efficient traffic control algorithm, their mutual interference can worsen.

The previous example illustrates the application of a control algorithm for robotic swarms when they need to reach a common target region. Thus, the common target problem occurs when a large number of robots in a swarm have to access the same area (Marcolino et al., 2017). Figure 1.1 illustrates a situation covered by the common target problem with few robots (Passos, 2012). In it, robots must reach the area inside the inner circle, leave that area and stop on one of the sides of the region at the end by using only attractive and repulsive potential fields and no control algorithm for minimising their interference. Figure 1.1a shows the initial configuration of that experiment. Figure 1.1b displays the robot going to the common target area. After some time (Figure 1.1c), the robots face congestion due to the absence of an algorithm to lessen it. After overcoming the obstruction, they finish the experiment by stopping on the sides (Figure 1.1d).

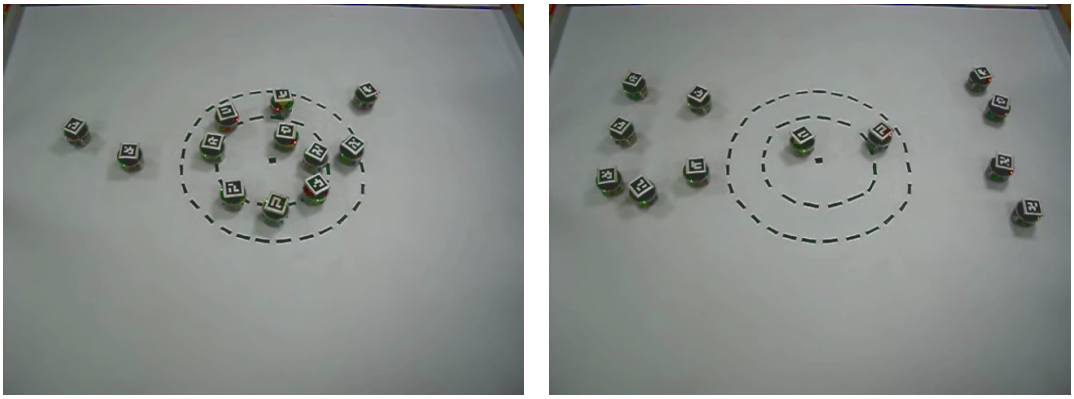
1.1.2 The congestion problem in robotic swarms

The robotic swarms in the problem described above can be seen as a multi-agent system with spatial computers, that is, a group of devices displaced in the space with its objective defined in terms of spatial structure, and its interaction depends on the distance between them (Giavitto et al., 2013). Swarms have recently been receiving attention in the multi-agents systems literature in problems such as logistics (Giordano et al., 2021), flocking formation (Cohen and Agmon, 2021), pattern formation (Cicerone et al., 2021) and the coordination of unmanned aerial vehicle swarms (Yang, Ma, and Xia, 2021). In such problems relating to spatial distribution, conflicts may be created by the trajectories of the robots, which may slow down the system, especially when a group is intended to go to a common region of space. Traffic congestion appears when several robots try to reach the same area simultaneously (Treuille, Cooper, and Popović, 2006; Graciano Santos and Chaimowicz, 2011; Yan, Jouandeau, and Cherif, 2013; Grossman, 1988), for instance, in waypoint navigation (Marcolino and Chaimowicz, 2008; Duarte et al., 2016) and foraging (Ducatelle et al., 2011; Fujisawa, Ichinose, and Dobata, 2019). Due to the local processing by the individuals in a swarm, they cannot see beyond their range of sensing and do not have global knowledge of every trajectory. Even if few of them had such global information, the processing power would not be sufficient when the number of robots grows beyond their capabilities. Thus, designing algorithms for local sensing to minimise congestion is needed in swarms.

In the current state of the art, congestion problems in robotic swarms are mainly



(a) (b)



(c) (d)

Figure 1.1: Example of the common target problem with a few robots (Passos, 2012). Available on `http://youtu.be/GvzgMhxjqk0`, accessed on 18 October 2023.

solved by collision avoidance in a decentralised manner because it enables better scalability of the algorithms (Batra et al., 2022; Majcherczyk et al., 2018; Borrmann et al., 2015). Despite that, only avoiding collisions does not necessarily lead to a good performance in the common target problem. For example, Marcolino et al. (2017) showed that using only the decentralised collision avoidance algorithm Optimal Reciprocal Collision Avoidance (ORCA) (Berg et al., 2011) creates local minima around the common target region. They present experimental solutions for the congestion problem that arises when the swarm shares the same goal. However, these algorithms may completely fail in small target regions, with an area measuring less than five times the area of the robots. In addition, no formal analysis of the cluttered environment is done in that work.

1.1.3 Theoretical analysis of the congestion in swarms

As discussed above, congestion in the Swarm Robotics context is still not well understood, and theoretical work is needed to measure the optimality of the algorithms. A better understanding of this topic should lead to a variety of new algorithms adapted to specific environments. Thus, this thesis introduces the first theoretical study on this problem, which should lead to future enhancements in handling congestion in robotic swarms.

Additionally, a robotic swarm system is desired to work well as the number of robots increases. As congestion in these systems grows with the number of robots, its analysis needs to incorporate this. If one has a finite measure which abstracts the optimality of any algorithm as the number of robots goes to infinity, it can be used as a metric to compare different approaches to the same problem. In this work, I present as a metric the target area throughput, defined as the inverse of the average time between arrivals at the target region, and theoretical strategies to maximise the throughput in scenarios where the robots have constant linear speed and distance between each other. Although these scenarios are useful for ease of mathematical analysis, they are not practical in real applications as inter-robot distances and speeds vary, but the presented strategies help to design new algorithms, such as some presented in this thesis: the Single Queue Former (SQF) and the Touch and Run Vector Fields (TRVF).

1.1.4 Ad hoc teamwork and the common target problem

As new algorithms such as those mentioned above are publicly available for controlling the congestion of swarms, there may be circumstances where different groups of swarms execute different algorithms for accessing the same target area. Consider a situation where a robotic swarm runs some coordination algorithm for accessing an area for rescuing people in an accident, and different institutions want to help. Assume that they deployed their robots and, for a fast response, did not meet to settle the swarm congestion control algorithm for quickly accessing the common area. Then, the first swarm to arrive will execute a congestion control algorithm, but the next does not know which algorithm is in execution. This situation is an example of *ad hoc teamwork*, i.e., when the system needs to solve a known objective, but the agents do not know each other's characteristics. To illustrate this, Stone et al. (2010) give a scenario example involving people in ad hoc teamwork. Assume that a transit accident occurred, and a group of passers-by's goal (the agents) is to help the injured. They do not know each other and discover by questioning their experiences that may help in this situation. People with any experience in nursing should perform first aid, and people with a telephone should call an ambulance or police, among other activities. Usually, the agents' task in this scenario includes coordinating with each

other and learning about themselves to achieve the goal. Approaches to this problem often involve learning about other agents (Mirsky et al., 2022; Albrecht and Stone, 2018), but this requires long iterations of training, and errors may be introduced during the learning process.

That said, a simple distributed approach using local sensing is proposed in this thesis to enable a group of robots in a swarm (the *ad hoc robots*) to minimise congestion in the common target problem without learning the coordination algorithm executed by the other robots (the *aware robots*). This approach results from experiments of a previous follower algorithm to solve this problem, which is summarised as follows. When ad hoc robots are not perceiving another robot in their neighbourhood, they follow a fixed alternative algorithm (for example, to go directly to the target). Otherwise, it follows the other robot while the angle between the vector pointed by the alternative algorithm and the direction vector of the other robot is not larger than a fixed angle. Accordingly, this approach does not need to estimate the algorithm of the other robots because they will follow the path of their potential field by getting in the proximity of one that knows the algorithm as probable as encountering a robot aware of the swarm congestion control algorithm. However, the experiments described in this work show that if the robots do not follow any robot, the time to complete the task performed in the experiments (*task completion time*) is minimised. Thus, our proposed approach is to execute a fixed alternative algorithm without following any robot.

1.1.5 Global analysis from local potential field specification

These proposed algorithms have in common the usage of artificial potential fields in the same manner as they are usually in robotic swarms due to their distributed and local characteristic for path planning (Cai et al., 2023; Barnes, Fields, and Valavanis, 2007; Galvez et al., 2017). Consequently, when the local individual specification based on potential fields is modified, global swarm behaviours, such as the task completion time, will have specific outcomes accordingly. The task completion time is the total time for all robots to arrive at the target area and leave a fixed-size area that includes the target area (the working area). In contrast with the throughput, which is associated with the arrival at the target, the task completion time includes information about congestion when leaving the target. Although state-of-the-art and the presented algorithms have less time to go away from the target, some algorithms cause congestion when leaving the target, such as one of the proposed solutions for ad hoc teamwork.

In order to compare distinct potential fields that work better for various numbers of robots, experiments with different values must be simulated to form a plot of the function of the task completion time versus the number of robots, environmental or algorithmic parameters (for instance, working area and target

area sizes). Nevertheless, plotting it for many robots through simulation is time-consuming. A swarm designer could use estimations instead of simulations to choose between available solutions. Additionally, such estimation is required to estimate the expected task completion time of algorithms that utilises those as a component, such as the proposed ad hoc teamwork algorithm.

An alternative approach would be using regression over the experimental data. However, it does not explain the time in relation to the number of robots, dynamical variables such as average speed, the average distance between the robots, and environmental or algorithmic variables such as the radius of the target area, radius of the task working area and how many directions the robots must go after they reach the target area. Therefore, this thesis also suggests and discusses such estimations.

1.2 Contributions and Objectives

The main objective of this thesis is to validate a new theoretical study of the common target problem in robotic swarms and new algorithms to control congestion when the robots in a swarm have a common target area.

Consequently, this thesis also has as contributions:

- the proposal of a method for evaluating algorithms for the common target problem in a robotic swarm by using the throughput in theoretical or experimental scenarios;
- the presentation of an extensive theoretical study of the common target problem, allowing one to understand better how to measure the access to a common target using a metric not yet used in other works on the same problem;
- assuming a circular target area and robots with a constant linear speed and a fixed distance from each other, the development of theoretical strategies for entering the area and calculating their theoretical throughput for a fixed time and their asymptotic throughput; additionally, the verification of the correctness of these calculations by simulations is shown;
- based on these theoretical strategies, the presentation of two novel algorithms for handling congestion: Single Queue Former (SQF) and Touch and Run Vector Fields (TRVF); SQF outperforms the state-of-the-art and can handle a small-sized target area which is a situation where previous approaches completely failed in our simulations, while TRVF helps us to understand the influence of the variation in the speed and distance between the robots when translating an idea from the theoretical strategies to concrete algorithms;

- the proposal of simple distributed approaches, the Ad Hoc Follower (AHF) and Mixed Teams (MT), for local sensing to enable ad hoc robots in a swarm to minimise congestion in the common target problem without installing on the robots or learning the coordination algorithm executed by the other robots;
- the estimations of the expected task completion time in relation to the number of robots, environmental or algorithmic parameters for a swarm with no coordination (that is, using only attractive and repulsive force fields), SQF and TRVF algorithms and MT.

This thesis is a fundamental step to initiate a discussion about how the global system behaviour from individual local controllers based on potential fields can be better inferred in the common target problem.

1.3 Thesis Organisation

This thesis is organised as follows.

- Chapter 2 presents the background: robotic swarms, artificial potential fields, the common target problem and the state-of-the-art algorithms for solving it.
- Chapter 3 discusses the related work.
- Chapter 4 explains the mathematical notation being used, formally defines the common target area throughput, proves statements about this measure for theoretical strategies that allow robots to enter the common target area and presents experiments and their results to verify the correctness of those strategies.
- In Chapter 5, these strategies are the inspiration for the two algorithms shown here, conveyed in the first sections of that chapter. It also presents the follower algorithm described above and the proposed approach for ad hoc robots in more detail. The results of experiments with them, comparisons with state of the art and how the ad hoc teamwork approaches behave when used for these algorithms is also discussed.
- Chapter 6 demonstrates the task completion time estimations for these algorithms, explains how they were obtained and presents the experiments and result for them.
- Finally, Chapter 7 summarises the results, provides final remarks and discusses future works.

Chapter 2

Background

In this chapter, subjects closely related to this thesis are presented: robotic swarms, artificial potential fields, the common target problem in a swarm of robots and state-of-the-art algorithms for that problem.

2.1 Robotic Swarms

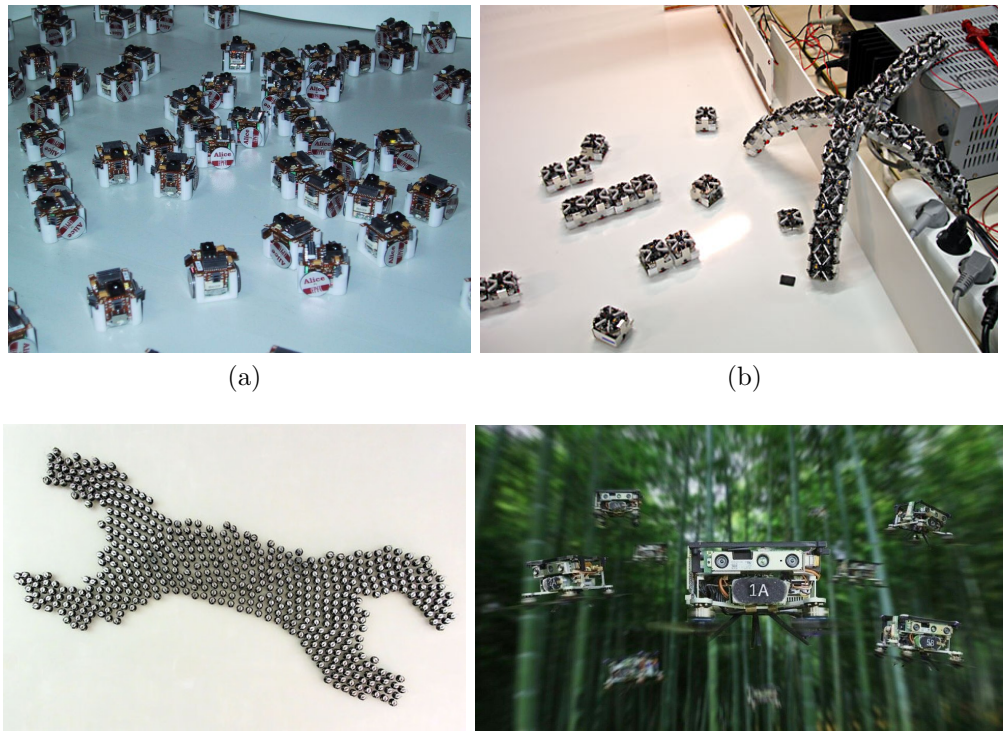
Robotic swarms are groups of numerous simple robots that work together to achieve a common goal (Shen et al., 2004). The behaviour of insect colonies, such as ants and bees, serves as inspiration (Kube and Zhang, 1993). Individuals in these colonies interact among themselves through independent processing and local communication: there is no central control unit overseeing each individual.

Influenced by that, the execution of a given task is decentralised. Each robot must process the data collected by its sensors and eventually communicate the results to the other robots in the swarm. Robots, in general, have limited processing and communication capabilities. Because of this constraint, algorithms must be robust and scalable, ensuring that the addition of more robots improves the efficiency of these systems (Sahin, 2005) or minimises their interference in applications that naturally lose its performance as the number of robots grows (Hamann, 2018). Decentralisation and local communication in multiple robot control results in increased system robustness and scalability.

According to Sahin (2005), robustness occurs when individuals fail in swarms, reducing the system efficiency but not preventing it from performing its task. Scalability is the ability to remain stable as the number of robots increases. A robotic swarm control algorithm must be designed so that the inclusion of more robots in the group tends to improve system performance whenever this is possible. A centralised processing unit cannot accomplish that because adding new robots would only increase its workload.

In addition, a swarm of robots is implemented to improve flexibility. The robots will execute in a manner that different situations generate different behaviours while still performing the primary task. An ant colony is an example of flexibility in nature. Its main goal is to collect food for the anthill, but sub-tasks must be completed for unusual situations arising during the main task execution (for example, diverting paths blocked by spilt liquids or objects left in the middle of the route, escaping predators and carrying heavy food).

Furthermore, robotic swarms have been tested in real-world experiments inside university laboratories and outdoors in the last few decades. Figure 2.1 shows examples of robotic swarms with real robots. Alice swarm system comprised of robots made by Caprari et al. (1998) in Figure 2.1a. Another swarm adapting itself to overcome obstacles by combining its robots is in Figure 2.1b from the project SYMBRION (Bristol Robotics Laboratory, 2017). A self-organising swarm of hundreds of robots (Rubenstein, Cornejo, and Nagpal, 2014) forms a wrench in Figure 2.1c. Figure 2.1d shows a rescue and search swarm of drones (Soria, 2022).



(a) (b)

(c) (d)

Figure 2.1: Examples of robotic swarms with real robots: (a) an Alice swarm (Jain, 2019), (b) a combining swarm (Golden, 2021), (c) a self-organising swarm forming a wrech (Yong, 2014) and (d) a rescue and search swarm of drones (Aubourg, 2022).

To conclude, recent advances in this area have revealed potential applications such as military missions in the air (Wu, Xiao, and Bi, 2020) or complex urban environments (Rudd, 2017), space exploration (stimulated by programs such as the Swarmathon Challenge from NASA (Aeronautics and NASA, 2018; Aeronautics and NASA, 2015)) and patrolling (Kobayashi, Higuchi, and Ueno, 2023). Current advances and predictions of applications until 2050 are discussed by Dorigo, Theraulaz, and Trianni (2020): agriculture, infrastructure inspection and maintenance, maritime and deep-sea applications (for example, ecological monitoring, surveillance and fishing), entertainment, cleaning, grazing, delivering goods, space exploration, pest control, microplastic collection in aquatic environments and medical application (for instance, drug delivery).

2.2 Artificial Potential Fields

Created by Khatib (1985), this method assumes that the position to be reached is an attractive pole (as an electrical charge in the centre of its electrical potential field) and the obstacles are repulsive surfaces. The robot position is a point-like ball moving by the resultant force originated by that attractive and repulsive potential fields.

Formally speaking, let $\mathbf{p} = (p_x, p_y)$ and $\mathbf{G} = (G_x, G_y)$ be the robot and goal positions. An attractive potential field is denoted $U_A(\mathbf{p}, \mathbf{G})$ and a repulsive potential field, $U_R(\mathbf{p}, \mathbf{q})$ for the nearest obstacle edge point located at position $\mathbf{q} = (q_x, q_y)$. The resultant potential field is

$$U(\mathbf{p}) = U_A(\mathbf{p}, \mathbf{G}) + U_R(\mathbf{p}, \mathbf{q}),$$

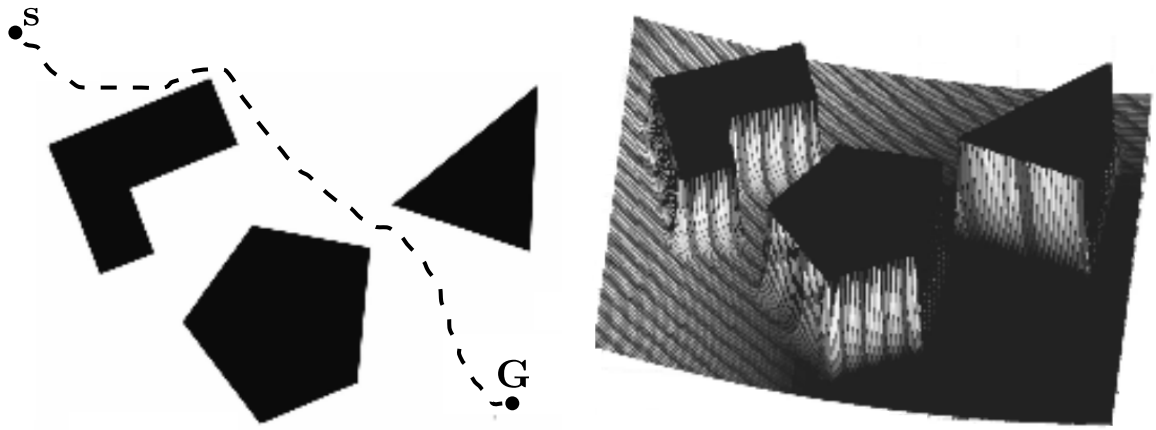
and the resultant force for a given robot position is

$$\mathbf{F}(\mathbf{p}) = -\nabla U(\mathbf{p}).$$

Figure 2.2 illustrates an example by Siegwart, Nourbakhsh, and Scaramuzza (2011). A robot is initially located in position $\mathbf{s}$ and has to move to position $\mathbf{G}$ (Figure 2.2a). For this example, the surface map of $U(\mathbf{p})$ is shown in Figure 2.2b. Observe that the obstacles have high altitude while the places without obstacles have a slope going towards the goal position from the starting point. The path to be followed in Figure 2.2a is obtained from that with lowest energy. It can be locally calculated by the robot as the potential field $U(\mathbf{p})$ and its gradient only depends on the robot position, assuming the goal and obstacles are fixed. This local feature is the main reason for using it as path planner for robotic swarms.

In this thesis, the attractive and repulsive potential fields are (Siegwart, Nourbakhsh, and Scaramuzza, 2011)

$$U_A(\mathbf{p}, \mathbf{G}) = \frac{1}{2} K_a \|\mathbf{p} - \mathbf{G}\|^2$$



(a) (b)

Figure 2.2: Example of artificial potential field, adapted from Siegwart, Nourbakhsh, and Scaramuzza (2011). (a) Initial setup and followed path. (b) Resultant potential field surface from goal and obstacles.

and

$$U_R(\mathbf{p},\mathbf{q}) = \begin{cases} \frac{1}{2}K_r\left(\frac{1}{\|\mathbf{p}-\mathbf{q}\|}-\frac{1}{I}\right)^2, & \text{if } \|\mathbf{p}-\mathbf{q}\| \le I, \\ 0, & \text{otherwise,} \end{cases}$$

where K_a and K_r are scaling factors, and I is the distance of influence of the object (i.e., the maximal distance from the robot position to the obstacle for avoidance). Algorithms may change the I value in this work. Hereafter, I_d and I denote a fixed default influence radius from the robot position and a variable one, respectively. The resultant force is obtained by negating the gradient of the total potential field $U(\mathbf{p})$ obtained from the above potential fields:

$$\mathbf{F}(\mathbf{p}) = -\nabla U(\mathbf{p}) = -\nabla U_A(\mathbf{p},\mathbf{G}) - \nabla U_R(\mathbf{p},\mathbf{q}) = \mathbf{F_A}(\mathbf{p}) + \mathbf{F_R}(\mathbf{p}),$$

for

$$\mathbf{F_A}(\mathbf{p}) = K_a(\mathbf{G}-\mathbf{p})$$

and

$$\mathbf{F_R}(\mathbf{p}) = \begin{cases} K_r\left(\frac{1}{\|\mathbf{p}-\mathbf{q}\|}-\frac{1}{I}\right)\frac{\mathbf{p}-\mathbf{q}}{\|\mathbf{p}-\mathbf{q}\|^3}, & \text{if } \|\mathbf{p}-\mathbf{q}\| \le I, \\ \mathbf{0}, & \text{otherwise.} \end{cases}$$

Nonetheless, artificial potential fields may have local minima, that is, some configurations of the goal and the obstacles may put the robot in a position with zero force. Figure 2.3 illustrates a situation of local minima: repulsive and attractive forces are of the same magnitude but in opposite directions, resulting in a zero resultant

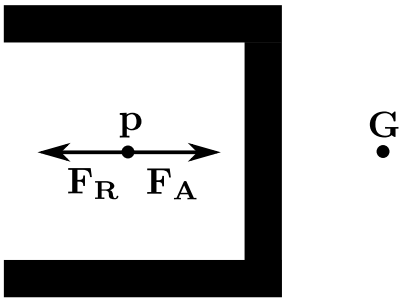


Figure 2.3: Example of local minima.

force. Although there are solutions to avoid them (Mabrouk and McInnes, 2008; Matoui, Boussaid, and Abdelkrim, 2015), this situation rarely occurs in this work because the obstacles are other moving robots.

2.3 Common Target Problem

The common target problem for robotics swarms occurs when N robots have the same target area. They must cooperate in a decentralised manner and may use local communication to enable them to reach the target, avoiding congestion, accessing and leaving that area in the shortest possible time.

That congestion problem could be solved using a central processing unit to compute the best trajectories for each robot. However, an obvious disadvantage would be the system dependency on this unit. Additionally, it is not a scalable solution for various robots. The challenge of its solution for swarms is the guarantee that the whole system will work without each part being aware of the global environment. Hence, the solutions must be decentralised and with local sensing.

Closely related to such solutions is robotic traffic control. This topic has been studied for a long time (Grossman, 1988; Kato, Nishiyama, and Takeno, 1992; Caloud et al., 1990), but assuming robots navigating in delimited lanes and coordination is necessary only at the intersections. Other more recent works still focus on alleviating congestion in delimited lanes and circuits (Masud et al., 2020; Hoshino and Seki, 2013; Hoshino, 2011; Viswanath and Madhava Krishna, 2009). For instance, an algorithm based on Petri nets has been introduced to avoid deadlocks at intersections (Zhou et al., 2017), where each robot follows a pre-determined closed path, but it works only for robots on lanes and does not show performance in terms of simulation time nor throughput, as in Chapters 4, 5 and 6. Saska et al. (2020) developed a control algorithm for motion planning of formations for unmanned aerial vehicles in environments with narrow passages. The robots in that system follow a leader and maintain proximity while avoiding obstacles in a constrained space. Yoshimoto

et al. (2018) describe a decentralised algorithm to maintain proximity between robots. Although their work is designed for a robotic swarm, it depends on a leader, which is a robot initially chosen and remains the same during the experiments.

Also related to the congestion problem, there are many relevant works in the multi-agent systems literature, but assuming autonomous cars that navigate following lanes or roads, and coordination is needed at the junctions (Carlino, Boyles, and Stone, 2013; Sharon and Stone, 2017; Cui et al., 2021). Other lines of work try to optimise trajectories across edges of complex traffic networks, for example, by global planning and using incentives or tolls for self-interested agents (Sharon et al., 2017; Sharon et al., 2018). Choudhury et al. (2021) and Shahar et al. (2021) also deal with multi-agents and pathfinding, but not in a situation where the target of every agent is the same area. Furthermore, distributed solutions are considered here where agents only have local information, while Choudhury et al. (2021) and Shahar et al. (2021) propose centralised solutions. Xia, Sun, and Xia (2021) investigate the topology of the neighbourhood relations between multiple unmanned surface vehicles in a swarm. They deal with maintaining formation in swarms, but they have to keep virtual leaders, and their goal is not to minimise congestion. Cenedese, Favaretto, and Occioni (2016) use the Kuramoto model to coordinate multiple vehicles towards a target while avoiding collisions and keeping them next to each other, but they do not analyse the time to get to the target and exit from it as the number of vehicles grows, as in this thesis, and their model uses communication between agents, which is unneeded by the proposed algorithms in Chapter 5. Ma, Ouimet, and Cortés (2020) describe a reinforcement learning algorithm for multi-agent planning for a swarm of vehicles to go to objectives distributed over space, but it does not perform well when the targets cannot be well divided into regions over the environment.

Although minimising congestion by traffic control is a sub-problem in robotic swarm applications, few works deal with it because a small number of robots are often used in real-world swarms (Tarapore, Groß, and Zauner, 2020), and congestion is not an issue in such works. The problem of alleviating congestion when a swarm of robots has a common target has not yet been well studied. Thorough surveys about Swarm Robotics (Sahin, 2005; Sahin et al., 2008; Barca and Sekercioglu, 2013; Brambilla et al., 2013; Bayındır, 2016; Chung et al., 2018) do not discuss these situations. Additionally, a recent survey on collision avoidance (Hoy, Matveev, and Savkin, 2015) provides insights into multiple vehicle navigation, for instance, the model predictive control (MPC) – also known as Receding Horizon Control (RHC). Their work also discusses algorithms for two- and three-dimensional kinematics, which affect the swarm coordination algorithms presented in this thesis from the individual level.

In order to show the influence of different kinematics, the experiments in this thesis were applied to holonomic and non-holonomic robots. A robot is holonomic

when the number of dimensions in the velocity space is equal to the degree of freedom of its workspace (Siegwart, Nourbakhsh, and Scaramuzza, 2011). A holonomic robot can freely move in any direction, while a non-holonomic robot has restrictions (for example, car-like robots cannot skid). Hence, three-dimensional robots will influence the algorithms differently, and the algorithms in Chapter 5 should be adapted and extended to these types of robots. However, they do not discuss the common target problem in multi-robot navigation.

As an alternative to these solutions, the first algorithm to minimise congestion for the common target problem was presented by Marcolino and Chaimowicz (2009), which propose an algorithm using potential fields. Their solution was improved later by Marcolino et al. (2017), giving birth to the algorithms Probabilistic Congestion Control (PCC), Entrance and Exit Regions (EE) and the combination of both (PCC-EE), which will be described in more detail below and whose pseudocode are in Appendix A. These algorithms have also been the inspiration for new algorithms since their publication. For instance, inspired by the PCC algorithm, Oliveira, Inácio, and Macharet (2017) developed an algorithm for the common target problem in distinct groups of robots in a swarm to minimise the difference between the time of arrival of individuals that belong to the same group. Brown and Adams (2023) analysed the congestion in a heterogeneous swarm formed by unmanned ground and aerial vehicles in the military program of the Defense Advanced Research Projects Agency (DARPA) named OFFensive Swam-Enabled Tactics (OFFSET) and suggested, without testing, the usage of a variant of the probabilistic state machine similar to the employed in the PCC algorithm for congestion control of the aerial vehicles which assigns a random wait time before they are set in motion. Table 2.1 summarises these works.

Reference	Summary
Marcolino and Chaimowicz (2009)	First mention of the problem and description of an earlier version of the PCC algorithm.
Marcolino et al. (2017)	Description of PCC, EE, PCC-EE algorithms. Tests with ORCA fail for the common target problem.
Oliveira, Inácio, and Macharet (2017)	Presentation of an algorithm for the common target problem in distinct groups of robots.
Brown and Adams (2023)	Discussion of algorithms for a congestion analysis of swarms in a military application, including the solutions for the common target problem.

Table 2.1: Summary of the works on the common target problem.

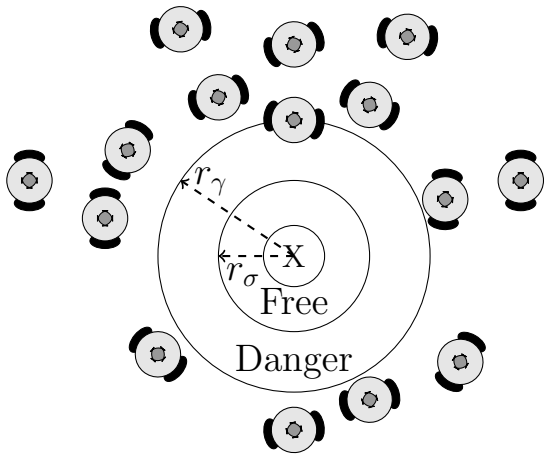


Figure 2.4: Free and danger regions. "X" denotes the target area. Adapted from Marcolino et al. (2017).

2.3.1 Probabilistic Congestion Control

The Probabilistic Congestion Control (PCC) minimises congestion by making some robots wait randomly before going to the target area. The PCC algorithm utilises two regions around the target: the *free* region, a circular region with radius r_σ and the *danger* region, a ring-shaped region with inner radius r_σ and outer radius r_γ (Figure 2.4). Cooperation is required for robots reaching the danger region in order to reduce access to the free region. After entering this region, they go to the target.

In addition, the algorithm utilises an α-area, which is a sub-area in the robot sensor region. Assume a coordinate system centred at the robot position with the y-axis pointing towards the target. The α-area is the circular sector $[-\alpha, \alpha]$ centred in y with radius δ (Figure 2.5). The circle around the robot position with radius δ is used to detect the presence of another. When a robot detects the presence of another, it sends a message saying its target and current state, but only after every T_{comm} iterations, to decrease the number of messages sent. The communication radius for sending such a message is δ_{comm}.

A probabilistic finite state machine models the behaviour of each robot (Figure 2.6). Its states are *normal* (the starting state), *waiting*, *locked* and *impatient*. A *normal* robot moves in the direction of the target while avoiding collisions. When a robot in the *normal* state is in the danger region and detects another robot, the *normal* robot checks if the other is within its α-area and if they have the same target. If both conditions are met, there is a transition from *normal* state to *waiting*. For the verification of the α-area in the latter condition, the constant α is called α_w.

A *waiting* robot tries to remain stationary at the point where it changed its state while simultaneously avoiding collisions. That robot periodically checks if it can make

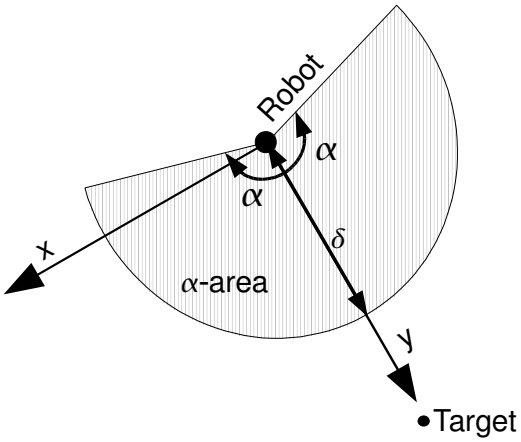


Figure 2.5: An α-area of the PCC algorithm (Marcolino et al., 2017).

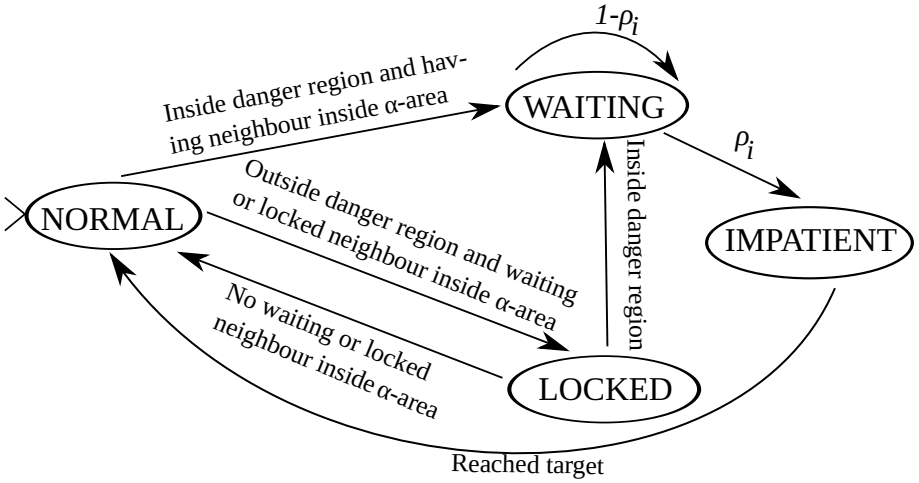


Figure 2.6: Probabilistic finite state machine of the PCC algorithm. Adapted from Marcolino et al. (2017).

a state transition every T_η iterations. It can switch to the *impatient* with probability $\rho_I > 0$ or stay in the same state with probability $1 - \rho_I$.

An *impatient* robot moves in the direction of the target as a *normal* robot, but it does not stop nor alter its state until it reaches the target. After that, it switches to *normal* and moves towards its next destination.

In Figure 2.6, a *normal* robot can also change its state to *locked* to force the robots outside the danger region to wait before being allowed to enter the target area. For that transition, the robot has to check if another robot is *locked* or *waiting* only in its front, so the α-area can be narrower than the area used to switch from *normal* to *waiting*. Thus, the constant for that area is denoted α_l.

A *locked* robot does not move (similar to a waiting robot), but the transition from the *locked* state does not depend on probabilities. A *locked* robot switches back to *normal* when no *waiting* or *locked* robot is in its α-area. Additionally, if a *locked* robot is pushed into the danger region, it becomes a *waiting* robot. Moreover, a robot sends a message if it is in the danger region or in *locked* state.

2.3.2 Entrance and Exit Regions

The Entrance and Exit Regions (EE) algorithm separates the area around the target into *entry* and *exit* regions. A circular area centred in the target centre is divided into four circular sectors: two defined by angle ω for entering the target area and two by angle $\beta = \pi - \omega$ for leaving. The entry region is a circular area with radius r_γ around the target centre, as shown in Figure 2.7.

The EE algorithm affects the robots in the ring-shaped region defined by the inner radius r_γ and outer radius D. When a robot is moving towards the target and is within a distance D from it, the robot verifies whether it is outside the entry region. If the robot is not in the entry region, it is compelled to move to the nearest point in the border of the entry region. This verification is equivalent to

$$\begin{aligned}(p_y - G_y \geq \tan(\phi)(p_x - G_x)) \wedge (p_y - G_y \geq \tan(\psi)(p_x - G_x)) \wedge (p_y > G_y) \vee \\ (p_y - G_y \leq \tan(\phi)(p_x - G_x)) \wedge (p_y - G_y \leq \tan(\psi)(p_x - G_x)) \wedge (p_y \leq G_y),\end{aligned} \tag{2.1}$$

for a robot position $\mathbf{p} = (p_x, p_y)$, target centre at $\mathbf{G} = (G_x, G_y)$, $\phi = \pi/2 - \omega/2$ and $\psi = \pi/2 + \omega/2$, as illustrated in Figure 2.8. From that figure and using trigonometry, equations of a line and the distance of a point to it, the waypoint $\mathbf{w} = (w_x, w_y)$ to where the robot has to go is given by

$$\begin{aligned}w_x &= p_x + d\cos(\vartheta) \\ w_y &= p_y + \tan(\theta)(w_x - G_x),\end{aligned} \tag{2.2}$$

for

$$d = \frac{p_y - \tan(\theta)(p_x - G_x) - p_y}{\sqrt{(\tan^2(\theta) + 1)}},$$

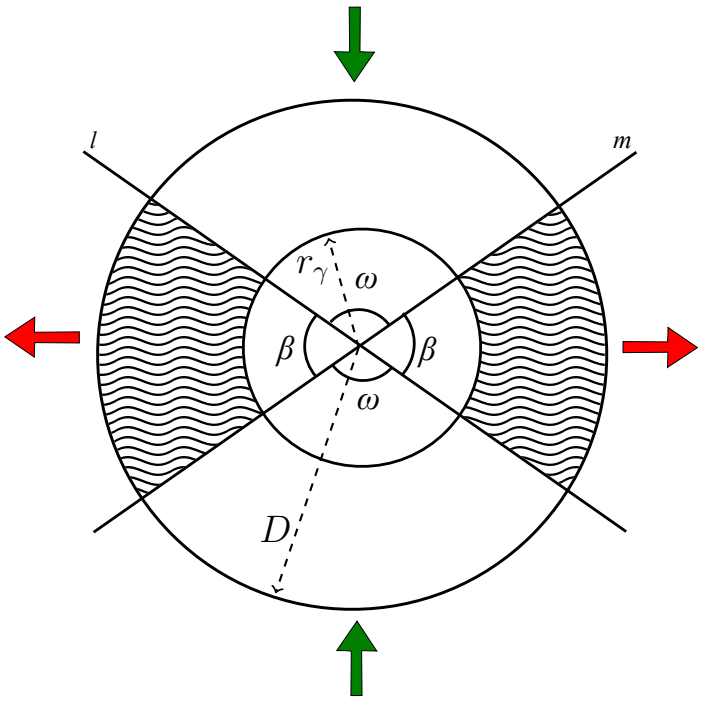


Figure 2.7: Division of the region around the target in entry (white) and exit (shaded) areas. Adapted from Marcolino et al. (2017).

$$\vartheta = \begin{cases} \pi/2 + \phi, & \text{if } ((p_y > G_y) \wedge (p_x > G_x)) \vee \\ & \quad ((p_y \leq G_y) \wedge (p_x \leq G_x)) \\ \psi - \pi/2, & \text{otherwise,} \end{cases}$$

and

$$\theta = \begin{cases} \phi, & \text{if } ((p_y > G_y) \wedge (p_x > G_x)) \vee \\ & \quad ((p_y \leq G_y) \wedge (p_x \leq G_x)) \\ \psi, & \text{otherwise.} \end{cases}$$

Additionally, robots must also react to repulsive forces relative to their neighbours. In the entry region, repulsive forces applied to a robot are divided by half if they push it outside that region. The repulsive force vector is reduced by checking if it crosses the nearest delimiting line from inside to outside the entry area (Figure 2.9). Upon reaching the target, the repulsive forces induced by other robots are also divided in half. After a robot in the entry region reaches the target, it will leave it by the exit region without congestion caused by robots in the opposite direction. It is unexpected that robots are in the opposite direction, as the robots going to the target area in the exit area are induced to go to the entry area.

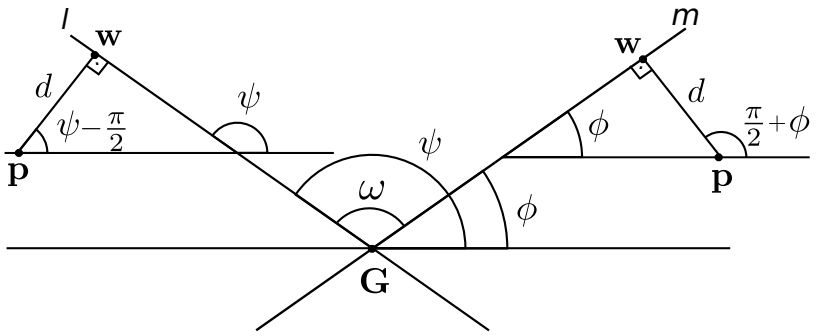


Figure 2.8: Nearest distance between a point in the *exit* region and the *entry* region boundary. Adapted from Marcolino et al. (2017).

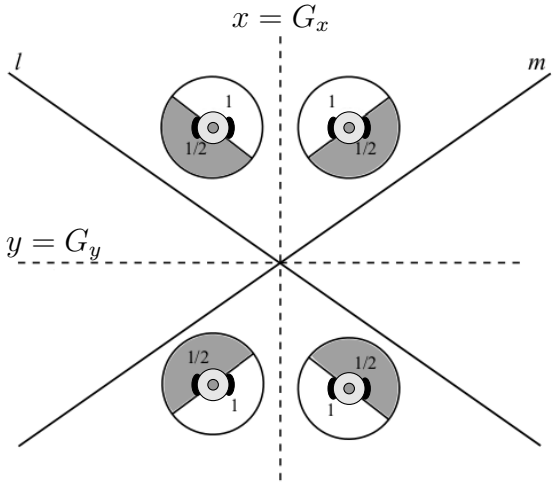


Figure 2.9: Forces that push the robots away of the *entry* region are divided by half in the grey semicircles. Adapted from Marcolino et al. (2017).

2.3.3 Probabilistic Congestion Control with Entrance and Exit Regions

The combination of the two previous algorithms results in the Probabilistic Congestion Control with Entrance and Exit Regions (PCC-EE). The circular region of radius r_γ for the EE algorithm corresponds to the danger region (of the same radius) of the PCC algorithm.

Robots follow both algorithms in the entry region but only the EE algorithm in the exit region. They must verify if they are inside the entry region while moving towards the target. If they are outside, they move towards the nearest point in the entry region. Inside the entry region, the robots also follow the PCC algorithm. Therefore, they can change their state to either *waiting*, if in the danger region, or *locked*, if there is a robot in the *waiting* or *locked* state in front of them.

Chapter 3

Related work

The related work is divided into three major groups as seen below: congestion and traffic in robotic swarms, supporting Chapter 4 and the new algorithms in Sections 5.1 and 5.2; studies of Swarm Robotics and other areas on the local to global analysis related to Chapter 6; and the definition of ad hoc teamwork, applications and its relation with multi-robot systems and robotic swarms aiding the solutions presented in Section 5.3 and estimated in Section 6.1.4.

3.1 Traffic and Congestion Problems in Robotic Swarms

In the literature on multiple robot systems, collision avoidance is closely related to our problem, but works on this topic do not measure the efficiency concerning a common target. Also, heed that collision avoidance alone is not enough to solve the common target problem. As presented shortly ahead, there is a lack of theoretical papers about this problem in robotic swarms. Thus, along with the theoretical work in Chapter 4, I suggest a measure that the control algorithm should maximise for getting to the target area. Thus, works with similar measures are discussed here, but they differ from the common target problem. Finally, this problem is treated mathematically in Chapter 4, whose described strategies inspired the algorithms presented in Sections 5.1 and 5.2.

Works in Collision Avoidance. Collision avoidance is an important related topic of study in robotics. In the found literature, algorithms let robots move efficiently in environments with a lot of obstacles, but they do not measure the efficiency of the algorithm in the common target problem (Hewawasam, Ibrahim, and Appuhamillage, 2022). As an example, Ferrera et al. (2017) have proposed a decentralised algorithm where a set of local reactive rules are followed to avoid

collisions. Their work shows an extensive list of benchmarks and metrics for robotics problems, but performance in common target situations is not included. Evolution-based algorithms may learn traffic rules for collision avoidance after thousands of iterations (Fujisawa, Ichinose, and Dobata, 2019), but they were not yet explored for the common target problem. Additionally, Sections 5.1 and 5.2 focuses on algorithms directly designed to handle congestion based on our theoretical work in Chapter 4 without requiring long iterations of training or optimisation.

Inadequacy of Employing Only Collision Avoidance. However, only avoiding collisions does not necessarily lead to a good performance in the common target problem. For instance, Marcolino et al. (2017) showed that the ORCA algorithm (Berg et al., 2011) reaches an equilibrium where the robots cannot go to the target. They proposed three algorithms using artificial potential fields and offered experimental solutions for the target congestion problem. However, their proposed algorithms face issues as the target area gets smaller. That paper also presents three algorithms using artificial potential fields for the common target congestion problem, but no formal analysis of the cluttered environment was conducted.

Lack of Theoretical Studies on Congestion in a Common Target Area. Therefore, congestion is still not well understood, and more theoretical work is needed to measure the optimality of the algorithms. A better understanding of this topic should lead to a variety of new algorithms adapted to specific environments. Thus, Chapter 4 aims to discuss the first theoretical study on this problem, which should lead to future enhancements in handling congestion in robotic swarms. That chapter is a theoretical work which fits in the literature on mathematical models of Swarm Robotics, such as the works by Lima and Oliveira (2017), which models a cellular automata ant memory to control a robotic swarm for foraging tasks; Varghese and McKee (2010), for pattern transformation modelling; Li and Chen (2006), for box-pushing; Taylor-King et al. (2015), which studies the effect of turning delays on the behaviour of groups of robots; Galstyan, Hogg, and Lerman (2005), for microscopic robots that reside in a fluid and can detect chemicals; Khaluf and Dorigo (2016), which models swarm performance measures using the integral of linear birth–death processes; and Mannone, Seidita, and Chella (2022), which uses category theory and quantum computing to model the development of robotic swarm systems. However, as mentioned, these theories do not yet allow one to better understand swarm congestion.

The Common Target Area Throughput and Similar Studies. Accordingly, any elaborated analysis on that subject must investigate the effect of the increase in the number of individuals on the swarm congestion, as it is desirable for the system to perform well as it grows in size. If one has a finite measure that abstracts the optimality of any algorithm as the number of robots goes to infinity, this can be used as a metric to compare different approaches to the same problem. Thus, this work presents as a metric the common target area throughput. That is, a measure of the

rate of arrival in this area is proposed as the time tends to infinity as an alternative approach to analyse the congestion in swarms with a common target area. In network and parallel computing studies (Daduna, Pestien, and Ramakrishnan, 2003; Hockney, 1994), asymptotic throughput is used to measure the throughput when the message size is assumed to have infinite length. The same idea is used here, but instead of message size, it is applied with infinite time, as if the algorithms run forever. As it will be presented in the next chapter, this implies dealing with an infinite number of robots. Thus, time is being used here instead of message size or bytes, as in computer network studies.

3.2 Many-Body Problem and Robotic Swarms

The effect of the task completion time of a robotic swarm based on the dynamic of one individual is an example of an analysis from local to global. Similar works can be found in other study areas. In special, the many-body problem in Physics has elegant and strong mathematical solutions. Some works for traffic problem or Swarm Robotics have applied solutions for the many-body problem. However, the solutions presented there are not adequate for the common target problem. All algorithms developed for the common target problem use potential fields as a local path planner in order to emerge global congestion control. These works will now be discussed in detail.

Studies of Local-to-Global. Biology, Chemistry and Physics share the study of individual parts of different systems to explain their global behaviour. The manifestation of emergence (Boogerd et al., 2005; Holland, 2000; Feltz, Crommelinck, and Goujon, 2006; Couzin, 2023) is a good example of such studies. In it, the microscopic characteristics enable the arising of unexpected macroscopical behaviour. As the understanding of these relations grows, advances in other areas like engineering and robotics are made. Related to that, the interdisciplinary Synergetics (Haken, 1983) explain self-organised phenomena in systems with non-equilibrium such as cloud dissipation, lasers and patterns in slime mould or chemical reactions.

The Many-Body Problem and Solutions. In Physics, problems about the interaction of microscopic particles is generally mentioned as the *many-body problem* (March, Young, and Sampanthar, 1995; Kuzemsky, 2017). For instance, in statistical mechanics, the Maxwell-Boltzmann distribution function (Mandl, 2013; Huang, 2008) states the speed probability distribution of ideal gas particles. It is calculated by analysing the movement of the particles under the influence of gravity and the probability frequency of the energy from the distance to the ground. Then, the velocities are averaged over all directions and that probability frequency to yield the desired distribution. There are different approaches to this problem. The perturbation theory (Kato, 2012; Holmes, 2015) starts with a simpler version of the problem. Then,

an expression, often called perturbative expansion, using power series of successive corrections to the simpler problem is developed to solve the original problem. The coupled cluster technique is used in computational chemistry (Bartlett and Musiał, 2007; Shavitt and Bartlett, 2009) for solving this problem numerically. Similarly, the density-functional theory (Glossman-Mitnik, 2019; Dreizler and Gross, 1995) is applied in computational modelling to understand the electronic and molecular organisation from atoms to solids by using functionals of electron density that depend on the space. The mean-field theory (Opper and Saad, 2001) substitutes the high number of interactions between the particles with a few averaged ones, reducing the number of calculations but also the exactness of the solution. There are interesting applications of this method in multi-agent research such as Chen, Georgiou, and Pavon (2018) and Lasry and Lions (2007). The variational method (Jiao et al., 2022; Pérez-Torres, 2019) is a solution similar to that used in quantum mechanics for approximating the desired energy state by selecting wave functions whose parameters minimise the expected value of the energy. The neural network quantum states (Carleo and Troyer, 2017) approximate the solution using machine learning to minimise the variational energy.

Similar to Many-Body in Swarm Robotics. Additionally, the literature on many-body problem is mostly Physics-oriented. An interesting approach to the intersection of Physics and Swarm Robotics is described by Hamann (2018), where he applied Brownian motion, Langevin equation and Fokker-Plank equation for estimating global behaviour from individual robot specification by random movement. Another approach to the above solutions for the many-body oriented to real-world application is the Herman-Prigogine kinetic equations for vehicular traffic (Prigogine and Andrews, 1960; Iannini and Dickman, 2016; Prigogine and Herman, 1971), considering the drivers with intentions but random actions. However, as these approaches give results assuming that every action of every driver or particle are random as the time passes, their equations cannot be applied directly in this work because only the initial position is random, but all the movement is deterministic. The survey of Elamvazhuthi and Berman (2019) also present another work on Swarm Robotics and Physics aligned with our problem, but the references provided by Elamvazhuthi and Berman (2019) do not answer our research question directly. Instead of assuming randomness in the movement of the individual robots and showing purely theoretical deductions, the estimations calculated here use experimental observation from simulations performed in Stage (Gerkey, Vaughan, and Howard, 2003). Additionally, these estimations are a fundamental first step to start a discussion on theoretical aspects on the relation of time and the number of robots on the common target problem.

The Robotic Swarm Common Target Problem and Algorithms. Moreover, as explained in Section 2.3, the common target problem is usually handled by

using potential fields. In fact, the algorithms discussed in Chapter 5 use them in the role of an individual path planner to accomplish global congestion control. Any of these algorithms modify the attractive and the repulsive force field to minimise the time to access the common target area or the task completion time, which, in addition to the time to access it, includes the time for leaving it. This may bring to light the question about whether it is possible to find the exact attractive potential field that minimises this time by a method similar to the Lagrangian in mechanics (Morin, 2007), which elegantly solves curves with minimum energy requirement using stationary action or with variability as in the many-body approaches discussed above. For finding such a minimal potential field, the function of the task completion time per number of robots has to be obtained beforehand. As far as I know, no work presents how to get this function from the motion equations of a robot, which is a challenge in Swarm Robotics. Thus, experimental approximations are presented in Chapter 6 by observing the swarm, making this work a fundamental first step for studying the common target problem as a local-to-global analysis to serve as a basis for comparing new approaches for a theory capable of finding such minimal potential field.

3.3 Ad Hoc Multi-Robot Systems

The definition of the ad hoc teamwork problem, the similar names and its applications are presented. Different metrics and techniques were studied in the literature on this problem in multi-robot systems and robotic swarms, as shown below.

Definition and Similar Names. Ad hoc teamwork is the problem of coordinating autonomous agents without prior knowledge about the team where they are inserted but have the same goal (Stone et al., 2010). Another name for this problem is impromptu teamwork (Melo and Sardinha, 2016; Bowling and McCracken, 2005). A recent survey presentation about this problem (Mirsky et al., 2022) mentioned a closely related area named "zero-shot coordination", which studies methods for agents increasing rewards when set into a cooperative situation with new teammates (Hu et al., 2020; Bullard et al., 2020). In that survey, the problem is formulated such that the inserted agent learns about the others. Although the literature of learning about other agents is rich – as surveyed by Albrecht and Stone (2018) – no learning is needed in this work.

Applications of Ad Hoc Teamwork in Multi-Agent Systems. Furthermore, ad hoc teamwork multi-agent systems can be used in the distribution of essential goods (Palanca et al., 2020) and games (Cortés and Sheremetov, 2002). In my opinion, the work of scientists in the world with the aim of finding a solution to a problem, such as the cure for a disease, can be seen as a multi-agent system so that each agent – the scientist – contributes to the solution of this problem, but the other agents do not know much about those working in the present moment on the same problem

although may know previous work and learn from it. In this regard, scientific work can be seen as an ad hoc teamwork multi-agent system.

Studies of Ad Hoc Teamwork in Multi-Robot Systems. Additionally, for multi-robot systems, ad hoc teamwork is most applied in robot soccer, although there are works in marine applications and treasure hunting by robots. MacAlpine et al. (2014) made simulations and real-world experiments with three different leagues of RoboCup 2013 for a pick-up game. Teams were formed with agents without prior coordination. They measured average goal difference and average human-judged score. Others metrics for that work were proposed later by MacAlpine and Stone (2017). Bowling and McCracken (2005) apply adaptive and predictive techniques for one agent joining unknown teammates. They tested with simulations the effectiveness of pick-up and baseline players, measuring the probability of a impromptu team scoring the next goal in a game. Barrett and Stone (2015) devised an algorithm that learns policies to work with other agents and can be easily adapted to new ones. They tested in a robot soccer simulator and compared the results by measuring the fraction of the time that the team scores in half field offence task. This algorithm is further analysed by Barrett (2015) and Barrett (2014). Apart from robot soccer, Carlucho et al. (2022) discuss the challenges and developments of ad hoc teamwork in robotic marine operations for intervention, search and rescue, but no experiment was presented. Jones et al. (2006) utilised it for a treasure hunt and experimented with a system of two heterogeneous robots. No metric nor comparison was performed, and only the test of their implementation was reported.

Applications in Robotic Swarms. Similarly, there are a few studies handling ad-hoc teamwork in the context of swarm robotics. Genter, Agmon, and Stone (2013) proposes a heuristic to flock simulated bird agents. They measured the number of steps to achieve the desired orientation and compared their heuristic with other methods without considering the ad hoc agent. This work was later extended by Genter and Stone (2014). Menashe and Agmon (2022) also study ad hoc teamwork in the flocking of a swarm in simulations. They equipped the agent with a signalising mechanism which tells other agents to act in a specified manner but may misunderstand the signal. They also examined this mechanism in swarm dispersion. Grabe et al. (2021) consider ad hoc teamwork in robotic swarms by employing reinforcement learning in a robot that follows a group. They counted the rate of how many times the trained robot and groups achieved their goals per experiment and compared it in different situations. Avrahami and Agmon (2019) introduces the contamination problem in robotic swarms, i.e., a consensus problem where the surrounding, internal state and other robots affect the outcome of avoiding adversarial diversion by other robots. They used real robots and simulations to evaluate their results with experiments to ensure consensus in geometric formations. They compared the number of healthy and contaminated robots, the time of convergence to that

consensus and the rate of convergence versus the resistance level (i.e., the level of immunity to the contamination). As these authors, the solutions presented in Section 5.3 and estimated in Section 6.1.4 aim to contribute to the literature of ad hoc teamwork in robotic swarms.

Chapter 4

Analysis of the theoretical strategies

In this chapter, I present an extensive mathematical study of the common target problem, allowing to understand how to measure the access to a common target using the target area throughput. Assuming a circular target area and that the robots are constantly moving at the maximum linear speed and have a fixed minimum distance from each other, theoretical strategies for entering the area are shown, and their maximum theoretical throughput for a fixed time (also called *instantaneous* throughput) and their maximum asymptotic throughput when time goes to infinity are calculated (or bounds for it).

These quantities are formally defined in Section 4.2 after describing the notation used in this work in Section 4.1. The presented theoretical strategies depend on the size of the target area. Section 4.3 justifies why robots should go in a queue by assuming the target is a point. When the target area is a small circle, Section 4.4 describes how the robots must move to that target area and calculates its instantaneous and asymptotic throughput. From these results, in Section 4.5, the strategies for a large target area are developed by forming a corridor towards the target area or making multiple curved trajectories towards the boundary of the target area. For the corridor strategy, the throughput when the robots are moving towards the target in square and hexagonal packing formations is discussed in Sections 4.5.1 and 4.5.2. The same is done for the curved trajectories strategy in Section 4.5.3. These strategies are theoretically compared in Section 4.5.4. In Section 4.6, the correctness of these calculations is demonstrated by simulations. Artificial potential fields are not considered in this chapter. A summary of the results of this chapter is in Section 4.7.

The presented strategies are the theoretical grounding for new distributed algorithms for robotic swarms described in Chapter 5. When assuming that the robots are constantly moving at maximum linear speed and maintaining a fixed minimum

distance, analytical calculations of the maximum possible throughput for a given time and bounds or the exact value of the maximum asymptotic throughput for the different theoretical strategies are provided. Based solely on these calculations, it is possible to compare which strategy is better. However, for robots using artificial potential fields, it is not straightforward to obtain explicit throughput equations due to the changeability of those quantities previously assumed constant. Then, in the lack of closed asymptotic equations, simulations were performed in Section 4.6 for the algorithms inspired by these strategies in order to obtain experimental throughput and compare algorithms for varying linear speeds and inter-robot distances. As shown by these experimental data, their variation and the effect of the other robots in the trajectory does affect the throughput. However, the analytically calculated maximum throughput in this work serves as an upper bound to the ones obtained from the simulations in more realistic conditions when considering the mean speed and mean distance between the robots in place of the constant values on the obtained equations.

Moreover, this chapter has many proofs whose main objective is to show how to obtain the instantaneous and asymptotic throughput of the presented strategies (or give bounds if no exact value can be obtained), along with lemmas and other propositions for helping these proofs. To avoid impairing its flow, whenever the formal proof is longer than a page, the main ideas of the proofs are given to summarise them. The corresponding formal proofs are shown in Appendix B.

4.1 Notation

In this chapter, geometric notation is used as follows. $\overleftrightarrow{AB}$, $\overrightarrow{AB}$ and $\overline{AB}$ represent a line passing through points A and B, a ray starting at A and passing through B and a segment from A to B, respectively. $|\overline{AB}|$ is the size of $\overline{AB}$. $\overleftrightarrow{AB} \parallel \overleftrightarrow{CD}$ means $\overleftrightarrow{AB}$ is parallel to $\overleftrightarrow{CD}$. If a two-dimensional point is represented by a vector P_1, its x- and y-coordinates are denoted by $P_{1,x}$ and $P_{1,y}$, respectively.

$\triangle ABC$ expresses the triangle formed by the points A, B and C. $\triangle ABC \cong \triangle DEF$ and $\triangle ABC \sim \triangle DEF$ mean the triangles ABC and DEF are congruent (same angles and same size) and similar (same angles), respectively. Depending on the context, the notation is omitted for brevity.

$\widehat{AOB}$ means an angle with vertex O, one ray passing through point A and another through B. Depending on the context, if only one $\triangle EFG$ is being dealt with, its angles will be named only by $\widehat{E}$, $\widehat{F}$ and $\widehat{G}$. All angles are measured in radians in this thesis.

4.2 Definitions and Preliminaries

This chapter considers the scenario where a large number of robots must reach a common target. After reaching the target, each robot moves towards another destination which may or may not be common among the robots. It is assumed that the target is defined by a circular area of radius s. A robot reaches the target if its centre of mass is at a distance below or equal to the radius s from the centre of the target. If a point different from the centre of mass is considered (for instance, the first point of the robot's surface to get in contact with the target), the time to reach the target between the robots is affected by the shape of the robots. Using their centre of mass ignores the shape of the robots by considering them as points because the centre of mass can be computed from any shape. In addition, it is supposed that there is no minimum amount of time to stay at the target. Additionally, the angle and the speed of arrival have no impact on whether the robot reached the target or not. In this chapter, theoretical strategies are constructed to solve that task and show limits for the efficiency of real-life implementations, which is described in Chapter 5. To measure performance, the following definition is presented.

Definiton 1. *The* throughput *is the inverse of the average time between arrivals at the target.*

Informally speaking, the throughput is measured by someone located on the common target (i.e., on its perspective). It is considered that an optimal algorithm minimises the average time between two arrivals or, equivalently, maximises throughput. The unit for throughput can be in s^{-1}. It will be noted f (as in frequency).

Assume an experiment was run with $N \geq 2$ robots for t units of time, such that the time between the arrival of the i-th robot and the $i+1$-th robot is t_i, for i from 1 to $N-1$. Then, by Definition 1,

$$f = \frac{1}{\frac{1}{N-1}\sum_{i=1}^{N-1} t_i} = \frac{N-1}{\sum_{i=1}^{N-1} t_i} = \frac{N-1}{t},$$

because $\sum_{i=1}^{N-1} t_i = t$. Thus, an equivalent definition of throughput is given:

Definiton 2. *The* throughput *is the ratio of the number of robots that arrive at the common target area, not counting the first robot to reach it, to the time taken for the last robot to arrive at the target area after the arrival of the first robot.*

The target area is a limited resource that must be shared between the robots. Since the linear speeds of the robots have an upper bound, a robot needs a minimum amount of time to reach and leave the target before letting another robot in. Let the *asymptotic throughput* of the target area be its throughput as the time tends

to infinity. Because any physical phenomenon is limited by the speed of light, this measure is bounded. Then, the asymptotic throughput is well suited to measure the access of a common target area as the number of robots grows.

One should expect that the asymptotic throughput depends mainly on the target size and shape, the speed of the robots, and the distance between robots. As any bounded target region can be included in a circle of radius s, only circular target regions will be dealt with hereafter. If the robots are moving at maximum speed and keeping the distance between each at a minimum value all the time, then it is also expected that the throughput and asymptotic throughput reach their maximum value. Thus, it is assumed hereafter that the robots move at a constant maximum linear speed, v, and the distance between each other is either constant when possible or no lower than a fixed value, d.

To efficiently access the target area, two main cases are identified: $s \geq d/2$ and $s < d/2$. There are targets that several robots can simultaneously reach without collisions. That is the case if the radius $s \geq d/2$. Thus, one approach is making lanes arrive in the target region so that as many robots as possible can simultaneously arrive. After the robots arrive at the target, they must leave the target region by making curves. However, this approach does not obtain good results in realistic simulations due to the influence of other robots, although it is theoretically the best approach if the robots could run at a constant speed and maintain a fixed minimum distance between each other (see Section 5.4 for more details).

The case where $s < d/2$, when only one robot can occupy the target area simultaneously, is of interest. Making two queues and avoiding the inter-robot distance being less than d is good guidance to work efficiently. Particularly, the case $s = 0$ offers interesting insights, so this is discussed next.

4.3 Common Target Point: $s = 0$

Consider the case where robots are moving in straight lines at constant linear speed v, maintaining a distance of at least d between each other. A robot has reached the target when its centre of mass is over the target. When $s = 0$, the target is a point. The first result is the optimal throughput when robots are moving in a straight line to a target point. It is illustrated in Figure 4.1. This section constructs a solution to attain the optimal throughput.

First, consider two robots, Robot 1 and Robot 2. Their trajectories are straight lines towards the target. Assume the straight-line trajectory of Robot 1 has an angle δ_1 with the x-axis and the one of Robot 2 has δ_2. Define $\delta_2 - \delta_1 = \delta$ as the angle between the two lines. The positions of the robots are described by the kinematic Equation (4.1) below, where $(x_1(t), y_1(t))$ and $(x_2(t), y_2(t))$ are the positions of Robot 1 and Robot 2, respectively, and $t \in \mathbb{R}$ is an instant of time. Without loss of generality,

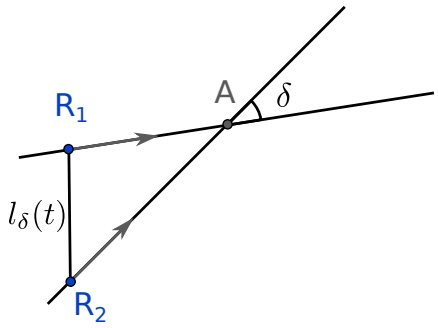


Figure 4.1: Two robots, R_1 and R_2, are moving in straight lines toward a target at A. The angle between their trajectory is δ. The distance between the two robots over time is denoted by $l_\delta(t)$.

the origin of time is set when Robot 1 reaches the target, and the target is located at $(0, 0)$. Thus, $(x_1(0), y_1(0)) = (0, 0)$. τ is the delay between the two arrivals at the target. Then, $(x_2(\tau), y_2(\tau)) = (0, 0)$, and

$$\begin{bmatrix} x_1(t) \\ y_1(t) \end{bmatrix} = \begin{bmatrix} vt\cos(\delta_1) \\ vt\sin(\delta_1) \end{bmatrix} \text{ and } \begin{bmatrix} x_2(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} v(t-\tau)\cos(\delta_2) \\ v(t-\tau)\sin(\delta_2) \end{bmatrix} \tag{4.1}$$

In order to find the optimal throughput, this chapter starts with its first lemma:

Lemma 1. *To maintain a distance of at least d between the two robots, the minimum delay between their arrival is $\frac{d}{v}\sqrt{\frac{2}{1+\cos(\delta)}}$.*

Idea of the proof. A function of the distance of two robots over the time is defined and constrained to be always greater or equal to d. The angle δ is a parameter of that function, and the aim of this proof is to provide the minimum distance between the robot as a function of d and δ. Two cases are calculated: (i) $\delta \neq 1$ and (ii) $\delta = 1$. Both cases yield the same result. The formal proof is in Section B.1. □

This result leads to Proposition 1.

Proposition 1. *The maximum throughput f for a point-like target ($s = 0$) is $f = \frac{v}{d}$. It is achieved when robots form a single line, i.e., the angle between the trajectories of the robots must be* 0.

Proof. It is shown by induction on N, which is the number of robots moving towards the target. Define δ_N as the angle between the trajectories of Robot $N-1$ and Robot N; τ_N, the minimum delay between the arrival of Robot $N-1$ and Robot N; and Δ_N, the minimum delay between the arrival of Robot 1 and Robot N. The aim is to show the following predicate: for all $N \geq 2$, $\Delta_N = (N-1)d/v$ for $\delta_2 = \delta_3 = \ldots = \delta_N = 0$.

Base case ($N = 2$): Let τ_2 be the delay between the arrival of Robot 1 and Robot 2. From Lemma 1, the minimum delay between Robot 1 and Robot 2 is equal to $\frac{d}{v}\sqrt{\frac{2}{1+\cos(\delta_2)}}$, which is minimised by $\delta_2 = 0$. Then, the minimum delay between the two robots is $\tau_2 = d/v = \Delta_2$.

Inductive step: suppose the predicate is true for a given $N - 1 \geq 2$. It will be shown that this implies the predicate is true for N robots. As in the previous case, it is concluded from Lemma 1 that the minimum delay between Robot $N - 1$ and Robot N is equal to $\frac{d}{v}\sqrt{\frac{2}{1+\cos(\delta_N)}}$, which is minimised by $\delta_N = 0$. Then, the minimum delay between the two robots is $\tau_N = d/v$, thus

$$\Delta_N = \Delta_{N-1} + \tau_N = (N-2)\frac{d}{v} + \frac{d}{v} = (N-1)\frac{d}{v}.$$

Consequently, the minimum delay between Robot 1 and Robot N is $\Delta_N = \sum_{i=2}^{N} \tau_i = (N-1)\frac{d}{v}$ and the time of arrival of Robot N, for all N, is minimised by $\delta_2 = \delta_3 = \ldots = \delta_N = 0$. Finally, by Definition 2, the throughput is $f = \frac{N-1}{\Delta_N} = \frac{v}{d}$. □

The insight derived from Proposition 1 implies that one should increase the speed of the robots or decrease the minimum distance between them to increase the throughput. It is also noted that the optimal trajectory for all the robots is to form a queue behind the target and Robot 1. As a result, the optimal path is to create one lane to reach the target. When the angle δ between the path of a robot and the next one is increased, a delay from the optimal throughput is introduced. For instance, Figure 4.2 shows the normalised delay for different angles δ (normalised by dividing τ by $\tau_{min} = d/v$) between two robots, according to Lemma 1. This figure shows that for an angle of $\pi/3$, the minimum delay is 15% higher than for an angle of 0, and the minimum delay is 41% higher for an angle of $\pi/2$.

4.4 Small Target Area: $0 < s < d/2$

This section supposes a small target area where $0 < s < d/2$; hence, two lanes with a distance d cannot fit towards the target yet. The next results are based on a strategy using two *parallel lanes* as close as possible to guarantee the minimum distance d between robots. Figure 4.3 describes these two parallel lanes. This strategy is called *compact lanes* hereafter. Proposition 2 considers a target area with radius $0 < s \leq \frac{\sqrt{3}}{4}d$, and Proposition 3 assumes $\frac{\sqrt{3}}{4}d < s < \frac{d}{2}$.

Proposition 2. *Assume two parallel lanes with robots at constant speed v and maintaining a constant distance d between them. The throughput of a common target area with radius $0 < s \leq \frac{\sqrt{3}}{4}d$ at a given time t after the first robot has reached the target area is*

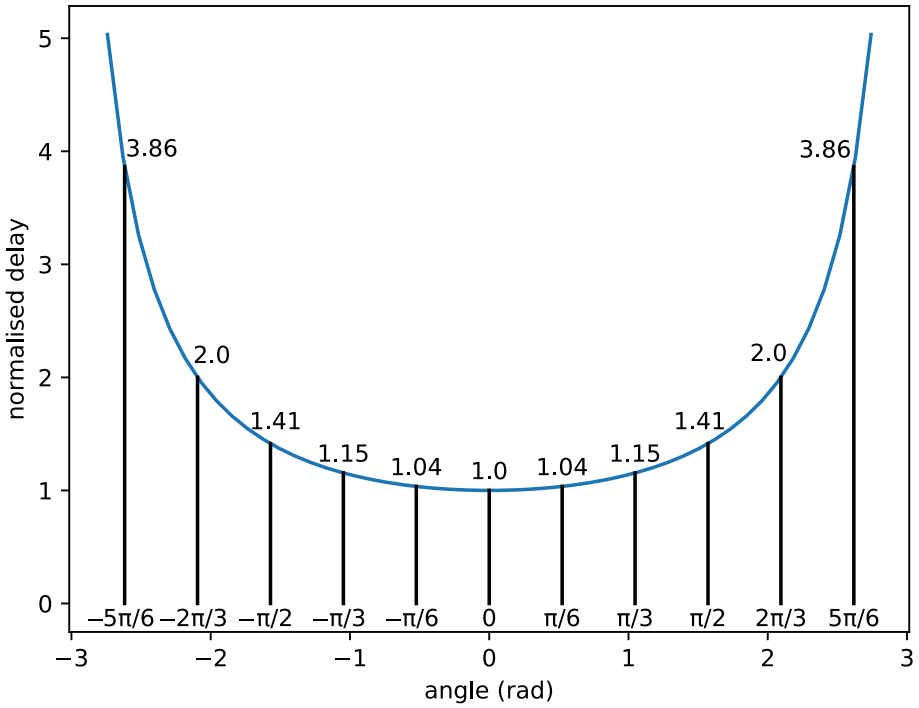


Figure 4.2: Normalised delay versus the angle between the trajectories of the robots.

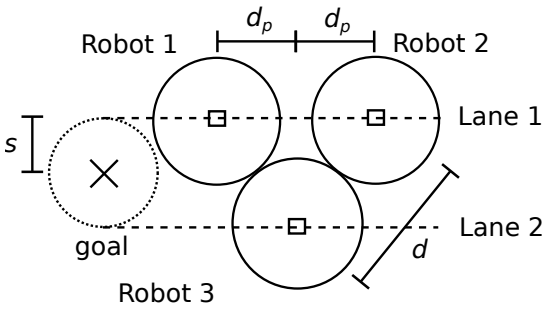


Figure 4.3: Two parallel robot lanes for a small target, illustrating the compact lanes strategy.

$$f(t) = \frac{1}{t}\left(\left\lfloor \frac{vt}{2\sqrt{d^2-(2s)^2}} \right\rfloor + \left\lfloor \frac{vt}{2\sqrt{d^2-(2s)^2}} + \frac{1}{2} \right\rfloor\right) \tag{4.2}$$

and is limited by

$$f = \lim_{t\to\infty} f(t) = \frac{v}{d\sqrt{1-(\frac{2s}{d})^2}}. \tag{4.3}$$

Proof. Consider Figure 4.3. The distance between the lanes is $2s$, and the distance between two robots is d. Thus, $d_p = \sqrt{d^2-(2s)^2}$. Hence, the distance between two robots in the same lane is $d_e = 2d_p = 2\sqrt{d^2-(2s)^2}$.

The distance between two robots in the same lane must not be less than d, so $d_e \geq d$. This is true, because, as $0 < s \leq \frac{\sqrt{3}}{4}d$,

$$d_e = 2\sqrt{d^2-(2s)^2} \geq 2\sqrt{d^2-\left(2\frac{\sqrt{3}}{4}d\right)^2} = 2\sqrt{d^2-\left(\frac{\sqrt{3}}{2}d\right)^2} = 2\sqrt{d^2-\frac{3}{4}d^2}$$
$$= 2\sqrt{\frac{1}{4}d^2} = d.$$

Without loss of generality, assume the first robot to reach the target area being at the top lane in Figure 4.3. The number of robots on any lane is the integer division of the size of the lane by the offset between the robots plus one (because the first robot is included in this counting). Therefore, the number of robots for a given time t in the top lane is $N_1(t) = \left\lfloor \frac{vt}{d_e} + 1 \right\rfloor$ and in the bottom lane is $N_2(t) = \left\lfloor \frac{vt-d_p}{d_e} + 1 \right\rfloor = \left\lfloor \frac{vt}{d_e} + \frac{1}{2} \right\rfloor$. By Definition 2,

$$f(t) = \frac{N_1(t)+N_2(t)-1}{t} = \frac{1}{t}\left(\left\lfloor \frac{vt}{2\sqrt{d^2-(2s)^2}} \right\rfloor + \left\lfloor \frac{vt}{2\sqrt{d^2-(2s)^2}} + \frac{1}{2} \right\rfloor\right).$$

By the definition of the floor function, $\lfloor x \rfloor = x - frac(x)$ with $0 \leq frac(x) < 1$. Thus,

$$\begin{aligned}
\lim_{t\to\infty} f(t) &= \lim_{t\to\infty} \frac{1}{t}\left(\left\lfloor \frac{vt}{d_e}\right\rfloor + \left\lfloor \frac{vt}{d_e}+\frac{1}{2}\right\rfloor\right) \\
&= \lim_{t\to\infty} \frac{1}{t}\left(\frac{vt}{d_e} - frac\left(\frac{vt}{d_e}\right) + \frac{vt}{d_e} + \frac{1}{2} - frac\left(\frac{vt}{d_e}+\frac{1}{2}\right)\right) \\
&= \lim_{t\to\infty} \frac{1}{t}\left(\frac{2vt}{d_e}\right) + \frac{1}{t}\left(-frac\left(\frac{vt}{d_e}\right) + \frac{1}{2} - frac\left(\frac{vt}{d_e}+\frac{1}{2}\right)\right) \\
&= \frac{2v}{d_e} = \frac{2v}{2\sqrt{d^2-(2s)^2}} = \frac{v}{d\sqrt{1-(2s/d)^2}},
\end{aligned}$$

as $\lim_{t\to\infty} \dfrac{frac(x)}{t} = 0$, for any x. □

Proposition 3. *Assume two parallel lanes with robots at constant speed v and maintaining a constant distance d between them. The throughput of a common target area with radius $\frac{\sqrt{3}}{4}d < s < \frac{d}{2}$ at a given time t after the first robot has reached the target area is*

$$f(t) = \frac{1}{t}\left(\left\lfloor \frac{vt}{d}\right\rfloor + \left\lfloor \frac{vt}{d}+\frac{1}{2}\right\rfloor\right) \tag{4.4}$$

and is limited by

$$f = \lim_{t\to\infty} f(t) = \frac{2v}{d}. \tag{4.5}$$

Proof. As the distance between the robots must be at least d and $\frac{\sqrt{3}}{4}d < s < \frac{d}{2}$, $d_p = d/2$ is assigned in Figure 4.3. By doing so, two robots side by side in one lane and a robot in the other lane form an equilateral triangle with a side measuring d, whose height has size $\frac{\sqrt{3}}{2}d$. Hence, the minimum diameter of the circular target region must be this value, and the hypothesis says so.

Moreover, the radius of the target area is less than $d/2$, implying that the three robots in Figure 4.3 must stay in the equilateral triangle formation because the two lanes cannot be far by d units of distance.

Thus, the throughput for a given time t is calculated similarly as in Proposition 2, resulting

$$f(t) = \frac{1}{t}\left(\left\lfloor \frac{vt}{d}\right\rfloor + \left\lfloor \frac{vt}{d}+\frac{1}{2}\right\rfloor\right) \text{ and } f = \lim_{t\to\infty} f(t) = \frac{2v}{d}.$$

□

Observe that if $t = k\frac{d}{v}$ for any $0 < k \in \mathbb{Z}$ is used in (4.4), the compact lanes strategy can achieve the throughput of two parallel lanes of robots going in the direction of the target region when $t = k\frac{d}{v}$ for any $k \in \mathbb{Z}$ or when $t \to \infty$, even though two robots cannot reach the target region at the same time.

4.5 Large Target Area: $s \geq d/2$

This section focuses on situations where more than two robots can simultaneously touch the target. Three feasible strategies are presented.

The simplest strategy is to consider several parallel lanes being at a distance d from each other. However, it is possible to obtain higher throughput. In particular, two other strategies are identified: (a) using parallel straight line lanes that may be distanced lower than d and (b) robots moving towards the target following curved trajectories. Strategy (a) uses more than two compact lanes, extending the strategy presented in the previous section. By doing this, the robots fit in a hexagonal packing arrangement moving toward the target region. Strategy (b) uses a touch and run approach. In it, robots do not cross the target area, they only reach it and return in the opposite direction using curved trajectories which respect the minimum distance d.

The next section starts with the parallel lanes strategy, which has the lowest asymptotic throughput over the strategies presented in this section, for comparison with the other strategies. In particular, it will be used later as a justification for the lowest number of lanes used in the strategy (b) in (4.14) in Proposition 7. Following their description and properties, a discussion comparing them is provided.

4.5.1 Parallel Lanes

It is considered here that the robots are moving inside lanes. The lanes are straight lines, and the linear speed v of the robots is constant. The lanes are separated by a distance d, and each robot maintains a distance d from each other. Figure 4.4 illustrates an example of this strategy. The first lane, Lane 1, is at the top. The first robot of each lane is located at $(s, s-(i-1)d)$ for the Lane i. The next proposition states the throughput for a given time and the asymptotic throughput for this strategy.

Proposition 4. *Assume a circular target region with its centre at $(0,0)$ and radius $s \geq \frac{d}{2}$ and parallel lanes starting at $(s, s-(i-1)d)$ for $i \in \{1, \ldots, \lfloor \frac{2s}{d} \rfloor + 1\}$. At each Lane i, the first robot is located at the point $(s, s-(i-1)d)$ in the starting configuration. Then, the first robot to reach the target is located at $(s, s-(J-1)d)$, for*

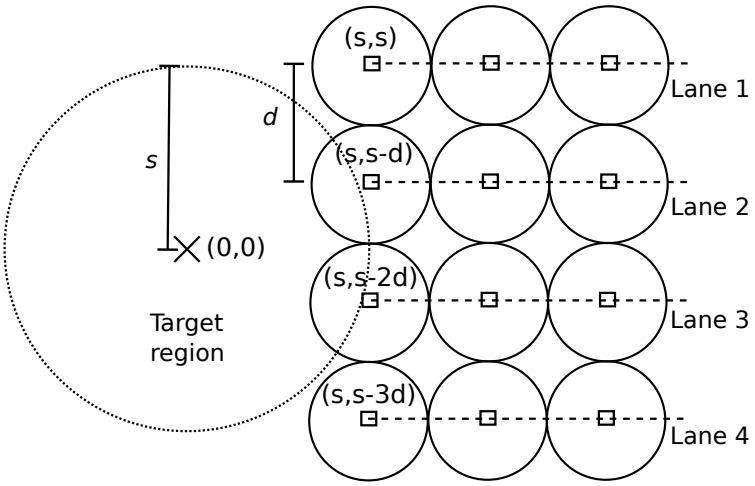


Figure 4.4: Example of the parallel lanes strategy.

$$J = \begin{cases} \left\lfloor \frac{s}{d} \right\rfloor + 1, & \textit{if } \left| s - \left\lfloor \frac{s}{d} \right\rfloor d \right| \leq \left| s - \left\lceil \frac{s}{d} \right\rceil d \right|, \\ \left\lceil \frac{s}{d} \right\rceil + 1, & \textit{otherwise.} \end{cases}$$

The throughput for a given time t after the first robot reaches the target region is:

$$f_p(t) = \frac{1}{t} \left(\sum_{i=1}^{\left\lfloor \frac{2s}{d} \right\rfloor + 1} N_i(t) \right) - \frac{1}{t}, \tag{4.6}$$

for

$$N_i(t) = \begin{cases} \left\lfloor \frac{vt - d_i + d_J}{d} + 1 \right\rfloor, & \textit{if } t \geq \frac{d_i - d_J}{v}, \\ 0, & \textit{otherwise,} \end{cases}$$

$$d_i = s - \sqrt{s^2 - (s - (i-1)d)^2}, \textit{ and}$$

$$f_p = \lim_{t \to \infty} f_p(t) = \left\lfloor \frac{2s}{d} + 1 \right\rfloor \frac{v}{d}. \tag{4.7}$$

Idea of the proof. The throughput is calculated regarding, for each parallel lane, the position of the circular target area where the robot arrives. As the first robot reaching it in each lane takes a different time, the total number of robots that arrived at the target area at a given time considers this difference. The formal proof is in Section B.2. □

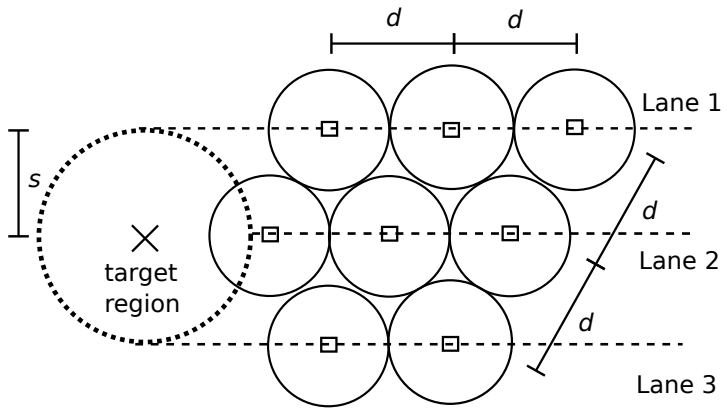


Figure 4.5: Robot lanes for hexagonal packing.

4.5.2 Hexagonal Packing

By extending the compact lanes to more than two lanes, the robots will be packed in a hexagonal formation. An illustration of this strategy is shown in Figure 4.5. As it can be seen, robots from different lanes are still able to move towards the target keeping a distance d from each other, even though the lanes have a distance lower than d.

An upper bound of the asymptotic throughput for the *hexagonal packing* strategy is first computed, then the throughput for a given time using this strategy is calculated.

Proposition 5. *Assume robots moving at speed v, going to a circular target of radius s. The upper bound of the asymptotic throughput for the hexagonal packing strategy is*

$$f_h^{max} = \frac{2}{\sqrt{3}} \left(\frac{2s}{d} + 1 \right) \frac{v}{d}. \tag{4.8}$$

Idea of the proof. The upper bound of the throughput of the hexagon packing relies on the results of the circle packing problem. By the optimal surface occupied by the circles in a rectangle, the number of robots in hexagonal packing that reach the target area through a rectangular corridor at a given time is calculated. When taking the limit as time goes to infinity, the upper bound of the throughput is obtained. The formal proof is in Section B.3. □

Proposition 5 presents an upper bound of the asymptotic throughput using hexagonal packing, but it does not tell us which is the best placement of the robots

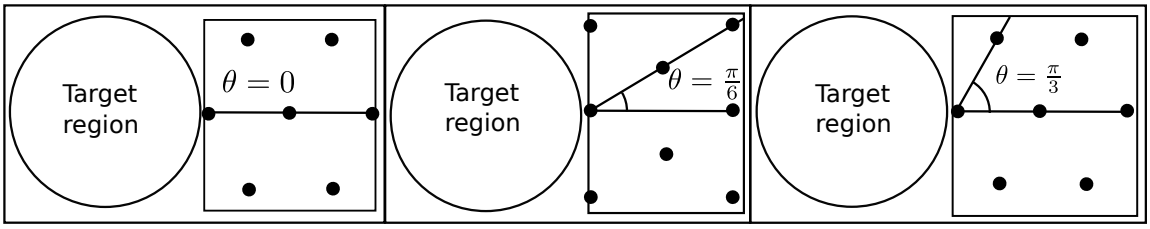


Figure 4.6: Example of hexagonal packing with different angles. The robots are the black dots.

inside a corridor since the hexagonal formation can be rotated by different angles. Hence, the results about the throughput considering the placement of the hexagonal packing inside a corridor of robots going to the target region will be presented. First, however, the following definition will be needed.

Definiton 3. *The hexagonal packing angle θ is the angle formed by the x-axis and the line formed by any robot at position (x, y) and its neighbour at $(x+d\cos(\theta), y+d\sin(\theta))$ under the target region reference frame.*

Observe that any robot at (x, y) under the hexagonal packing has at most six neighbours located at $\left(x+d\cos\left(\theta\right), y+d\sin\left(\theta\right)\right)$, $\left(x+d\cos\left(\theta+\frac{\pi}{3}\right), y+d\sin\left(\theta+\frac{\pi}{3}\right)\right)$, $\ldots$, $\left(x+d\cos\left(\theta+\frac{5\pi}{3}\right), y+d\sin\left(\theta+\frac{5\pi}{3}\right)\right)$ (Figure 4.6). If $\theta = \frac{\pi}{3}$, putting this value in the previous series results in the first neighbour robot being at $(x+d\cos(\pi/3), y+d\sin(\pi/3))$ and the last neighbour robot at $(x+d\cos(0), y+d\sin(0))$. This is the same result if $\theta = 0$ was used. Consequently, due to this periodicity, hexagonal packing angles in $[0, \frac{\pi}{3})$ are assumed.

The next proposition states the bounds of the throughput in the limit towards the infinity for hexagonal packing using an arbitrary, but fixed, hexagonal packing angle θ. A fixed θ is assumed because normally in a robotic swarm the robots rely on local sensing. In order to obtain the maximum number of robots inside the corridor, all robots should know the size of the corridor and communicate by local-ranged message sending. It would take time to send information, and for all robots to adjust their orientation each time a new robot joins the swarm when using this local sensing approach.

In other words, if the corridor where the robots are going in the direction of the target is increasing over time, then θ should change over time for the optimal throughput. However, in practice, changing the hexagonal packing angle implies all robots must turn to a hexagonal packing angle θ^* depending on the size of the new rectangle based on the added robots to it to maximise the number of robots inside the corridor. In addition to the time to send messages with this parameter, more time would be needed for every robot to adapt to the updated computed θ^* because

the turning speed of the robots is finite. Therefore, this thesis does not handle this adjustable scenario.

Proposition 6. *Assume the robots using hexagonal formation coming to a circular target area with radius s such that the first robot to reach it was at time 0 at $(x_0, y_0) = (w, 0)$, for any $w \geq s$. For a given time t, the robots are going to the target at linear speed v, keeping a distance d between neighbours ($0 < d \leq 2s$), using fixed hexagonal packing angle $\theta \in [0, \pi/3)$. The throughput for a given time is given by*

$$\begin{aligned} f_h(t,\theta) =& \frac{1}{t} \sum_{x_h=-n_l^-}^{n_l^+-1} \left(\lfloor Y_2^R(x_h) \rfloor - \lceil Y_1^R(x_h) \rceil + 1 \right) + \\ & \frac{1}{t} \sum_{x_h=B}^{U} \left(\lfloor Y_2^S(x_h) \rfloor - \lceil Y_1^S(x_h) \rceil + 1 \right) - \frac{1}{t}, \end{aligned} \tag{4.9}$$

for $\left\lfloor Y_2^R(x_h) \right\rfloor \geq \left\lceil Y_1^R(x_h) \right\rceil$ and $\left\lfloor Y_2^S(x_h) \right\rfloor \geq \left\lceil Y_1^S(x_h) \right\rceil$ (if for some x_h, either of these conditions are false, it is assumed that the respective summand for this x_h is zero). Additionally,

$$n_l^- = \left\lfloor \frac{2s \sin\left(|\pi/6 - \theta|\right)}{\sqrt{3}d} \right\rfloor,$$

$$n_l^+ = \left\lfloor \frac{2(vt - s)\cos(\pi/6 - \theta) + 2s \sin(|\pi/6 - \theta|)}{\sqrt{3}d} + 1 \right\rfloor,$$

$$Y_1^R(x_h) = \begin{cases} \max\left(\dfrac{\sin(\frac{\pi}{3} - \theta)x_h - \frac{s}{d}}{\cos\left(\theta - \frac{\pi}{6}\right)}, \dfrac{-\cos(\frac{\pi}{3} - \theta)x_h}{\sin\left(\frac{\pi}{6} - \theta\right)} \right), & \text{if } \theta < \pi/6, \\ \max\left(\dfrac{\sin(\frac{\pi}{3} - \theta)x_h - \frac{s}{d}}{\cos\left(\theta - \frac{\pi}{6}\right)}, \dfrac{\frac{vt-s}{d} - \cos(\frac{\pi}{3} - \theta)x_h}{\sin\left(\frac{\pi}{6} - \theta\right)} \right), & \text{if } \theta > \pi/6, \\ \dfrac{x_h}{2} - \dfrac{s}{d}, & \text{if } \theta = \pi/6, \end{cases}$$

$$Y_2^R(x_h) = \begin{cases} \min\left(\dfrac{\sin(\frac{\pi}{3} - \theta)x_h + \frac{s}{d}}{\cos\left(\theta - \frac{\pi}{6}\right)}, \dfrac{\frac{vt-s}{d} - \cos(\frac{\pi}{3} - \theta)x_h}{\sin\left(\frac{\pi}{6} - \theta\right)} \right), & \text{if } \theta < \pi/6, \\ \min\left(\dfrac{\sin(\frac{\pi}{3} - \theta)x_h + \frac{s}{d}}{\cos\left(\theta - \frac{\pi}{6}\right)}, \dfrac{-\cos(\frac{\pi}{3} - \theta)x_h}{\sin\left(\frac{\pi}{6} - \theta\right)} \right), & \text{if } \theta > \pi/6, \\ \dfrac{x_h}{2} + \dfrac{s}{d}, & \text{if } \theta = \pi/6, \end{cases}$$

$$B = \begin{cases} \left\lceil \dfrac{2(\sin(\pi/3 - \theta)(c_x - l_x) + \cos(\pi/3 - \theta)(y_0 - l_y - s))}{\sqrt{3}d} \right\rceil, & \text{if } t > \dfrac{s}{v}, \\ \left\lceil -\dfrac{2\sqrt{2svt - (vt)^2}}{\sqrt{3}d} \sin\left(\theta + \dfrac{\pi}{6}\right) \right\rceil, & \text{otherwise,} \end{cases}$$

for $c_x = x_0 + vt - s$ *and*

$$(l_x, l_y) = \begin{cases} \underset{(x,y)\in Z}{\operatorname{argmin}} |vt - s + x_0 - x| + |y_0 - y|, & \text{if } t > \frac{s}{v}, \\ (x_0, y_0), & \text{otherwise,} \end{cases}$$

where Z *is the set of robot positions inside the rectangle measuring* $(vt - s) \times 2s$ *for* $vt - s > 0$. *If* $t > \frac{s}{v}$ *or* $\arctan\left(\frac{\frac{s}{2} - \sin(\theta)(vt-s)}{\frac{\sqrt{3}s}{2} + \cos(\theta)(vt-s)}\right) < \frac{\pi}{2} - \theta$,

$$U = \left\lfloor \frac{2(\sin(\pi/3 - \theta)(c_x - l_x) + \cos(\pi/3 - \theta)(y_0 - l_y) + s)}{\sqrt{3}d} \right\rfloor,$$

otherwise,

$$U = \left\lfloor \frac{2\sqrt{2svt - (vt)^2}}{\sqrt{3}d} \cos\left(\theta - \frac{\pi}{3}\right) \right\rfloor.$$

In addition,

$$Y_1^S(x_h) = \frac{dx_h - C_{-\theta,x} + \sqrt{3}C_{-\theta,y} - \sqrt{\Delta(x_h)}}{2d} \text{ and}$$

$$Y_2^S(x_h) = \begin{cases} \min(L(x_h), C_2(x_h)) - 1, & \text{if } \min(L(x_h), C_2(x_h)) \\ & \quad = \lfloor L(x_h) \rfloor \text{ and } t > \frac{s}{v}, \\ \min(L(x_h), C_2(x_h)), & \text{otherwise,} \end{cases} \tag{4.10}$$

$$C_{-\theta} = \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{bmatrix} \begin{bmatrix} c_x - l_x \\ y_0 - l_y \end{bmatrix},$$

$$\Delta(x_h) = 4s^2 - \left(\sqrt{3}(dx_h - C_{-\theta,x}) - C_{-\theta,y}\right)^2,$$

$$C_2(x_h) = \frac{dx_h - C_{-\theta,x} + \sqrt{3}C_{-\theta,y} + \sqrt{\Delta(x_h)}}{2d},$$

$$L(x_h) = \begin{cases} \dfrac{\sin\left(\frac{\pi}{2} - \theta\right)(dx_h - C_{-\theta,x}) + \cos\left(\frac{\pi}{2} - \theta\right) C_{-\theta,y}}{d \sin\left(\frac{5\pi}{6} - \theta\right)}, & \text{if } t > \dfrac{s}{v}, \\ \dfrac{\sin\left(\frac{\pi}{2} - \theta\right) x_h}{\sin\left(\frac{5\pi}{6} - \theta\right)}, & \text{otherwise,} \end{cases}$$

and

$$f_h(\theta) = \lim_{t\to\infty} f_h(t,\theta) \in \left(\frac{4vs}{\sqrt{3}d^2} - \frac{2v\cos(\theta - \pi/6)}{\sqrt{3}d}, \frac{4vs}{\sqrt{3}d^2} + \frac{2v\cos(\theta - \pi/6)}{\sqrt{3}d} \right]. \tag{4.11}$$

Idea of the proof. The number of robots that reach the target area in hexagonal packing at a given time is divided into two groups: the robots inside a rectangle and those inside a semicircle of radius s. The coordinate space (x_h, y_h) is defined to help the calculation. In this space, integer coordinates correspond to robot positions in hexagonal packing. The number of parallel lines intersecting an integer x_h-coordinate inside a given rectangle is calculated. Then, the total number of robots for a given time inside a rectangle is obtained for each parallel line. In order to count the number of robots inside a rectangle in hexagonal packing, the minimum and maximum integer y_h-coordinate of robots inside the given rectangle for each x_h are obtained: $Y_1^R(x_h)$ and $Y_2^R(x_h)$. The set of integer values on the x_h-axis containing robots inside a given rectangle is divided into three groups that define three different equations for obtaining $Y_1^R(x_h)$ and $Y_2^R(x_h)$. For calculating the number of robots in hexagonal packing inside a semicircle, a similar coordinate system transformation is employed but with a different origin and inclination than the previous one.

After obtaining the expressions for the number of robots for a given time, the limit when the time goes to infinity is obtained. This limit is proved to be zero for the equations that count the number of robots inside areas that are bounded regardless of the time and hexagonal packing angle parameters. For the unbounded area that extends infinitely as time grows, this limit is proven to exist. However, no exact limit could be calculated due to the equations being formed by trigonometric functions combined with floor and ceiling functions, complicating the acquisition of the exact outcome for a given time and hexagonal packing angle. Thus, only the lower and upper bounds of the limit are calculated. The formal proof is in Section B.4. □

The upper and lower bounds presented on (4.11) are below or equal the maximum asymptotic throughput presented by the Proposition 5, Equation (4.8). The result of the Proposition 5 only concerns the maximum asymptotic throughput and does not consider the hexagonal packing angle θ, while Proposition 6 gives a lower bound and tightens the bounds for a given θ. Figure 4.7 presents an example comparison of these equations for two different values of s. As expected, the maximum asymptotic throughput under the optimal density assumption (in (4.8)) is a possible value of the throughput using hexagonal packing and is above or equal to the interval in (4.11) for any given θ. However, for practical robotic swarms applications, a certain hexagonal packing angle must be fixed depending on the expected height of the corridor, target

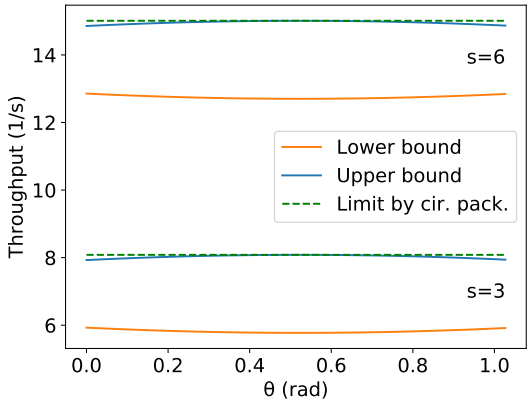


Figure 4.7: Limit given by (4.8) using the circle packing results and the lower and upper bounds of the hexagonal packing limit by (4.11) for $\theta \in [0, \pi/3)$, $d = 1$ m, $v = 1$ m/s and $s \in \{3, 6\}$ m.

size and the minimum distance between the robots, resulting in a throughput below or equal to the upper value presented in Proposition 5.

On the other hand, due to the discontinuities of (4.9), it is difficult to obtain an exact θ that maximises the throughput given the other parameters. In addition, there is no specific value of θ that achieves the maximum throughput for all possible values of the other parameters. Interestingly, given a fixed sub-interval of θ, depending on the number of sample values, new local maxima and minima can arise from these discontinuities. Additionally, a different parity of the number of samples can produce a global maximum in even or odd interval points. To illustrate this, Figures 4.8–4.11 present the result of this equation for some randomly generated parameters and a different number of samples of θ equally spaced and taken from the domain interval, that is, from 0 to $\pi/3$, including these values. Two different orders of magnitude are chosen for the number of equally spaced points in each plot (a small one, about two orders, and a large one of seven orders), and different parities are also given (99 and 100 for the small order, and 10^7 and $10^7 + 1$ for the large one). Although the graphs present an apparent symmetry about the angle $\pi/6$, the values for an angle θ and $\pi/3 - \theta$ may be slightly different, often one robot per total time. Thus, the graphs are not symmetric about that angle in general. As an example, for $t = 43, s = 3, d = 1, v = 1, \theta = \frac{\pi}{10000}$, the number of robots which arrived at the target region is $N(t, \theta) = 296$, but $N(t, \frac{\pi}{3} - \theta) = 295$.

In Figures 4.8–4.11, θ is over the x-axis, and the number of robots inside the given rectangle is over the y-axis. These plots use $v = 1$ m/s. The maximum value in each image is represented by an orange circle, and a rectangle represents the maximum between the left and the right image. No square means the maximum values in both sides are equal. Each one of the Figures 4.8–4.11 presents two different sets of

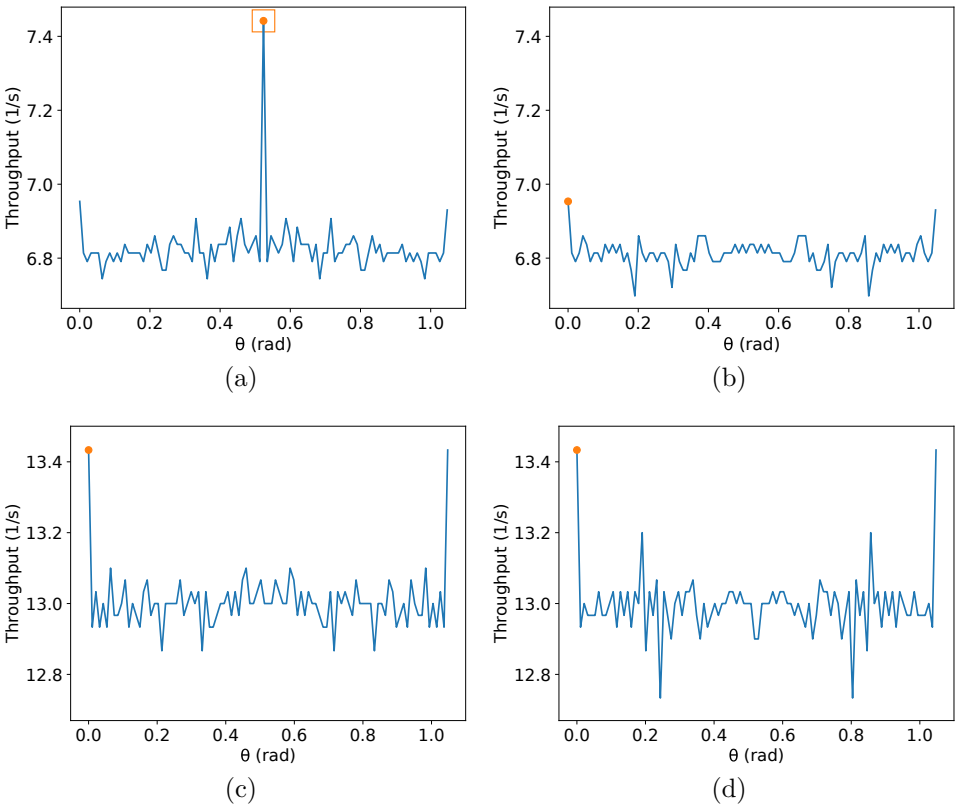


Figure 4.8: Examples of (4.9) varying θ from 0 to $\frac{\pi}{3}$ for different and randomly generated values of t, s, and d. It continues in Figure 4.9. (a) For 99 samples, $t = 43$ s, $s = 3$ m, $d = 1$ m. (b) For 100 samples, $t = 43$ s, $s = 3$ m, $d = 1$ m. (c) For 99 samples, $t = 30$ s, $s = 2.5$ m and $d = 0.66$ m. (d) For 100 samples, $t = 30$ s, $s = 2.5$ m and $d = 0.66$ m.

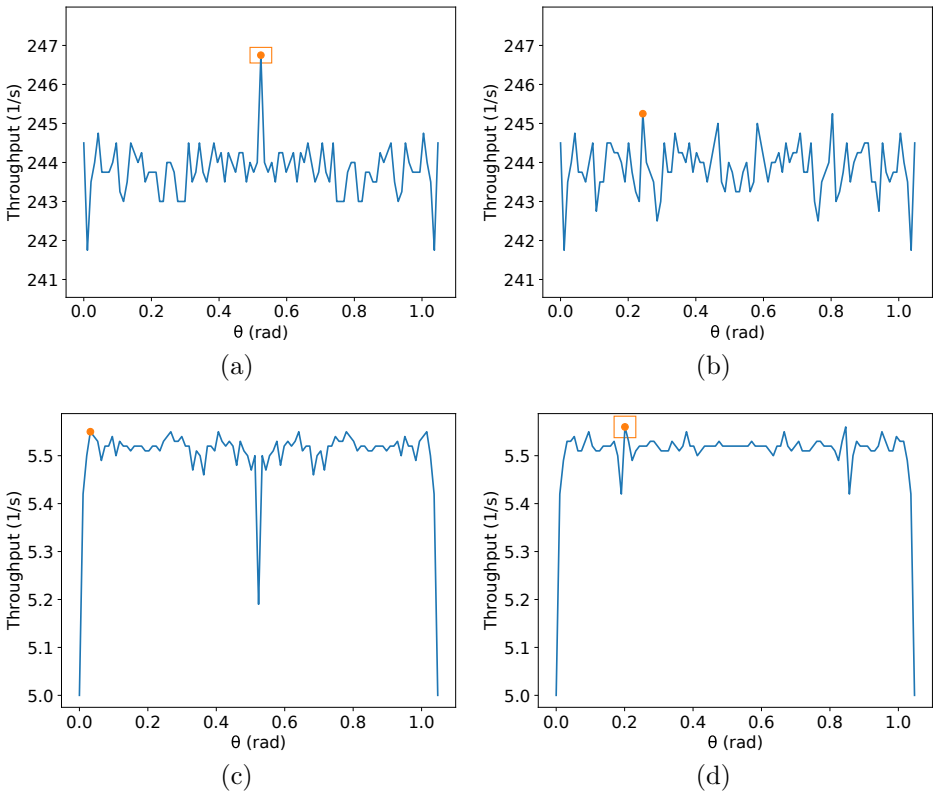


Figure 4.9: Continuation of Figure 4.8: examples of (4.9) varying θ from 0 to $\frac{\pi}{3}$ for different and randomly generated values of t, s and d. (a) For 99 samples, $t = 4$ s, $s = 2$ m and $d = 0.13$ m. (b) For 100 samples, $t = 4$ s, $s = 2$ m and $d = 0.13$ m. (c) For 99 samples, $t = 100$ s, $s = 2.40513$ m and $d = 1$ m. (d) For 100 samples, $t = 100$ s, $s = 2.40513$ m and $d = 1$ m.

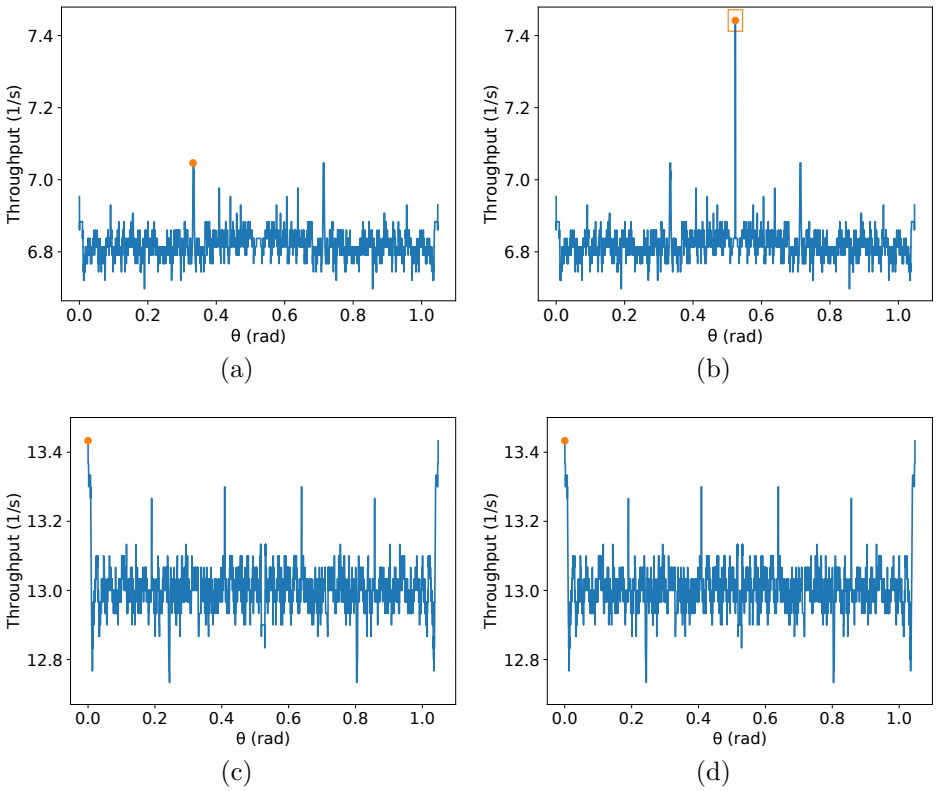


Figure 4.10: Similar to Figures 4.8 and 4.9 but using 10^7 and $10^7 + 1$ equally spaced points for $\theta \in [0, \pi/3)$. It continues in Figure 4.11. (a) For 10^7 samples, $t = 43$ s, $s = 3$ m, $d = 1$ m. (b) For $10^7 + 1$ samples, $t = 43$ s, $s = 3$ m, $d = 1$ m. (c) For 10^7 samples, $t = 30$ s, $s = 2.5$ m and $d = 0.66$ m. (d) For $10^7 + 1$ samples, $t = 30$ s, $s = 2.5$ m and $d = 0.66$ m.

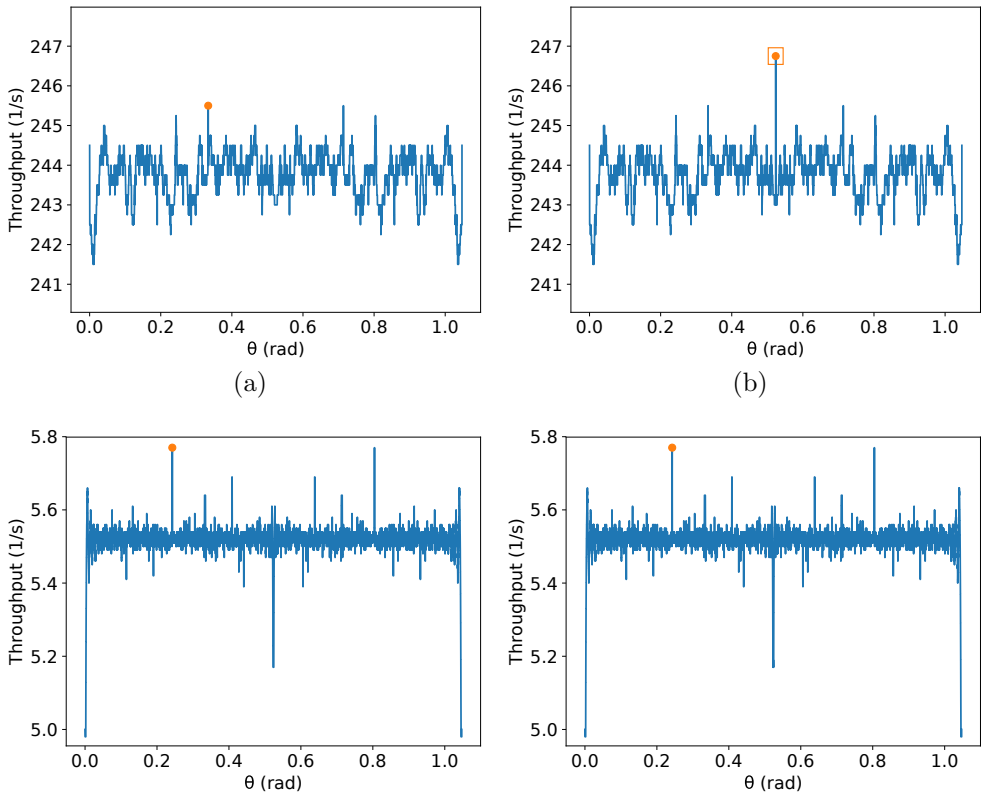


Figure 4.11: Continuation of Figure 4.10: examples similar to Figures 4.8 and 4.9 but using 10^7 and 10^7+1 equally spaced points for $\theta \in [0, \pi/3)$. (a) For 10^7 samples, $t = 4$ s, $s = 2$ m and $d = 0.13$ m. (b) For $10^7 + 1$ samples, $t = 4$ s, $s = 2$ m and $d = 0.13$ m. (c) For 10^7 samples, $t = 100$ s, $s = 2.40513$ m and $d = 1$ m. (d) For $10^7 + 1$ samples, $t = 100$ s, $s = 2.40513$ m and $d = 1$ m.

parameters. In Figures 4.8 and 4.9, 99 equally spaced values are shown for $\theta \in [0, \pi/3)$ on the left-hand side images and 100 on the right-hand side; then, the maximum on each side is compared, and the best one is chosen. The same is performed in Figures 4.10 and 4.11, but using 10^7 and $10^7 + 1$. Figures 4.8a, 4.9a, 4.10b and 4.11b show an example that $\theta \approx \pi/6$ reaches the maximum throughput, and in Figures 4.8c, 4.8d, 4.10c and 4.10d, the maximum is at $\theta = 0$. Moreover, Figures 4.9c and 4.9d have their maximum for θ different from the other examples. Figure 4.8c and 4.8d have the same maximum, despite the plots being different. This also occurs in Figures 4.10c, 4.10d, 4.11c and 4.11d. If the parameters are known, one can find an approximate best candidate for θ by searching several values, as presented. However, as far as I know, obtaining the true value which maximises that equation by a closed-form is an open problem.

Additionally, notice that whenever the number of samples is odd, the value $\theta = \pi/6$ is sampled. Observe in these figures that when the maximum is at $\theta = \pi/6$, it tends to be higher than the maximum found without considering it. For instance, compare the maximum found on the pairs (a) and (b) in Figures 4.8–4.11. On the other hand, $\theta = \pi/6$ is not always the optimal value. Thus, I suggest to compute first the value for $\theta = \pi/6$, then compare it with the result for a search for the maximum for any chosen number of samples in the interval from $\theta \in [0, \pi/3)$.

4.5.3 Touch and Run Strategy

Now, the *touch and run* strategy is discussed. Since a robot should spend as little time as possible near the target, a simple scenario is imagined where robots travel in predefined curved lanes and tangent to the target area where they spend minimum time on the target. To avoid collisions with other robots, the trajectory of a robot nearby the target is circular, and the distance between each robot must be at least d at any part of the trajectory. Hence, no lane crosses another, and each lane occupies a region defined by an angle in the target area, denoted by α and shown in Figure 4.12a.

Figure 4.12b shows the trajectory of a robot towards the target region following that strategy. This figure also shows the relationship between the target area radius (s), the minimum safety distance between the robots (d), the turning radius (r), the central region angle (α) and the distance from the target centre for a robot to begin turning (d_r) – used as justification for (4.12) and (4.13). The green dashed circle represents the whole turning circle. The robot first follows the boundary of the central angle region – that is, the entering ray – at a distance of $d/2$. Then, it arrives at a distance of s of the target centre using a circular trajectory with a turning radius r. Due to the trajectory being tangent to the target shape, it is close enough to consider that the robot reached the target region.

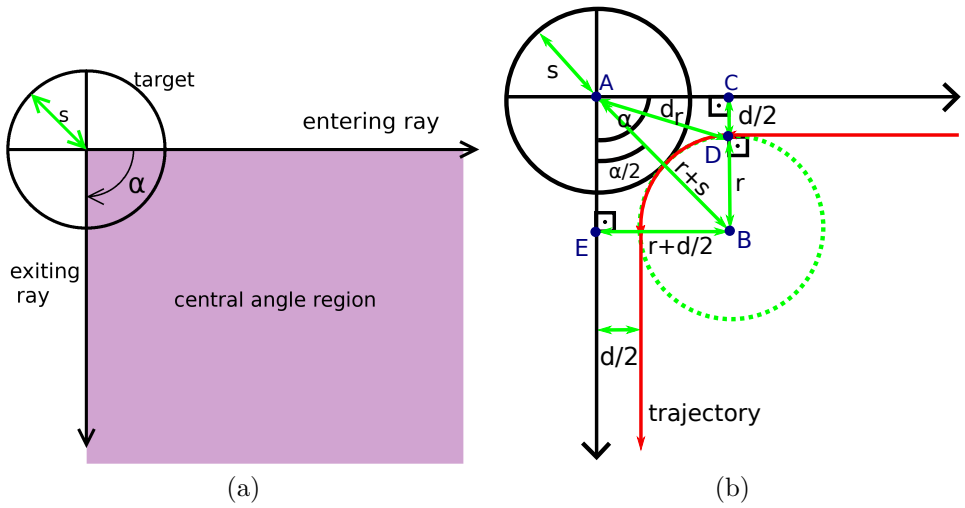


Figure 4.12: Illustration of the touch and run strategy. (a) Central angle region and its exiting and entering rays defined by the angle α. (b) Trajectory of a robot next to the target in red.

Finally, the robot leaves the target by following the second boundary of the central angle region – that is, the exiting ray – at a distance of $d/2$. Depending on the value of α, it is possible to fit several of these lanes around the target. For example, in Figure 4.13, when $\alpha = \pi/2$, it is possible to fit four lanes. In this figure, robots are black dots, and d_o is the desired distance between the robots in the same lane – which is calculated depending on the values of d, s, r and the number of lanes K as shown later. This desired distance is employed because the robots cannot reduce their speed to maintain the fixed distance d in the curved path, as they are supposed to have a constant linear speed. Consequently, $d_o \geq d$. When robots of all lanes simultaneously occupy the target region, their positions are the vertices of a regular polygon – it is represented in the figure by a grey square inside the target region.

The lemma below concerns the distance to the target centre where the robots start turning on the curved path. It will also be useful in the discussion about experiments using this strategy in Section 4.6.4.

Lemma 2. *The distance d_r to the target centre for the robot to start turning is*

$$d_r = \sqrt{s(2r+s) - rd}. \tag{4.12}$$

Proof. Figure 4.12b shows the distance d_r from the target centre where the robots begin turning. By symmetry, this is the same distance from the target centre

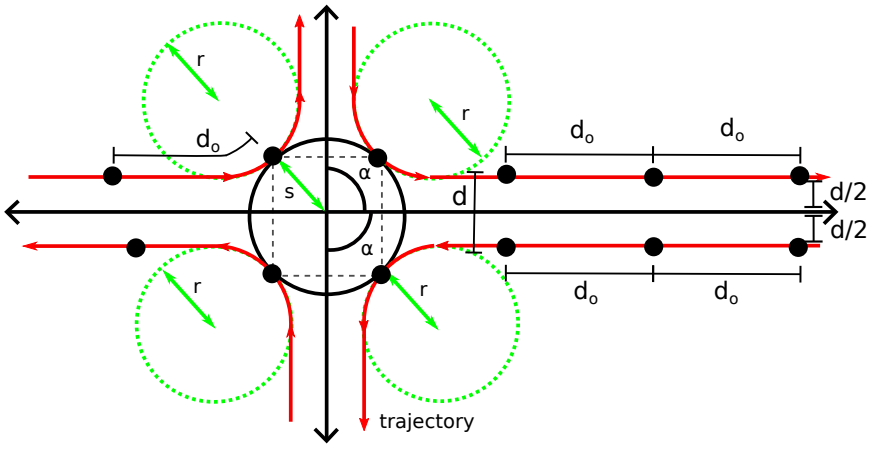


Figure 4.13: Theoretical trajectory in red, for $\alpha = \pi/2$ and $K = 4$.

where the robots stop turning. From the right triangle ABC on that figure, $|\overline{AC}| = \sqrt{(r+s)^2 - (r+d/2)^2}$ and from $\triangle ACD$, $d_r = \sqrt{(d/2)^2 + |\overline{AC}|^2}$. Thus,

$$d_r = \sqrt{(d/2)^2 + (r+s)^2 - (r+d/2)^2} = \sqrt{s(2r+s) - rd}.$$

□

Now, a lemma about the turning radius is presented, and then the domain of K and α are defined in order to calculate the throughput for the touch and run strategy.

Lemma 3. *The central region angle α, the minimum distance between the robots d and the turning radius r are related by*

$$r = \frac{s\sin(\alpha/2) - d/2}{1 - \sin(\alpha/2)}. \tag{4.13}$$

Proof. From Figure 4.12b, it can be seen that the right triangle ABE has angle $\widehat{EAB} = \alpha/2$, hypotenuse $r+s$ and cathetus $r + d/2$. Hence, it directly follows that

$$sin(\alpha/2) = \frac{r + d/2}{r+s} \Leftrightarrow r = \frac{s\sin(\alpha/2) - d/2}{1 - \sin(\alpha/2)}.$$

□

Proposition 7. *Let K be the number of curved trajectories around the target area, α be the angle of each central area region, and r the turning radius of the robot for the curved trajectory of this central area region. For a given $d > 0$ and $s \geq d/2$, the domain of K is*

$$3 \leq K \leq \frac{\pi}{\arcsin\left(\frac{d}{2s}\right)}, \text{ and} \tag{4.14}$$

$$\alpha = \frac{2\pi}{K}. \tag{4.15}$$

Proof. The number of trajectories K must be greater or equal to 3. The reason is that for the minimum possible value for s, $s = d/2$, $K = 2$ is enough to have parallel lanes. However, starting with $K = 3$, curved trajectories are needed to guarantee that robots of one lane do not interfere with robots from another lane.

Also, there are K identical trajectories around the target, each taking a central angle of α. As a result, the value of α given K is $\alpha = \frac{2\pi}{K}$, implying that $0 < \alpha \leq \frac{2\pi}{3}$.

Additionally, in the worst case, one robot in each lane arrives in the target region at the same time. When robots of all lanes simultaneously occupy the target region, their positions can be seen as the vertices of a regular polygon which must be inscribed in the circular target region of radius s (e.g., Figure 4.13 has a square whose sides are greater than d). The number of robots on the target region at the same time must be limited by the maximum number of sides of an inscribed regular polygon with a minimum side greater than or equal to d. The side of a K regular polygon inscribed in a circle of radius s measures $2s\sin\left(\frac{\pi}{K}\right)$. Thence, $2s\sin\left(\frac{\pi}{K}\right) \geq d \Rightarrow \frac{\pi}{\arcsin\left(\frac{d}{2s}\right)} \geq K$. □

Now that the correct parametrisation has been determined for the touch and run strategy, its throughput is obtained in the next proposition. Heed that the value of K below is constrained by the interval in Proposition 7. Consequently, the throughput of the touch and run strategy cannot increase indefinitely with K.

Proposition 8. *Assuming the touch and run strategy and that the first robot of every lane begins at the same distance from the target, given a target radius s, the constant linear robot speed v, a minimum distance between robots d, and the number of lanes K, the throughput for a given instant t is calculated by*

$$f_t(K,t) = \frac{1}{t}\left(K\left\lfloor\frac{vt}{d_o}+1\right\rfloor - 1\right), \text{ for} \tag{4.16}$$

$$d_o = \max(d, d'), \text{ and} \tag{4.17}$$

$$d' = \begin{cases} r(\pi-\alpha) + \frac{d-2r\cos(\alpha/2)}{\sin(\alpha/2)}, & \text{if } 2r\cos(\alpha/2) < d, \\ 2r\arcsin\left(\frac{d}{2r}\right), & \text{otherwise,} \end{cases} \tag{4.18}$$

with r obtained from (4.13). In addition,

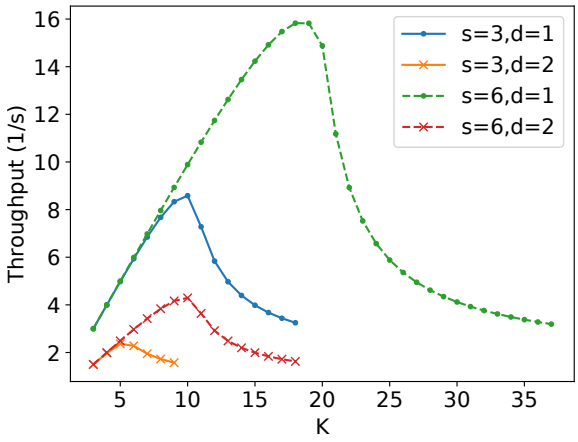


Figure 4.14: Plot of the asymptotic throughput of the touch and run strategy (given by (4.19)) for some values of s and d, in metres, and $v = 1$ m/s, for the interval of values for K obtained by (4.14).

$$f_t(K) = \lim_{t\to\infty} f_t(K,t) = \frac{Kv}{d_o}. \tag{4.19}$$

Idea of the proof. The calculation of the throughput at a given time for the touch and run strategy considers two cases: (i) when two robots cannot be on the lane curved path and (ii) when more than one robot can occupy the lane curved path. After this throughput is obtained, the limit for the time going to infinity is given. The formal proof is in Section B.5. □

Figure 4.14 presents examples of (4.19) for some parameters. Observe that the maximum throughput for different values of s, d and v can be found by a linear search in the interval obtained by (4.14). By assuming that K is a real number, the optimal value of K could be obtained by adequately choosing an integer K next to a value such that $\frac{df_t}{dK} = 0$ by substituting in (4.18) α by $2\pi/K$ and r by its definition in (4.13). However, using a linear search is easier than obtaining the closed form of $\frac{df_t}{dK}$ because it has a complicated combination of trigonometric functions with different cases depending on s, d and K, overburdening the function inversion to attain K. Table 4.1 summarises the instantaneous and asymptotic throughput presented in this chapter, indicating the proposition that states them and the symbols used for them.

Symbol	Proposition	Description
f	Proposition 1	Maximum throughput for a point-like target ($s = 0$) when robots form a single line
$f(t)$	Proposition 2	Instantaneous throughput of the parallel lanes strategy for a small target area and $0 < s \leq \frac{\sqrt{3}}{4}d$.
f	Proposition 2	Asymptotic throughput of the parallel lanes strategy for a small target area and $0 < s \leq \frac{\sqrt{3}}{4}d$.
$f(t)$	Proposition 3	Instantaneous throughput of the parallel lanes strategy for a small target area and $\frac{\sqrt{3}}{4}d < s < \frac{d}{2}$.
f	Proposition 3	Asymptotic throughput of the parallel lanes strategy for a small target area and $\frac{\sqrt{3}}{4}d < s < \frac{d}{2}$.
$f_p(t)$	Proposition 4	Instantaneous throughput of the parallel lanes strategy for a large target area ($s \geq \frac{d}{2}$).
f_p	Proposition 4	Asymptotic throughput of the parallel lanes strategy for a large target area ($s \geq \frac{d}{2}$).
f_h^{max}	Proposition 5	Upper bound of the asymptotic throughput of the hexagonal packing strategy for a large target area ($s \geq \frac{d}{2}$).
$f_h(t, \theta)$	Proposition 6	Instantaneous throughput of the hexagonal packing strategy for a large target area ($s \geq \frac{d}{2}$) for a given hexagonal packing angle θ.
$f_h(\theta)$	Proposition 6	Asymptotic throughput of the hexagonal packing strategy for a large target area ($s \geq \frac{d}{2}$) for a given hexagonal packing angle θ (only lower and upper bounds were given).
$f_t(K, t)$	Proposition 8	Instantaneous throughput of the touch and run strategy for a large target area ($s \geq \frac{d}{2}$) for a given number of lanes K.
$f_t(K)$	Proposition 8	Asymptotic throughput of the touch and run strategy for a large target area ($s \geq \frac{d}{2}$) for a given number of lanes K.

Table 4.1: Summary of the instantaneous and asymptotic throughput of the presented strategies.

4.5.4 Comparison of the Strategies

The parallel lanes strategy has the lowest of the limits concerning $u = \frac{s}{d}$, the ratio between the radius of the target region and the minimum distance between the robots. However, its asymptotic value is still higher than the minimum possible asymptotic throughput for hexagonal packing just for some values of u. This section will make explicit the dependence on the argument u in every throughput function defined previously to compare them to this ratio. Let $f_p(u) = \lim_{t\to\infty} f_p(t,u)$ and $f_h^{min}(u)$ be the asymptotic throughput for the parallel lanes strategy and the lower bound of the asymptotic throughput for the hexagonal packing strategy for a ratio u, respectively. Hence, by Proposition 4,

$$f_p(u) = \lfloor 2u+1 \rfloor \frac{v}{d},$$

and by (4.11) using $\theta = \pi/6$ as it minimises the lower bound of $\lim_{t\to\infty} f(t,\theta)$ in Proposition 6,

$$f_h^{min}(u) = \frac{2}{\sqrt{3}}(2u-1)\frac{v}{d}.$$

Proposition 9. *There are some $u < \frac{\sqrt{3}+2}{4-2\sqrt{3}}$ such that $f_p(u) > f_h^{min}(u)$, and for every $u \geq \frac{\sqrt{3}+2}{4-2\sqrt{3}}$, $f_p(u) \leq f_h^{min}(u)$.*

Proof. For any $u < \frac{\sqrt{3}+2}{4-2\sqrt{3}}$, $(2u+1)\frac{v}{d} > f_h^{min}(u)$, due to

$$\begin{aligned} &(2u+1)\frac{v}{d} > \frac{2}{\sqrt{3}}(2u-1)\frac{v}{d} \Leftrightarrow 2u+1 > \frac{2}{\sqrt{3}}(2u-1) \\ \Leftrightarrow\ &2u - \frac{4}{\sqrt{3}}u > -1 - \frac{2}{\sqrt{3}} \Leftrightarrow u < \frac{-1-\frac{2}{\sqrt{3}}}{2-\frac{4}{\sqrt{3}}} = \frac{-\sqrt{3}-2}{2\sqrt{3}-4} = \frac{\sqrt{3}+2}{4-2\sqrt{3}}. \end{aligned} \tag{4.20}$$

$f_p(u) = (2u+1)\frac{v}{d}$ when $2u+1 \in \mathbb{Z}$. Also, as $u < \frac{\sqrt{3}+2}{4-2\sqrt{3}} < 7$, u can be a number satisfying $(2u+1) = \lfloor 2u+1 \rfloor$. Thus, there are some values of u such that $f_p(u) = \lfloor 2u+1 \rfloor \frac{v}{d} > f_h^{min}(u)$.

From the equivalence in (4.20) and because for any x, $\lfloor x \rfloor \leq x$, it follows that for any $u \geq \frac{\sqrt{3}+2}{4-2\sqrt{3}}$, $f_p(u) \leq (2u+1)\frac{v}{d} \leq f_h^{min}(u)$. □

Figure 4.15 shows an example of $f_h^{min}(u)$, $f_p(u)$ and the maximum possible asymptotic throughput of the hexagonal packing $f_h^{max}(u) = \frac{2}{\sqrt{3}}(2u+1)\frac{v}{d}$ for $u \in [0, 10]$. Observe that, from the left side of $u = 7$, $f_p(u)$ has some values above $f_h^{min}(u)$ even though they are below $f_h^{max}(u)$ for every u.

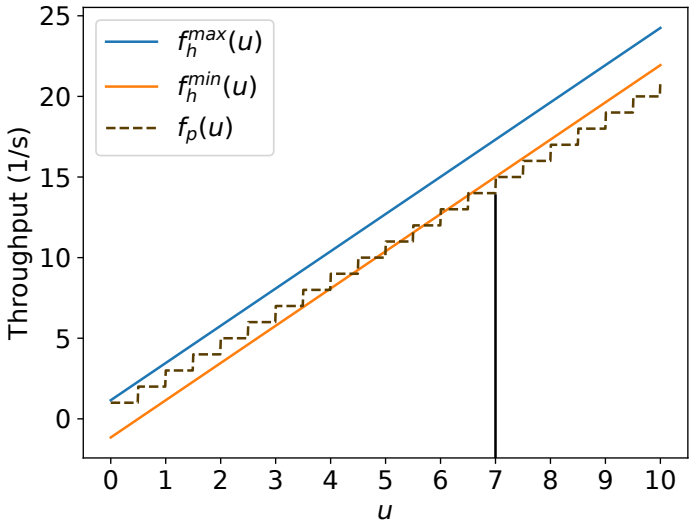


Figure 4.15: Example of u values such that $f_h^{max}(u) > f_p(u)$ for $v = 1$ m/s and $d = 1$ m.

Because of this proposition, for values of $u \geq \frac{\sqrt{3}+2}{4-2\sqrt{3}} \approx 7$, the hexagonal packing strategy at the limit will have higher throughput than parallel lanes. However, for values $u < \frac{\sqrt{3}+2}{4-2\sqrt{3}}$, there is the possibility of the parallel lanes strategy being better than hexagonal packing. As there is not an exact asymptotic throughput for the hexagonal packing strategy for a given angle θ, one can numerically find the best θ using large values of t on (4.9); then, after choosing θ, the numerical approximation of the asymptotic throughput using this fixed θ and those t values is calculated. This result can be compared with the throughput for the same large values of t for the parallel lanes strategy using (4.6). Furthermore, in a scenario with the target region only being accessed by a corridor with a finite height, the maximum time t can be inferred by its size, and then the exact throughput for this specific value can be calculated by (4.9) and (4.6) as stated before, but using only this specific value t, instead of a set of large values, to decide which strategy is more suitable.

Let $f_h(t, \theta, u)$ and $f_p(t, u)$ be (4.9) and (4.6) making explicit the parameter u. Let θ^* be the outcome from the search of the θ, which maximises $f_h(t, \theta, u)$ by numeric approximation. Thus, define $f_h(t, u) = f_h(t, \theta^*, u)$. Figure 4.16 illustrates the result of the procedure mentioned above for $t = 10{,}000$ for 100 equally spaced values of $u \in [0, 7]$ and seeking the maximum throughput using 1000 evenly spaced points between $[0, \pi/3)$ to find the best θ for the hexagonal packing strategy. Then, it is compared with the result for $\theta = \pi/6$ as explained previously when Figures 4.8–4.11 were discussed. Observe that for $u \in [0.5, 0.9]$ there is some values for which $f_h(10{,}000, u) < f_p(10{,}000, u)$. Figure 4.17 shows this by 100 equally spaced values

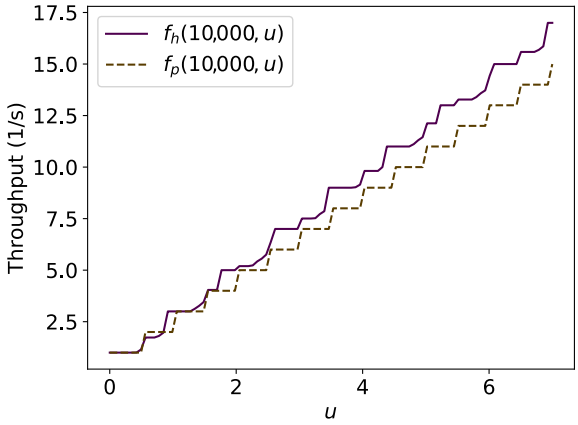


Figure 4.16: Comparison of $f_p(t,u)$ and $f_h(t,u)$ for $u \in [0,7]$, $t = 10{,}000$ s, $v = 1$ m/s and $d = 1$ m.

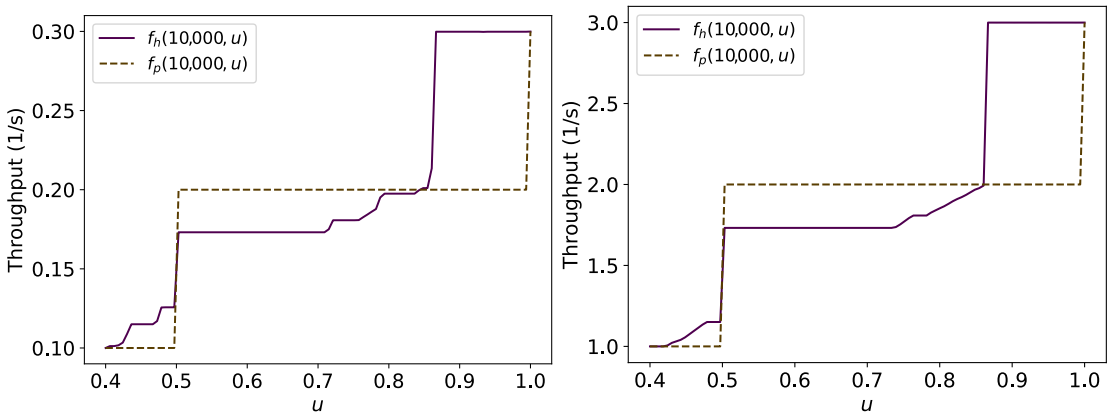
of $u \in [0.4, 1]$ for different values of v. This occurs because, for such values of u, using square packing fits more robots inside the circle over the time than hexagonal packing, as shown in Section 4.6.5.

Additionally, let $f_t(K,t,u)$, the throughput of the touch and run strategy for a given number of lanes, a time and a ratio u explicit in the argument list, and $f_t(u,K) = \lim_{t\to\infty} f_t(K,t,u)$. The asymptotic throughput of the touch and run strategy, $f_t(u) = \max_K f_t(u,K)$, for higher values of u is greater than the maximum possible asymptotic value of the hexagonal packing $f_h^{max}(u) = \frac{2}{\sqrt{3}}(2u+1)\frac{v}{d}$, as shown later by numeric experimentation. Before presenting this result, it is necessary to verify which values of u are allowed by $f_t(u)$ and to express the asymptotic throughput of the touch and run strategy from Proposition 8 in terms of the ratio u.

From Proposition 7, the possible number of lanes K is in $\{3,\ldots,K(u)\}$ with $K(u) = \left\lfloor \frac{\pi}{\arcsin\left(\frac{1}{2u}\right)} \right\rfloor$. Consequently, $f_t(u)$ is only allowed for any $u \geq \frac{1}{\sqrt{3}}$. In fact, by Proposition 7, $K \geq 3$, then $\frac{\pi}{\arcsin\left(\frac{1}{2u}\right)} \geq \left\lfloor \frac{\pi}{\arcsin\left(\frac{1}{2u}\right)} \right\rfloor \geq 3 \Rightarrow \frac{\pi}{3} \geq \arcsin\left(\frac{1}{2u}\right) \Leftrightarrow \sin\left(\frac{\pi}{3}\right) \geq \frac{1}{2u} \Leftrightarrow \frac{\sqrt{3}}{2} \geq \frac{1}{2u} \Leftrightarrow u \geq \frac{1}{\sqrt{3}}$.

The algebraic manipulations for expressing the asymptotic throughput of the touch and run strategy from Proposition 8 is shown below in terms of the ratio u. The asymptotic throughput expressed in (4.19) is

$$\begin{aligned} \frac{Kv}{d_o} &= \frac{K}{\frac{d_o}{d}}\frac{v}{d} = \frac{K}{\frac{\max(d,d')}{d}}\frac{v}{d} \quad [(4.17)] \\ &= \frac{K}{\max(1,\frac{d'}{d})}\frac{v}{d}, \end{aligned} \tag{4.21}$$



(a) (b)

Figure 4.17: Comparison of f_p and f_h for $u \in [0.4, 1]$, $t = 10{,}000$ s, $v \in \{0.1, 1\}$ m/s and $d = 1$ m. The difference in the lines of f_h is due to θ^* being different for each value of v. (a) $v = 0.1$ m/s; (b) $v = 1$ m/s

.

for an integer $K \in \{3, \ldots, K(u)\}$. From (4.15), $\alpha = \frac{2\pi}{K}$, and, from (4.13),

$$\frac{r}{d} = \frac{\frac{s}{d}\sin(\alpha/2) - \frac{d}{2d}}{1 - \sin(\alpha/2)} = \frac{u\sin(\frac{\pi}{K}) - \frac{1}{2}}{1 - \sin(\frac{\pi}{K})} \stackrel{\text{def}}{=} r(u, K),$$

resulting in

$$\begin{aligned}
\frac{d'}{d} &= \begin{cases} \frac{r}{d}(\pi - \alpha) + \frac{d - 2r\cos(\alpha/2)}{d\sin(\alpha/2)}, & \text{if } 2r\cos(\alpha/2) < d, \\ 2\frac{r}{d}\arcsin\left(\frac{d}{2r}\right), & \text{otherwise,} \end{cases} \quad \text{[by (4.18)]} \\
&= \begin{cases} \frac{r}{d}\left(\pi - \frac{2\pi}{K}\right) + \frac{1 - 2\frac{r}{d}\cos(\frac{\pi}{K})}{\sin(\frac{\pi}{K})}, & \text{if } 2\frac{r}{d}\cos(\frac{\pi}{K}) < 1, \\ 2\frac{r}{d}\arcsin\left(\left(2\frac{r}{d}\right)^{-1}\right), & \text{otherwise,} \end{cases} \\
&= \begin{cases} r(u,K)\left(\pi - \frac{2\pi}{K}\right) + \frac{1 - 2r(u,K)\cos(\frac{\pi}{K})}{\sin(\frac{\pi}{K})}, \\ \qquad\qquad \text{if } 2r(u,K)\cos(\frac{\pi}{K}) < 1, \\ 2r(u,K)\arcsin\left(\frac{1}{2r(u,K)}\right), \text{ otherwise,} \end{cases} \\
&\stackrel{\text{def}}{=} d'(u, K).
\end{aligned} \tag{4.22}$$

Thus, from (4.21) and (4.22), $f_t(u, K) = \frac{K}{\max(1, d'(u,K))}\frac{v}{d}$, and the upper throughput

for the touch and run strategy in terms of u is given by

$$f_t(u) = \max_{K\in\{3,\dots,K(u)\}} f_t(u,K) = \max_{K\in\{3,\dots,K(u)\}} \frac{K}{\max(1, d'(u,K))}\frac{v}{d}$$
$$= \frac{K^*(u)}{\max(1, d'(u,K^*(u)))}\frac{v}{d},$$

for some function $K^*(u)$ that finds this maximum in $\{3,\dots,K(u)\}$. Similarly, for a fixed maximum time t, by (4.16), $f_t(t,u) = \max_{K\in\{3,\dots,K(u)\}} f_t(K,t,u)$.

Figure 4.18 presents a comparison of the asymptotic throughput $f_t(u)$ and the lower and upper values of the asymptotic throughput of the hexagonal packing $f_h^{min}(u)$ and $f_h^{max}(u)$ for values of u ranging from $1/\sqrt{3}$ to 1000. Observe that the asymptotic throughput of the touch and run strategy is greater than the maximum possible asymptotic throughput of the hexagonal packing strategy for almost all values of u, except for some in $(1.12, 1.25)$ (Figure 4.18b).

Additionally, numerical experiments for $f_t(t,u)$ and $f_h(t,u)$ are performed using fixed time t = 10,000 in (4.16), (4.9) and $u \in [1/\sqrt{3}, 7]$. For finding θ^*, the same procedure is applied, which was described before to compare $f_h(t,u)$ and $f_p(t,u)$. Figure 4.19 shows the result. It suggests the touch and run strategy has higher throughput than hexagonal packing for large values of t. Although hexagonal packing has lower asymptotic throughput than the touch and run strategy for almost all u values, it is suitable for $u < \frac{1}{\sqrt{3}}$ whenever it surpasses the parallel lanes strategy.

For real-world applications and assuming the robots are constantly at maximum linear speed and at fixed distance between other robots, the hexagonal packing strategy is adequate for a situation where the target is placed in a constrained region, for example, walls in north and south positions. In this example, the number of lanes used in the touch and run strategy would be reduced because of the surrounding walls. In an unconstrained scenario, if the ratio u and the maximum time t are known, the throughput value of the hexagonal packing strategy from (4.9) (for the θ which maximises it) can be compared with the throughput of the touch and run strategy from (4.16) (for $K^*(u)$) to choose which strategy should be applied. However, assuming a constant speed and a fixed minimum distance between robots in a swarm is not practical because other robots influence the movement in the environment. Hence, these strategies are the inspiration to propose novel algorithms based on potential fields for robotic swarms in Chapter 5.

4.6 Experiments and Results

In this section, the theoretical strategies and equations presented in this chapter are evaluated by realistic Stage (Gerkey, Vaughan, and Howard, 2003) simulations with

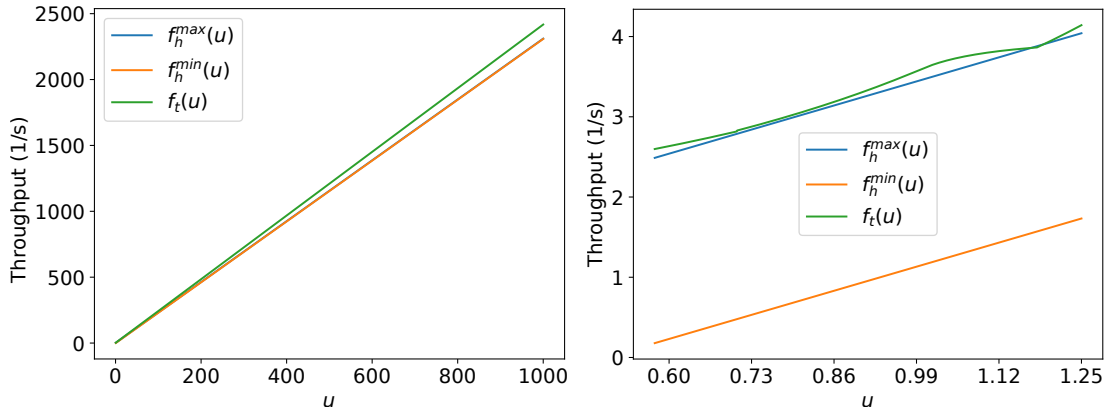


Figure 4.18: Graph varying u for $f_h^{min}(u)$, $f_h^{max}(u)$ and $f_t(u)$ with $v = 1$ m/s and $d = 1$ m for different intervals of u. In (a), $f_h^{min}(u)$ and $f_h^{max}(u)$ are almost overlapped. In (b), $f_t(u) > f_h^{max}(u)$ for all u, except in an interval within (1.12,1.25). (a) $u \in [1/\sqrt{3}, 1000]$; (b) $u \in [1/\sqrt{3}, 1.25]$.

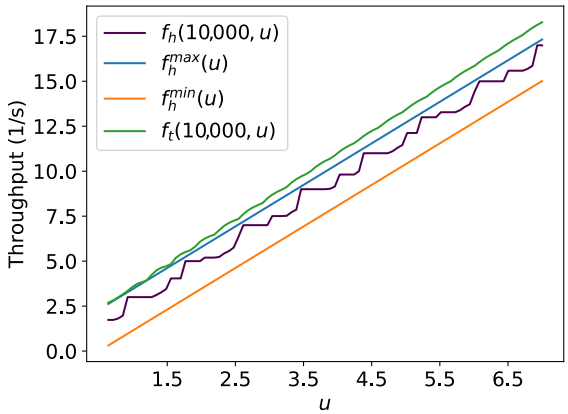


Figure 4.19: Example for $t = 10{,}000$ s, $v = 1$ m/s, $d = 1$ m and 100 equally spaced points of $u \in [1/\sqrt{3}, 7]$. $f_h(t,u) < f_t(t,u)$, albeit $f_h^{max}(u) \geq f_t(t,u)$ for a few values of $u < 1.5$.

holonomic and non-holonomic robots. Aditionally, these experiments corroborate that whenever an algorithm makes a swarm take less time to reach the target region than another algorithm, the throughput of the former is higher than the latter.

Hyperlinks to the video of executions are available in the captions of each corresponding figure. They are in real-time so that the reader can compare the time and screenshots presented in the figures in this section with those in the supplied videos. (The source codes of each experimented strategy are in Passos (2022c).) In addition, Appendix C also shows some of these screenshots and more graphs, adding to the results in this section.

Experiments were executed for all strategies considering $s > 0$. Experiments for point-like targets could not be performed because a point with a fixed value is nearly impossible to be reached by a moving robot in Stage computer simulations due to the necessity of exact synchronization of the sampling frequency of positions made by the simulator and the speed of the robot. Hence, a circular area with a radius $s > 0$ around the target must be used to identify that a robot reached it. After presenting the experiments and results for all strategies for circular target region with radius $s > 0$, they are compared experimentally considering the analysis previously discussed in Section 4.5.4.

It is saved for each robot its arrival time in milliseconds since the start of the experiment. The arrival time of every robot is subtracted by the arrival time of the first robot. By doing so, the experiment is assumed to begin in time $t = 0$ without worrying about the initial inertia. After this, the number of robots (N) is registered for each time value (t).

To alleviate some of the numerical errors caused by the floating-point representation, rounding on the 13th decimal place was used before using floor and ceiling functions on the equations presented. For example, in contemporary computers, by using double variables in C or float in Python, if you divide 9.6 by 1.6, the result is 5.999999999999999 for 15 decimal places formatting, but it should be 6. If the floor function was applied to the previous result, the outcome would be 5 instead of the expected 6.

For all experiments in this section, the robots are distant from each other by $d = 1$ m. In the figures of this section, black robots indicate they reached the target, and red did not. In addition, the experiments shown on this section were not repeated because the linear speed and initial positions are constant, so there is no random aspect, and the same results are obtained for different runs.

4.6.1 Compact Lanes

For compact lanes simulations, $v = 1$ m/s, and the first robot to reach the target is at the bottom lane and starts at the target. For a target area radius s, such that

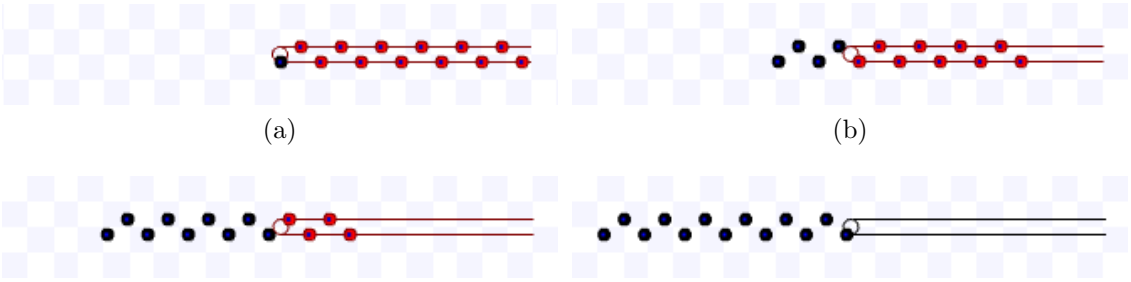


Figure 4.20: Simulation on Stage for compact lanes strategy using $s = 0.3$ m, $d = 1$ m during $t = 7.1$ s. Available on `https://youtu.be/e1cWJzWhQmQ`, accessed on 12 June 2022. (a) 0 s: beginning of the simulation; (b) After 2.7 s; (c) After 6.7 s; (d) 5 s: ending of the simulation.

$0 < s < \sqrt{3}d/4$, $s = 0.3$ m, and for $\sqrt{3}d/4 \leq s < d/2$, $s = 0.45$ m. Figure 4.20 shows screenshots of the simulation using $s = 0.3$ m during $t = 7.1$ s and Figure 4.21 for $s = 0.45$ m and $t = 10.1$ s.

Experiments were run in order to verify the throughput for a given time and the asymptotic throughput calculated by (4.2) to (4.5). Figure 4.22 shows the throughput for different values of time obtained by the experiments in Stage, i.e., $(N-1)/t$, in comparison with the calculated value by (4.2) and (4.3) for $s = 0.3$ m and by (4.4) and (4.5) for $s = 0.45$ m. "Simulation" stands for the data obtained from Stage, "Instantaneous" for the equations of the throughput for a given time calculated in (4.2) and (4.4) and "Asymptotic" for the asymptotic throughput obtained from (4.3) and (4.5). The mentioned results of the equations match the data obtained from simulations. These figures confirm that the equations presented in the theoretical section agree with the throughput obtained by simulations.

4.6.2 Parallel Lanes

The parallel lanes strategy was experimented for $v = 1$ m/s and $s \in \{3, 6\}$ m. Figures 4.23 and 4.24 present screenshots from executions using these parameters.

To verify the throughput for a given time calculated by (4.6) and its asymptotic value as in (4.7), they are compared with the throughput obtained from Stage simulations. Figure 4.25a presents these comparisons. "Simulation" stands for the data obtained from Stage, "Instantaneous" for the equations of the throughput for a given time calculated in (4.6), and "Asymptotic" for the asymptotic throughput obtained from (4.7). As expected, the values of (4.6) approximate to (4.7) as time passes. Additionally, observe that the values from (4.6) are almost aligned with the

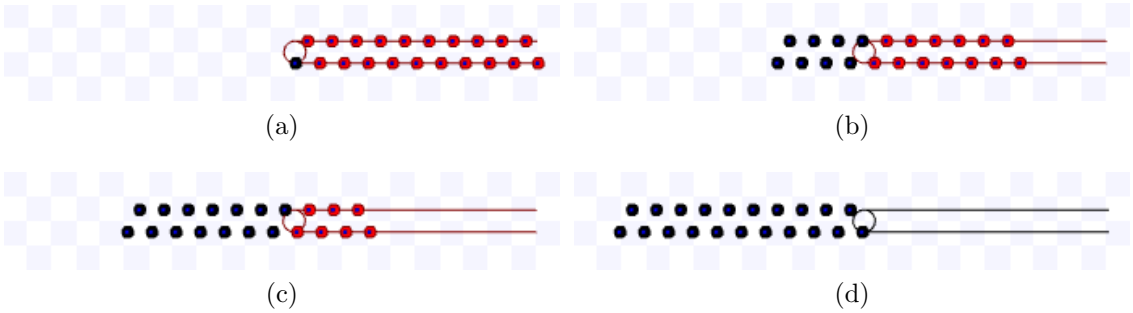


Figure 4.21: Simulation on Stage for compact lanes strategy using $s = 0.45$ m, $d = 1$ m during $t = 10.1$ s. Available on `https://youtu.be/9OXGC1w83j0`, accessed on 12 June 2022. (a) 0 s: beginning of the simulation; (b) After 3.5 s; (c) After 7 s; (d) 10.1 s: ending of the simulation.

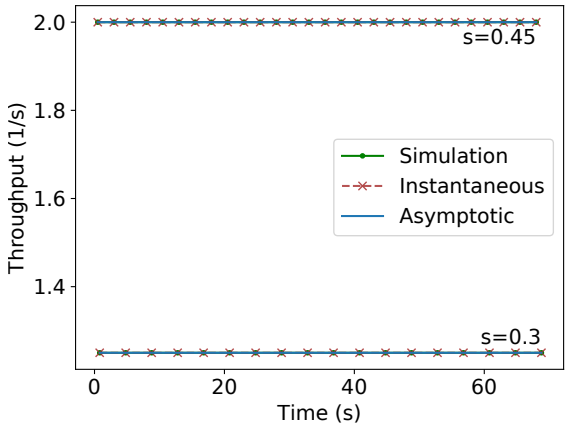
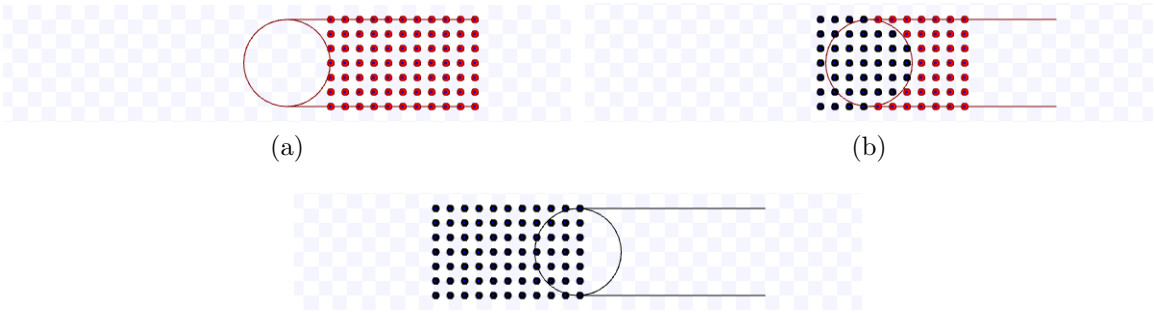


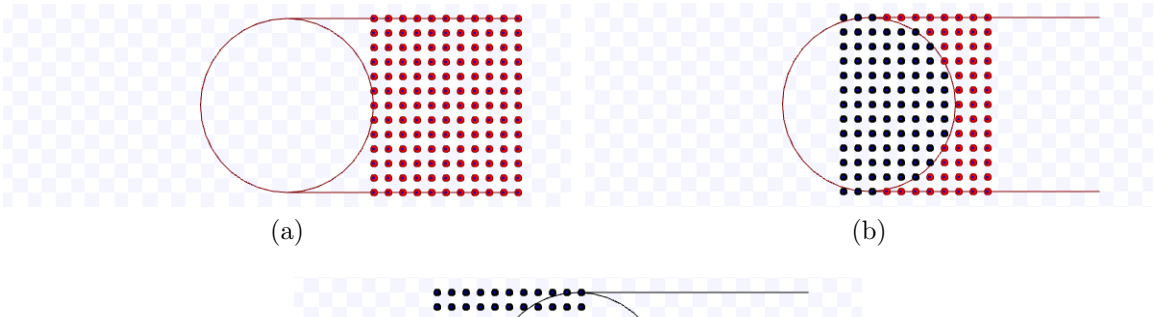
Figure 4.22: Throughput versus time plot for compact lanes strategy for different values of s.



(a) (b)

(c)

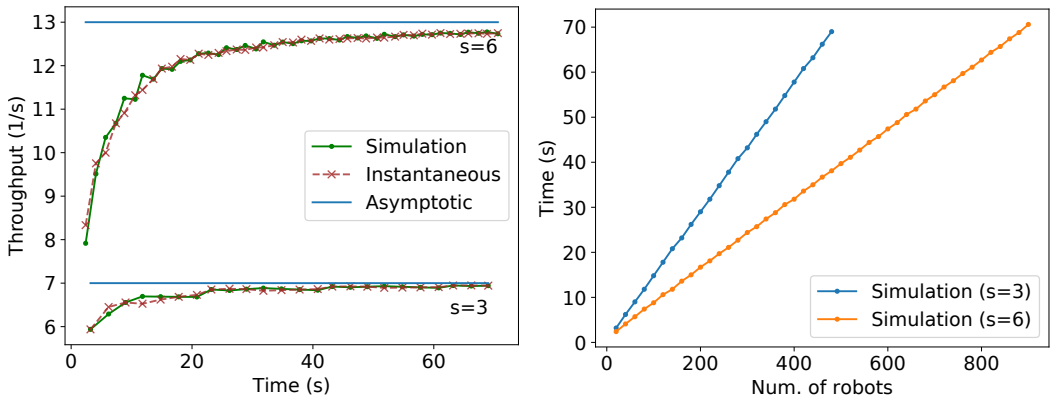
Figure 4.23: Simulation on Stage for parallel lanes strategy using $s = 3$ m, $d = 1$ m during $t = 13$ s. Available on `https://youtu.be/2Y1RHc9YVaw`, accessed on 12 June 2022. (a) 0 s: beginning of the simulation; (b) After 6.5 s; (c) 13 s: ending of the simulation.



(a) (b)

(c)

Figure 4.24: Simulation on Stage for parallel lanes strategy using $s = 6$ m, $d = 1$ m during $t = 16$ s. Available on `https://youtu.be/TVdka65fi1g`, accessed on 12 June 2022. (a) 0 s: beginning of the simulation; (b) After 8 s; (c) 16 s: ending of the simulation.



(a) (b)

Figure 4.25: Plots for the experiments of parallel lanes strategy for $s \in \{3, 6\}$ m. (a) Number of robots versus throughput. (b) Number of robots versus the time of arrival of the last robot.

values from the simulation, except for some points. The difference in those points is due to the floating-point error discussed at the beginning of Section 4.6 that happens in the division before the use of floor or ceiling functions used on (4.6). Figure 4.25b shows the number of robots versus the time of arrival of the last robot for the same data used in Figure 4.25a. As the running time is proportional to the number of robots in the experiments, observe that the higher throughput per time is reflected as a lower arrival time of the last robot per the number of robots. In addition, note that the values tend to infinity as the horizontal axis values grow.

4.6.3 Hexagonal Packing

The hexagonal packing was experimented for $v = 1$ m/s and the combination of the following variables and values: $s \in \{3, 6\}$ m and $\theta \in \{0, \pi/12, \pi/6, 5\pi/18\}$. Figure 4.26 presents screenshots from executions using $\theta = 0$ and $s = 3$ m. (The screenshots with the other values of hexagonal packing angle θ and radius s are presented in Section C.1.)

To evaluate the throughput for a given time and angle calculated in (4.9) and the bounds on the asymptotic throughput as in (4.11), they are compared with the throughput obtained from Stage simulations. Figure 4.27 presents these comparisons. Observe that the values from (4.9) are almost aligned with the values from the simulation, except for some points. The difference in those points is also due to the floating-point error – discussed in the introduction of Section 4.6 – over the divisions and trigonometric functions performed before the use of floor or ceiling functions used

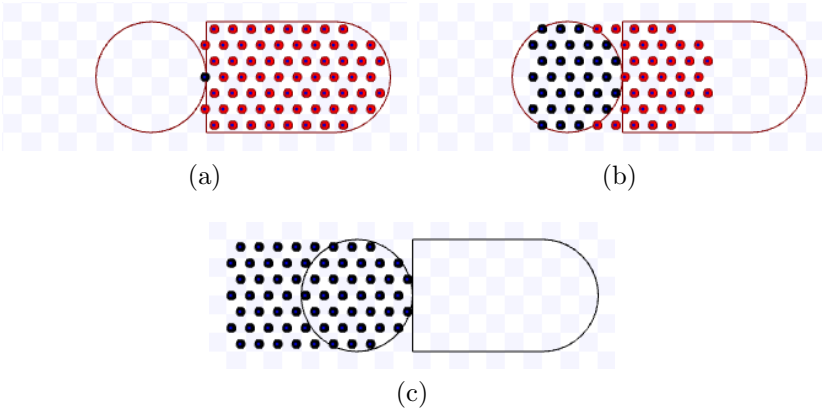


Figure 4.26: Simulation on Stage for hexagonal packing strategy using $s = 3$ m, $\theta = 0$ during $t = 9.8$ s. Available on `https://youtu.be/6_LgZWFOWd0`, accessed on 12 June 2022. (a) 0 s: beginning of the simulation; (b) After 4.9 s; (c) 9.8 s: ending of the simulation.

on (4.9). In addition, due to the floating-point error, in the computation of (4.10), instead of using $\min(L(x_h), C_2(x_h)) = \lfloor L(x_h) \rfloor$, $|\min(L(x_h), C_2(x_h)) - \lfloor L(x_h) \rfloor| < 0.001$ was checked.

Additionally, note in Figure 4.27 that for any value of s or θ, as time passes, the values of (4.9) asymptotically approach some value inside the bounds given by (4.11). Although the exact asymptotic value could not be given for the presented parameters, the experiments show that the bounds are correct. In the same manner, as occurred for parallel lanes, the higher throughput per time is reflected as a lower arrival time of the last robot per the number of robots, and it tends to infinity as the number of robots grows. These graphs showing the relation between the time of arrival at the target and the number of robots are in Section C.2.

4.6.4 Touch and Run

For the touch and run strategy, the robots maintain the linear speed over the whole experiment, then turn at a fixed constant rotational speed $\omega = v/r$, for r obtained from (4.13), when they are next to the target centre by the distance d_r obtained from (4.12). After they arrive at the target region, when they are distant from the target centre by d_r, they leave the curved path, stop turning and follow the linear exiting lane. On that lane, to stabilise their path following, the robots follow the queue using a turning speed equal to $\gamma - \beta$, such that β is the angle of the exit lane and γ is the robot orientation angle, both in relation to the x-axis.

The speed of these experiments was $v = 0.1$ m/s because the robots utilised on

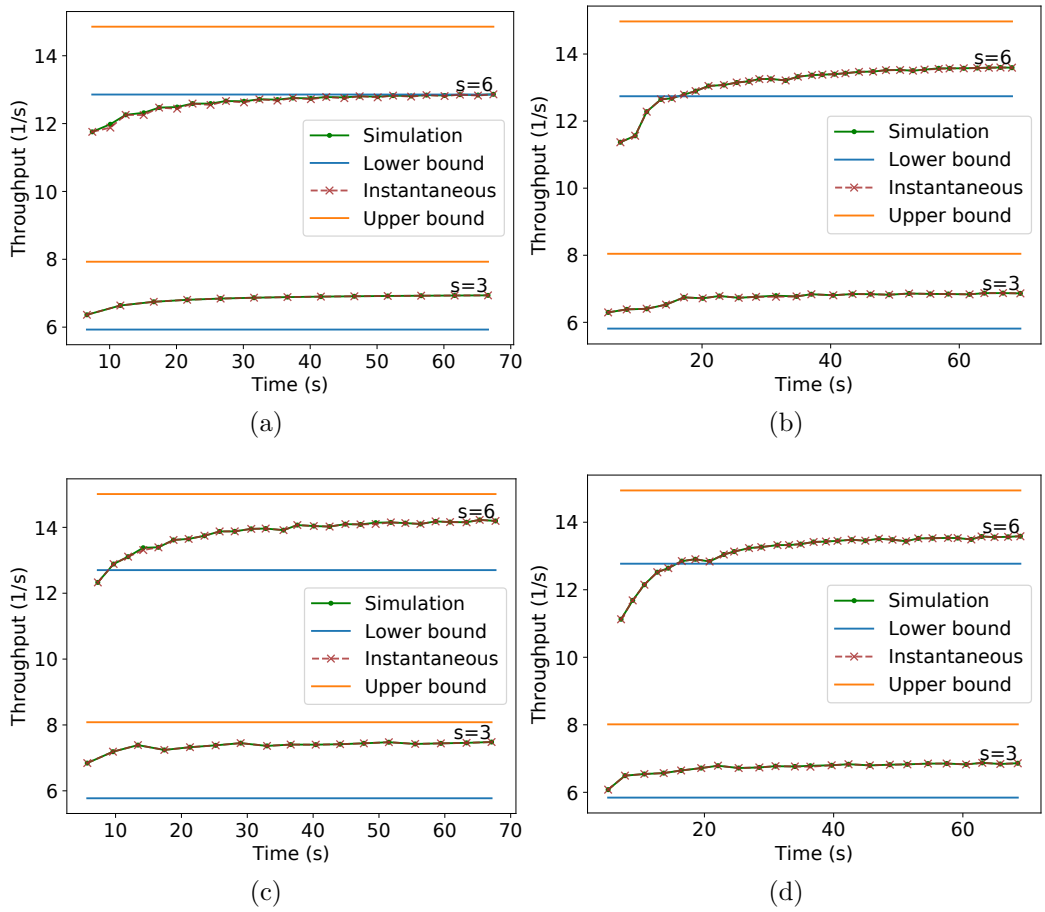


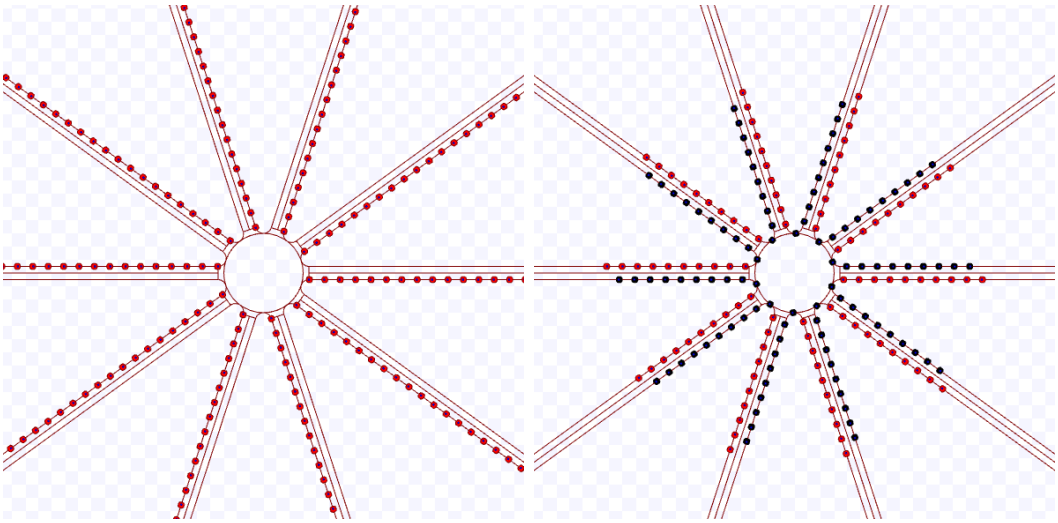
Figure 4.27: Comparison of simulation data with the asymptotic and instantaneous throughput for hexagonal packing with different values of s and θ. (a) $\theta = 0$; (b) $\theta = \pi/12$; (c) $\theta = \pi/6$; (d) $\theta = 5\pi/18$.

Stage have a maximum turning speed of $\pi/2$ rad/s. Choosing a low linear speed implies a greater number of lanes K, as the turning speed $\omega = v/r$ and r vary over K and s. In addition, a low linear speed diminishes the time measurement error, since the positions of the robots are sampled at every 0.1 s by the Stage simulator. Their positions are not guaranteed to be obtained at the exact moment they are far from the target centre by d_r; thus, this also yields an error in time measurement for their arrival in the target area.

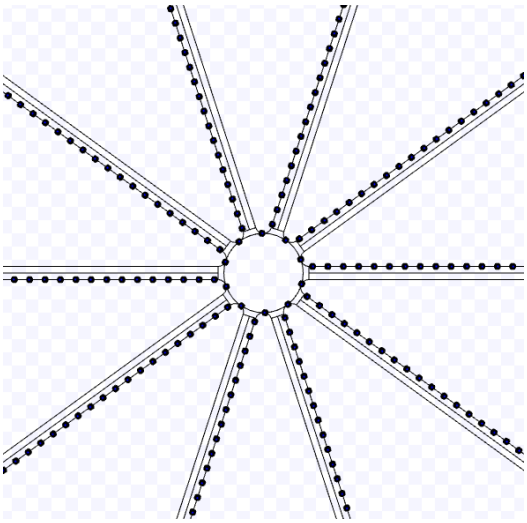
The value of s is in $\{3, 6\}$ m and all allowed K values are used for experimenting with the touch and run strategy with 200 robots. By (4.14), for the former s value, there is a maximum $K = 18$ and for the latter, $K = 37$. However, as the maximum angular speed is limited, the allowed K values range for $s = 3$ m is reduced to $\{3, \ldots, 16\}$ and for $s = 6$ m, $\{3, \ldots, 33\}$. Figures 4.28 and 4.29 present screenshots from executions using $s = 3$ m and different values of K. Other examples with $s = 6$ m are in Section C.1. The circle in the middle of these figures is the target region, and the lines where the robots are over represent the curved trajectory they follow by the touch and run strategy. Notice that as the number of lanes increases, the distance between the robots tends to be higher because, when the robot approaches the target area in the entering ray, it becomes more distant from the robots in the exiting ray. As v is constant, and they cannot change their speed, the distance between them may increase as the number of lanes grows, according to (4.17) and (4.18). Due to the throughput of the touch and run strategy in Proposition 8 being inversely proportional to that distance, the instantaneous and asymptotic throughput drop for some higher values of K.

Figure 4.30 presents the comparison of (4.16) and (4.19) for the throughput for a given time, the bound on its asymptotic value and the one obtained from Stage simulations. Although the total number of robots and the linear speed were fixed, the arrival times and the number of robots to reach the target change for each parameter used in this figure since the distance between the robots per lane varies and the number of robots simultaneously arriving is, in most cases, the number of lanes. In addition, the first two arrival times were not plotted because the first one is zero, yielding an indeterminate output by the throughput definition, and the second one is still too small in relation to the others, making the resultant throughput too high compared with the rest, thus producing an incomprehensible graph.

Observe that the values from (4.16) are almost equal to the simulation values, except for some points. They are different because of the floating-point error in the divisions and trigonometric functions before the use of floor function used on (4.16) – already mentioned in the introduction of Section 4.6 – as well as the time measurement errors for the arrival of the robots on the target area as explained at the beginning of this section. As expected, the values of (4.16) tend to come nearer to the asymptotic value given by (4.19). Differently from the previous strategies, notice that, for small

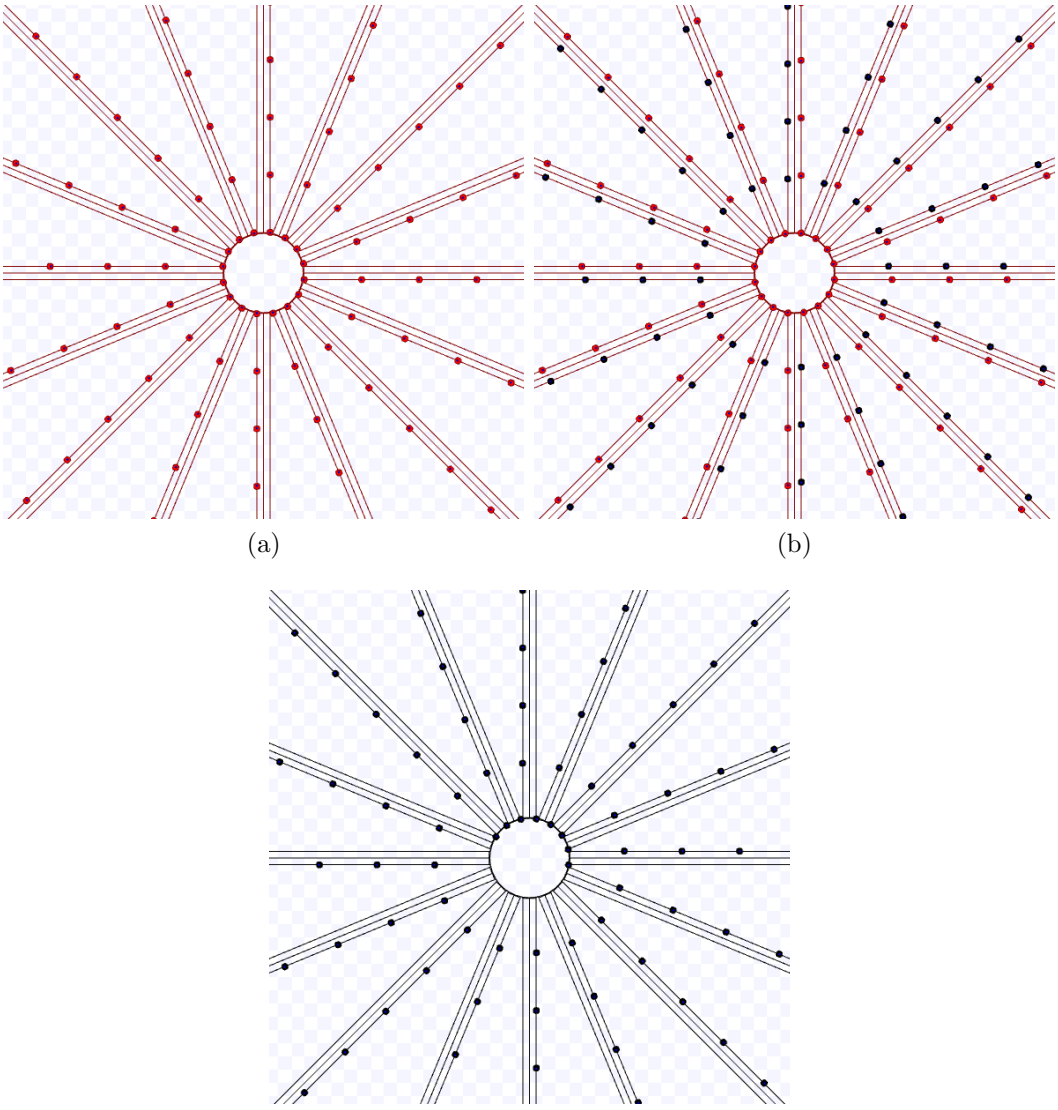


(a) (b)



(c)

Figure 4.28: Simulation on Stage for the touch and run strategy using $s = 3$ m, $K = 10$ during $t = 228$ s at $v = 0.1$ m/s. Available on `https://youtu.be/Z-ruOMYFyBU`, accessed on 12 June 2022. (a) 0 s: beginning of the simulation; (b) After 114 s; (c) 228 s: ending of the simulation.



(a) (b)

(c)

Figure 4.29: Simulation on Stage for the touch and run strategy using $s = 3$ m, $K = 16$ during $t = 523.1$ s at $v = 0.1$ m/s. Available on `https://youtu.be/FvAqv0zD4_Y`, accessed on 12 June 2022. (a) 0 s: beginning of the simulation; (b) After 261.6 s; (c) 523.1 s: ending of the simulation.

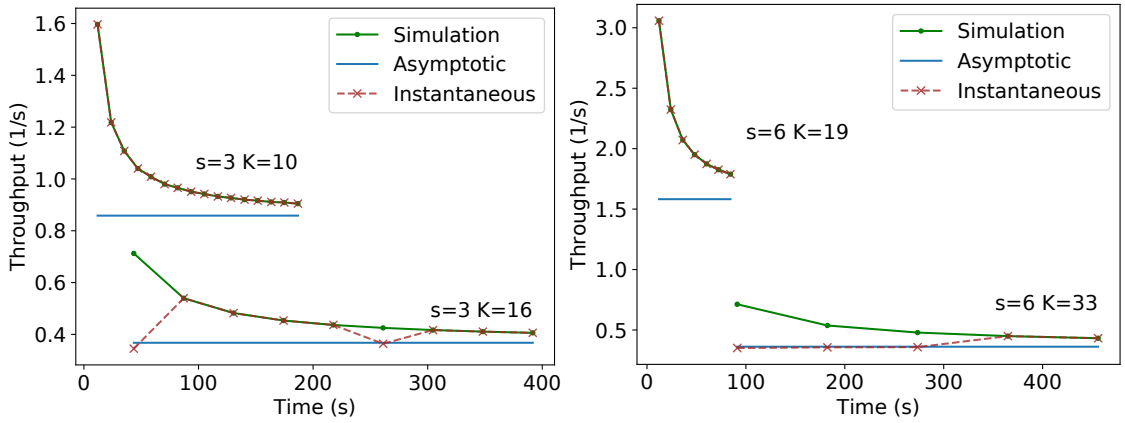


Figure 4.30: Throughput versus time comparison of the touch and run simulation on Stage with asymptotic values and the theoretical instantaneous equation for the throughput for different values of s and K. (a) $s = 3$ m and $K \in \{10, 16\}$; (b) $s = 6$ m and $K \in \{19, 33\}$.

values of t, the throughput is higher than for larger ones because, for a fixed K, (4.16) is decreasing for t. As occurred for the previous strategies, higher throughput per time is reflected as a lower arrival time of the last robot per number of robots, which tends to infinity as the number of robots grows. Figures displaying this relation between the time of arrival at the target and the number of robots are in Section C.2.

Figure 4.31 shows a comparison of the throughput at the end of the experiment – that is, for 200 robots and considering the difference between the time to reach the target region spent by the last robot and the first – and the asymptotic throughput obtained by (4.19) for all the possible number of lanes (K) for the used parameters and $s \in \{3, 6\}$ m. The simulation values tend to come close to the asymptotic value, confirming the theoretical results.

4.6.5 Comparison between Hexagonal Packing and Parallel Lanes

As discussed in Section 4.5.4, it is observed that the parallel lane strategy has a higher throughput than hexagonal packing for values of $u = s/d$ from 0.5 to a value of about 0.85 and for high values of t, despite the parallel lanes having lower asymptotic throughput for other values of u. In order to validate this observation, experiments on Stage were performed for these strategies using $t = 10{,}000$ s, $v = 0.1$ m/s, $d = 1$ m and s ranging from 0.4 to 0.95 m in increments of 0.05 m. The best hexagonal packing

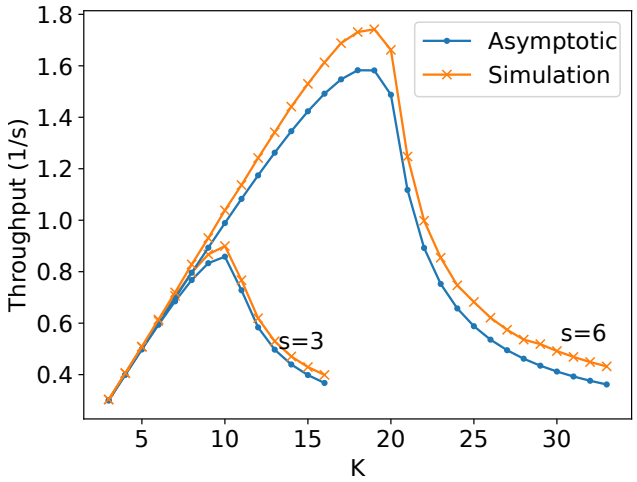
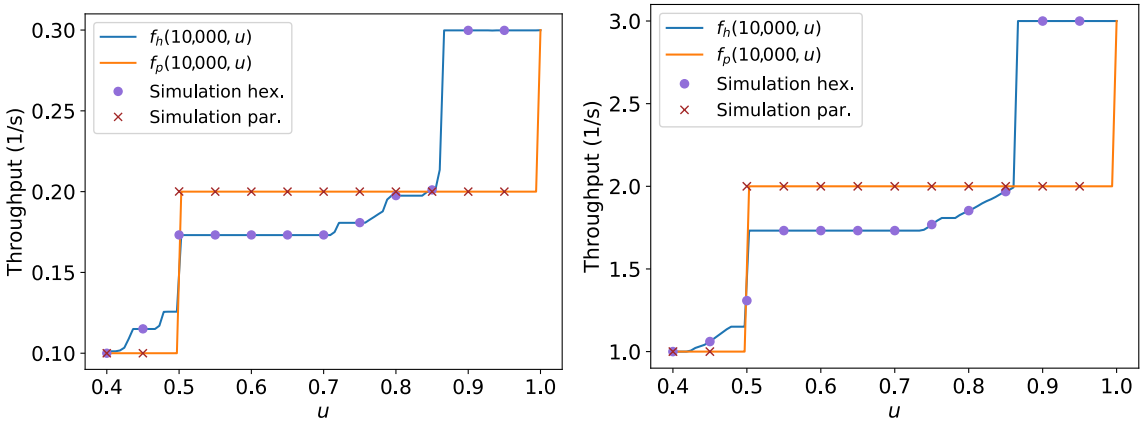


Figure 4.31: Throughput versus the number of lanes comparison of the simulation on Stage and asymptotic throughput for $s \in \{3, 6\}$ m.

angle θ was computed for hexagonal packing using the same method mentioned at the end of the theoretical section, i.e., the maximum throughput was searched using 1000 evenly spaced points between $[0, \pi/3)$ to find the best θ; then, it was compared with the result for $\pi/6$.

Figure 4.32 presents the results from the experiments with Stage and the theoretical results shown earlier. The functions f_h and f_p are the same presented in Figure 4.17. The labels "Simulation hex." and "Simulation par." stand for the throughput resultant from the experiments with hexagonal packing and parallel lanes strategies, respectively. The throughput improvement for the values of $u = s/d$ where the parallel lanes strategy overcomes the hexagonal packing is mainly caused by the square packing being more effective than hexagonal packing for fitting the robots inside the circle over the time for those values. To illustrate this, Figure 4.33 illustrates the execution for $v = 0.1$ m/s, $d = 1$ m and $s \in \{0.5, 0.85\}$ m. The robots run from right to left at a constant linear speed $v = 0.1$ m/s. The grey squares are highlighted – which measure 1×1 m^2 – to help estimate the time needed for about eight robots to arrive in the target region. This figure shows that the square packing fits more robots than hexagonal packing over time in these cases.

Observe in these figures that when the robots are arranged in squares, more robots arrive per unit of time than using hexagonal packing. To help visualise this, heed that in Figure 4.33a, there are $N = 9$ robots in black, occupying a rectangle including the circular target area with a width of approximately $W \approx 4.5$ m (this distance can be roughly measured by the grey squares, counting from the two last black robots on the right side to the first one in the left side). As $v = 0.1$ m/s was assumed, the



(a) (b)

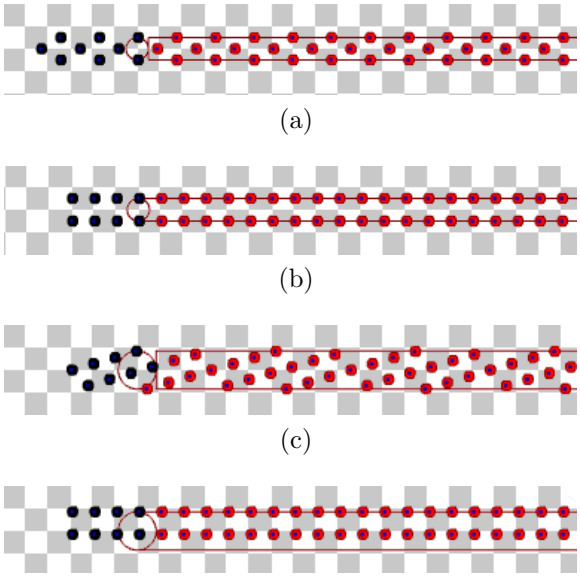
Figure 4.32: Throughput versus ratio $u = s/d$ comparing hexagonal packing and parallel lanes strategies for $v \in \{0.1, 1\}$ m/s, including results from Stage simulations. (a) $v = 0.1$ m/s; (b) $v = 1$ m/s.

throughput in this case is approximately $\frac{N-1}{\frac{W}{v}} \approx \frac{(9-1)0.1}{4.5} \approx 0.178\ \text{s}^{-1}$. Making similar calculations, Figure 4.33b–d have the approximate throughput $\frac{(8-1)0.1}{3} \approx 0.233\ \text{s}^{-1}$, $\frac{(8-1)0.1}{4} = 0.175\ \text{s}^{-1}$ and $\frac{(8-1)0.1}{3} \approx 0.233\ \text{s}^{-1}$, respectively. The results from the parallel lanes in this illustration – about 0.233 for both values of s – surpass the values for the hexagonal packing.

4.7 Summary

This chapter presented definitions of the common target area throughput (Definitions 1 and 2). The aim is to calculate the highest common target area throughput reached by different strategies by assuming that the target is a circle of radius s and the robots are moving at constant maximum linear speed v and constant minimum distance between each robot d. When the throughput is calculated for a given time t after the first robot arrives at the target area, call it the *instantaneous throughput* or the *throughput for a given time t*. As time passes, the throughput reaches a limit, when the time goes to infinity, defined as the *asymptotic throughput*.

In order to analyse the throughput for different strategies, the throughput of a common target point was calculated. Although a point is not reachable in practice, using a point-like target gives insights into the best angle between different straight-line trajectories of robots going to the same point (Lemma 1 and Proposition 1). After that, the throughput of a small target area is computed by using the *compact*



(a) (b) (c) (d)

Figure 4.33: Screenshots of the Stage simulation for hexagonal packing and parallel lanes strategy for $d = 1$ m and $s \in \{0.5, 0.85\}$ m. (a) Hexagonal packing with best θ for $s = 0.5$ m. Available on `https://youtu.be/IZBnFHLKXUA`, accessed on 12 June 2022. (b) Parallel lanes for $s = 0.5$ m. Available on `https://youtu.be/YYv1dJFkdPA`, accessed on 12 June 2022. (c) Hexagonal packing with best θ for $s = 0.85$ m. Available on `https://youtu.be/r9X0fsnngm0`, accessed on 12 June 2022. (d) Parallel lanes for $s = 0.85$ m. Available on `https://youtu.be/0cx-bHPIong`, accessed on 12 June 2022.

lanes, where the robots go in two lanes, and the robots are distant by themselves in a triangular pattern (Propositions 2 and 3).

These results are utilised for developing three strategies for large target areas. These strategies are divided into two groups. The first group is the corridor strategies, formed by the two first strategies shown for large areas, where the robots go through a corridor of lanes towards the target area. The second group is the multiple curved trajectories around the target area and is composed of the third strategy presented for large target areas.

The first strategy is the *parallel lanes*, where the robots go to the target area in parallel lanes distanced by d in a square pattern. For that strategy, Proposition 4 shows its instantaneous and asymptotic throughput.

The second strategy is the *hexagonal packing*, where the robots go in parallel lanes towards the target area, but their distance is less than d and the robots are in a hexagonal pattern. The upper bound of the asymptotic throughput of the hexagonal packing is calculated in Proposition 5. Nevertheless, this proposition relies on results that do not tell how the hexagons representing the position of the robots are placed inside the lanes. For this strategy, the hexagon pattern may be rotated at any angle in relation to the horizontal axis (denoted *hexagon packing angle* in Definition 3). Depending on this angle, the instantaneous throughput can be increased. Yet, obtaining the hexagonal packing angle that maximises the instantaneous throughput from the parameters s, v, d and t is not straightforward. Because of that, although Proposition 6 gives an exact instantaneous throughput for this strategy, it can only show the lower and upper bounds of the asymptotic throughput instead of an exact value.

The third strategy for large target areas is the *touch and run*. In it, the robots touch the border of the circular target area through curved lanes tangent to that border. These lanes have an entrance and leaving straight paths and are inside circular sector regions which form an angle of α with the target area centre (called *central region angle*). Lemma 2 calculates the distance from the target area that the robots start turning. Lemma 3 relates α, d and the turning radius of the curve tangent to the target area. Proposition 7 gives the possible number of lanes around the target area from the parameters s and d. Proposition 8 computes the instantaneous and asymptotic throughput of the touch and run strategy.

For all the strategies above, the ratio $\frac{s}{d}$ is in their expression of instantaneous and asymptotic throughput. Thus, a comparison is made in relation to that ratio. Proposition 9 shows that below a specific ratio value, the parallel lanes strategy has a higher asymptotic throughput than the lower bound of the hexagonal packing asymptotic throughput. That means the parallel lanes may be better for a hexagonal packing angle and these values of $\frac{s}{d}$. However, Proposition 9 also states that, for ratios no less than that specific value, the hexagonal packing is better than the parallel lanes.

However, the asymptotic throughput of the touch and run strategy is greater than the maximum possible asymptotic throughput of the hexagonal packing strategy for almost all values of $\frac{s}{d}$, except for some in the interval $(1.12, 1.25)$. In summary, depending on the value of this ratio, one of those strategies is the most suitable.

The results above were tested and corroborated in experiments described in Section 4.6. Some of them are used in the next chapter in the context of the SQF and TRVF algorithms, which are inspired by the corridor strategies (parallel lanes and hexagonal packing) and the touch and run strategy, respectively. The TRVF algorithm uses the same lanes of the touch and run strategy. Thus, the calculation in Lemma 2 is used in that algorithm, and the maximum number of lanes for this algorithm is obtained from Proposition 7. The lower bound of the asymptotic throughput in Proposition 6 for the hexagonal packing and the asymptotic throughput of the touch and run strategy in Proposition 8 are compared with the results obtained from experiments with SQF and TRVF algorithm in Sections 5.4.1.1 and 5.4.1.2.

Chapter 5

Proposed algorithms

This chapter presents algorithms based on the results in the previous one: the Single Queue Former (SQF) in Section 5.1 and the Touch and Run Vector Fields (TRVF) in Section 5.2. After these algorithms, algorithms for the ad hoc agents are presented in Section 5.3. Consider the scenario where a large number of robots must reach a common target. Each robot moves towards another destination after reaching the target, which may be shared among the robots. Assume that the target is a circular area of radius s. A robot reaches the target if its centre of mass is distant less or equal to the radius s from the target centre. Consider that there is no minimum time to stay at the target. The algorithms are only executed inside a circle of radius $D > s$ around the target centre (denoted the *working area*). Also, assume that if the robots have two or more targets, their distance is at least $2D$.

Differently from Chapter 4, which intended to find the maximum throughput for the strategies by using constant maximum linear speed and fixed minimum distance between the robots, in real applications, robots move at variable speeds, and they do not always keep the same distance between others, although they may try to maintain some distance to avoid bumping. In addition to that, the space around the target region is finite and may have other obstacles.

Inspired by these theoretical developments presented in Chapter 4, new algorithms are proposed in this chapter. That is, it discusses the analysis of the arrival rate in the target area as the time tends to infinity as an alternative approach to analysing congestion in swarms with a common target area. This measure has the advantage of being finite for any closed target region of any shape as the number of robots grows, as opposed to the number of robots per time of arrival at the target area. Assuming a circular target area and robots with constant maximum linear speed and fixed minimum distance from each other, theoretical strategies are developed for entering that area and calculated their theoretical throughput for a given time and their asymptotic throughput. The presented theoretical strategies were based

on forming a corridor towards the target area or making multiple curved trajectories towards the boundary of the target area. Based on the corridor strategy, the SQF algorithm is proposed in Section 5.1, and, inspired by the multiple curved trajectories, the TRVF algorithm is introduced in Section 5.2.

The Ad Hoc Follower (AHF) is an attempt to resolve the common target problem for ad hoc teamwork discussed in Chapter 1 without using learning. Shortly speaking, in this attempt, ad hoc robots follow a fixed alternative algorithm if they do not perceive another robot in their neighbourhood; otherwise, they go after the other robot while the angle between the vector pointed by the alternative algorithm and the direction vector of the other robot does not exceed a predetermined angle. However, as shown in Section 5.4.1.3, if the robots do not follow any robot, the task completion time is minimised.

Thus, our proposed approach is to execute a fixed alternative algorithm without following any robot. In this way, two teams will be running in the same space using different algorithms: one running the ad hoc robots' algorithm and the other, the aware robots' algorithm. Such approach uses mixed teams, and there are many situations where different swarm teams may be present in the same environment. For instance, in the multi-agent literature, we can find such mixed teams (MT) (Beavers and Hexmoor, 2001) in problems like Pursuit-Evasion Games (dos Santos et al., 2023; Salimi et al., 2016) and Multi-Agent Pickup and Delivery (Flammini, Azzalini, and Amigoni, 2023). Therefore, in this thesis, when robots use AHF without following any robot, they are in a Mixed Teams (MT).

Although not using the theoretical results in Chapter 4, AHF and MT described in Section 5.3 have another algorithm as a parameter. Thus, SQF and TRVF are described before it, aiding the description of the experiments with AHF and MT parameterised by them in Section 5.4.

5.1 Single Queue Former

Based on the corridor strategy from Chapter 4, an algorithm is proposed, named Single Queue Former, to improve throughput in case of target congestion whose aim is to form a single queue that goes towards the target. The queue is formed inside a rectangular corridor of width equal to the circular target diameter and a given fixed length. The robots should only enter the target region by this queue.

Specifically, this algorithm is based on the compact lanes and hexagonal packing strategies findings from Section 4.5.1 and 4.5.2. However, the hexagonal formation is not enforced here to avoid robots losing time forming it, although the hexagon packing is the configuration of the robots that maximises the throughput in an ideal scenario where the robots maintain the same distance and keep constant linear speed. As it will be shown in Section 5.4, the throughput for this algorithm is bounded by the

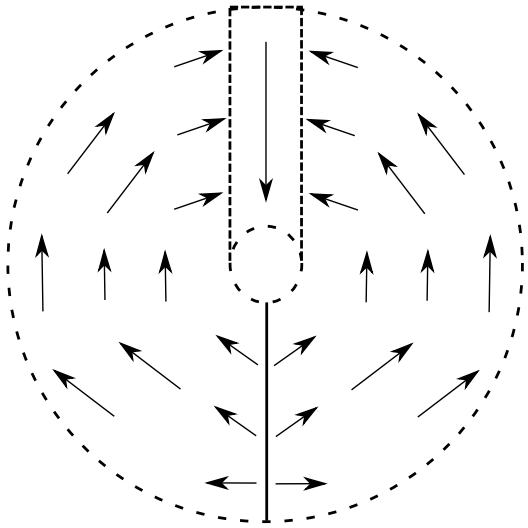


Figure 5.1: Rotational force field to reach the target with Single Queue Former algorithm.

hexagonal packing throughput for the mean distance and mean linear speed achieved by the robots.

Artificial potential fields (Siegwart, Nourbakhsh, and Scaramuzza, 2011) are used to apply forces on the robots to form the queue and exit the target area efficiently. Let s be the radius of the current target. The corridor queue has a width $2s$ and starts from the current target centre, denoted by $\mathbf{G} = (G_x, G_y)$. Let the length of the corridor be the same value used for the radius of the circular working area of the algorithms around the target, denoted D. Without any loss of generality, the corridor is located between the point at (G_x, G_y) and the point at $(G_x, G_y + D)$. Robots that are going to enter the target area are submitted to a rotational field whose centre is the position of the target centre. When they reach the corridor, they are submitted to an attractive force towards the target. An illustration is shown in Figure 5.1.

Therefore, robots outside the corridor are submitted to a force according to the following equations. For the right-hand side of the circle, an anti-clockwise rotational field:

$$F_x = -K_{SQF}\frac{p_y - G_y}{\|\mathbf{p} - \mathbf{G}\|}, \quad F_y = +K_{SQF}\frac{p_x - G_x}{\|\mathbf{p} - \mathbf{G}\|}. \tag{5.1}$$

For the left-hand side of the circle, a clockwise rotational field:

$$F_x = +K_{SQF}\frac{p_y - G_y}{\|\mathbf{p} - \mathbf{G}\|}, \quad F_y = -K_{SQF}\frac{p_x - G_x}{\|\mathbf{p} - \mathbf{G}\|}, \tag{5.2}$$

where F_x and F_y are the two components of the force applied, $\mathbf{p} = (p_x, p_y)$ is the

position of the robot, $\mathbf{G} = (G_x, G_y)$ is the target centre, and K_{SQF} is a constant for setting the force magnitude.

Additionally, it was noticed in previous algorithms (Marcolino et al., 2017) that the robots reaching the target tend to stop on the target and orient themselves towards their next goal. This increases the time during which the target is occupied and thus decreases the throughput. To solve this problem, robots exiting nearby the target are constrained by another rotational field. This field also is only applied inside a circle with radius D around the target (Figure 5.2). The aim of this field is to be aligned with the corridor orientation at the target and progressively allow robots to change directions. This field also depends on the new target position, not only on the robot location, because if some robots reach the target area on a side but have to go to the opposite side when they leave the circle, it will cause congestion next to the exit area. For a robot with a new target located on the right-hand side of the previous target centre, an anti-clockwise rotational field is applied:

$$F_x = -K_{SQF}\frac{p_y - Q_y}{\|\mathbf{p} - \mathbf{Q}\|}, \quad F_y = +K_{SQF}\frac{p_x - Q_x}{\|\mathbf{p} - \mathbf{Q}\|}. \tag{5.3}$$

For a robot with a new target located on the left-hand side of the previous target centre, a clockwise rotational field is applied:

$$F_x = +K_{SQF}\frac{p_y - P_y}{\|\mathbf{p} - \mathbf{P}\|}, \quad F_y = -K_{SQF}\frac{p_x - P_x}{\|\mathbf{p} - \mathbf{P}\|}, \tag{5.4}$$

where $\mathbf{Q} = (Q_x, Q_y) = (G_x + D, G_y)$ is a point on the right of the target, and $\mathbf{P} = (P_x, P_y) = (G_x - D, G_y)$ is a point on the left of the target.

This outgoing rotational field enables robots to stay on their course after they reach the target and gradually rotate towards their next goal. The orientation of the field depends on the position of the robot relative to the corridor. If the robot is on the left of the corridor, the robot follows the rotational field going to the left with centre $\mathbf{P}$. Otherwise, if it is on the right of the corridor, it follows the rotational field going to the right with centre $\mathbf{Q}$.

Finally, let the influence radius be the maximum distance a robot considers anything sensed as an obstacle to avoid from its mass centre. To enable the robots to deal with small target sizes, two different constants are used for the influence radius between the robots when calculating repulsive forces, I_d and I_{min}, with $I_{min} < I_d$: I_d is the default and maximum radius for influence, and I_{min} is the minimum allowed.

For robots inside the corridor or exiting the target region, the influence radius is set to I_{min}. Now, consider a robot is outside the corridor, but above the $\overline{\mathbf{PQ}}$ line in Figure 5.2. Let d' be the distance between the robot and the central line of the corridor (that is, from the position of the robot to the closest point in the vertical axis of Figure 5.2, starting from the target centre). Its influence radius I varies in

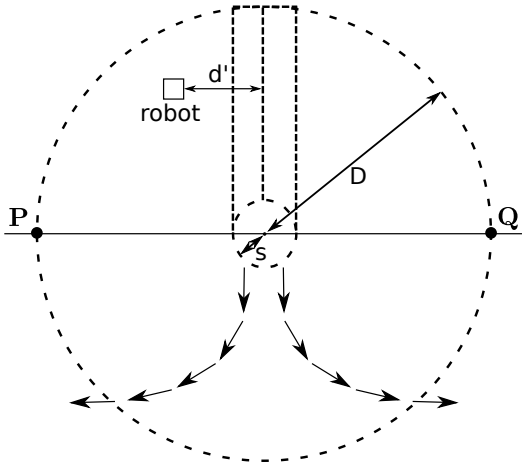


Figure 5.2: Rotational force field to leave the target in the SQF algorithm, and the distance d' from the central line of the corridor to the robot for varying its influence radius.

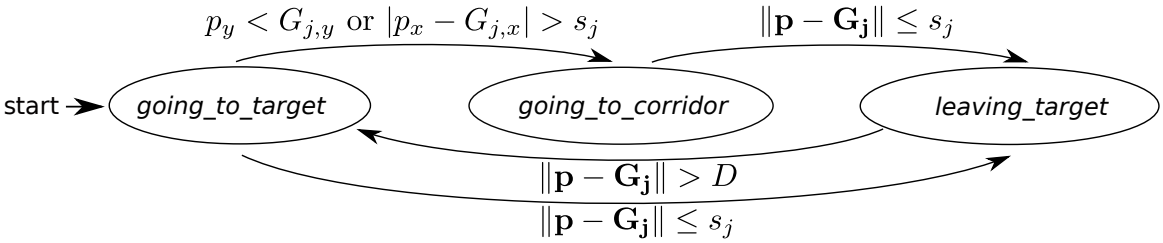


Figure 5.3: State machine transitions for the SQF algorithm.

relation to d' and is set to $I = I_{min} + d'$ only for $0 \leq d' \leq I_d - I_{min}$. This range for d' guarantees that $I_{min} \leq I \leq I_d$. For the rest of the robots the influence radius is I_d.

Algorithm 1 presents the SQF algorithm. Robots have the states *going_to_target*, *leaving_target* and *going_to_corridor*, which respectively means the robot is going straightly to the target region, it is leaving it, and it is going to the corridor. The robots begin at *going_to_target* state. Figure 5.3 shows the state machine and its transitions, assuming that the robot may have two or more targets located at $\mathbf{G}_j = (G_{j,x}, G_{j,y})$, the circular target region has radius s_j, and j is its current target index. Note that this algorithm is only executed when the robot is at a distance at most D from the target centre. When there are many robots inside the working area, due to the repulsive forces, the robots next to the target area that did not reach the corridor may be pushed towards the target area before they access the corridor. Thus, when this happens, they are allowed to go directly from *going_to_target* to *leaving_target* to avoid more congestion and optimise traffic in this situation.

Input : K_{SQF}: force magnitude;
D: the length of the corridor;
$\mathbf{G}_1 = (G_{1,x}, G_{1,y}), \ldots, \mathbf{G}_n = (G_{n,x}, G_{n,y})$: a list of $n \geq 2$ circular target region centres;
$s_1, \ldots, s_n$: a list of n circular target region radii;
j: the current target index;
I_{min}, I_d: minimum and default influence radius of the robot.
Output: $\mathbf{F} = (F_x, F_y)$: force vector;
I: influence of the robot for repulsive force calculation.

Get position of the robot $\mathbf{p} = (p_x, p_y)$, and let $\mathbf{G} = \mathbf{G}_j = (G_x, G_y)$ and $s = s_j$;
if $\|\mathbf{p} - \mathbf{G}\| \leq s$ **then**
  state $\leftarrow$ *leaving_target*;
  Increment j, then let $\mathbf{G} = \mathbf{G}_j = (G_x, G_y)$ and $s = s_j$;
if *state* $\neq$ *leaving_target* **then**
  if $\|\mathbf{p} - \mathbf{G}\| \leq D$ *and* ($p_y < G_y$ *or* $|p_x - G_x| > s$) **then**
    state $\leftarrow$ *going_to_corridor*;
    Set $\mathbf{F}$ according to (5.1) and (5.2);
  else
    state $\leftarrow$ *going_to_target*;
    $\mathbf{F} \leftarrow K_{SQF} \frac{\mathbf{G}-\mathbf{p}}{\|\mathbf{G}-\mathbf{p}\|}$;
else
  if $\|\mathbf{p} - \mathbf{G}_{j-1}\| \leq D$ **then**
    Set $\mathbf{F}$ according to (5.3) and (5.4);
  else
    state $\leftarrow$ *going_to_target*;
    $\mathbf{F} \leftarrow K_{SQF} \frac{\mathbf{G}-\mathbf{p}}{\|\mathbf{G}-\mathbf{p}\|}$;
if *state = going_to_target or state = leaving_target* **then**
  $I \leftarrow I_{min}$;
else if *state = going_to_corridor and* $p_y > G_y$ *and* $|p_x - G_x| < I_d - I_{min}$ **then**
  $I \leftarrow I_{min} + |p_x - G_x|$;
else
  $I \leftarrow I_d$;
return $(\mathbf{F}, I)$;

Algorithm 1: SQF algorithm, executed at every update in the position of the robot.

5.2 Touch and Run Vector Fields

Based on the results of the touch and run strategy in Section 4.5.3, an algorithm is devised to mimic the curved trajectories around the target region using vector fields, named Touch and Run Vector Fields (TRVF). The vector fields are only used inside a circle with radius D around the target centre and apply forces on the robot to make it follow a circular trajectory similar to the one described in the touch and run strategy.

As a way to do that, the vector fields proposed by Nelson et al. (2006) for straight line and orbit following trajectories are adapted and combined. Their straight line following vector field are employed whenever the robot has to follow a straight trajectory and an anti-clockwise orbit following vector field for curved trajectories. Firstly, our modified functions for these trajectories are explained, and afterwards, the TRVF algorithm is introduced.

Our adaptation of their work concerns reducing the orbit following to a circular sector, instead of the full circle, and adding an attractive force towards a fixed point when the robots are performing the orbit following next to the target region. The reduction to a circular sector is due to the curved trajectory next to the target having $\alpha < 2\pi$ as explained in the Section 4.5.3. Attractive forces towards a fixed point are added to the orbit following vector field intending to avoid the repulsive forces caused by nearby robots to push a robot far away from that fixed point.

Figure 5.4 shows the straight line and orbit following vector fields proposed by Nelson et al. (2006). Algorithm 2 presents our adapted function that computes the straight line following vector field, its parameters and values returned. It assumes a fixed segment to follow with initial and final waypoints $\mathbf{w_i}$ and $\mathbf{w_f}$, respectively. Based on the current position and orientation of the robot, $\mathbf{p}$ and ξ, respectively, a force vector with magnitude K_{TRVF} is computed. The orientation of the force vector changes gradually as the robot approaches the segment. These changes are calculated using the maximum linear speed of the robot v, and its rate of change depends on the constants k_s and k_ω.

In the original algorithm, ξ_e is the entry heading angle, and τ is the distance perpendicular to the line from which the vector field makes the heading angle begin to change (in Figure 5.4a, it is the distance from the dashed line to the solid line). In our adaptation, $\tau = I_d/5$ and $\xi_e = \pi/2$, for a given default influence radius of the robot I_d. Additionally, the force vector is only computed if the robot did not reach the point $\mathbf{w_f}$ or if the robot is in a position on the plane perpendicular to $\overline{\mathbf{w_i}\mathbf{w_f}}$ limited by its endpoints, otherwise, force vector zero is returned (line 4). At line 7, *sign* returns -1 for negative numbers and 1, otherwise. Here $\times$ stands for the cross-product of two vectors. A verification was also added for not a number (NaN) in the exponentiations and replace the result for zero as it is a common source of numerical errors (lines 14 and 15). It is assumed a proportional angular speed controller such that if the robot

1 `Function` straightPathFollowing $(K_{TRVF}, \mathbf{p}, \xi, \mathbf{w_i}, \mathbf{w_f}, I_d, k_s, k_\omega, v)$
Input : K_{TRVF}: force magnitude;
$\mathbf{p} = (p_x, p_y), \xi$: current position and orientation of the robot;
$\mathbf{w_i} = (w_{i,x}, w_{i,y}), \mathbf{w_f} = (w_{f,x}, w_{f,y})$: initial and final waypoints for the segment to follow;
I_d: influence radius of the robot;
$k_s > 1$: constant for exponentiation in the vector field calculation;
k_ω: constant for proportional angular speed controller;
v: maximum linear speed.
Output: $\mathbf{F} = (F_x, F_y)$: force vector;
t': indicates the position along the path (0 when the position of the robot is on the line perpendicular to $\overline{\mathbf{w_i}\mathbf{w_f}}$ on the initial waypoint and 1 when that happens for the final waypoint).
2 $\mathbf{w_{fi}} = \mathbf{w_f} - \mathbf{w_i}$;
3 $t' \leftarrow \frac{(\mathbf{p}-\mathbf{w_i})\cdot\mathbf{w_{fi}}}{\|\mathbf{w_{fi}}\|^2}$;
4 **if** $t' \geq 1$ **then return** $((0,0), t')$;
5 $\xi_f \leftarrow \text{atan2}(w_{fi,y}, w_{fi,x})$;
6 $\epsilon \leftarrow \|\mathbf{p} - \mathbf{w_i} - t'\mathbf{w_{fi}}\|$;
7 $\rho \leftarrow sign(\mathbf{w_{fi}} \times (\mathbf{p} - \mathbf{w_i}))$;
8 $\tau \leftarrow I_d/5$;
9 $\xi_e \leftarrow \pi/2$;
10 **if** $\epsilon > \tau$ **then**
11 $\quad \xi_c \leftarrow \xi_f - \rho\xi_e$;
12 **else**
13 $\quad \epsilon \leftarrow \rho\epsilon$;
14 $\quad P_1 \leftarrow \left(\frac{\epsilon}{\tau}\right)^{k_s}$; **if** P_1 *is NaN* **then** $P_1 \leftarrow 0$;
15 $\quad P_2 \leftarrow \epsilon^{k_s-1}$; **if** P_2 *is NaN* **then** $P_2 \leftarrow 0$;
16 $\quad \xi_c \leftarrow \xi_f - \xi_e P_1 - \left(\frac{k_s \xi_e v}{k_\omega \tau^{k_s}}\right) P_2 \sin(\xi)$;
17 $\mathbf{F} \leftarrow K_{TRVF}(\cos(\xi_c), \sin(\xi_c))$;
18 **return** $(\mathbf{F}, t')$;

Algorithm 2: Straight-line vector field algorithm adapted from Nelson et al. (2006).

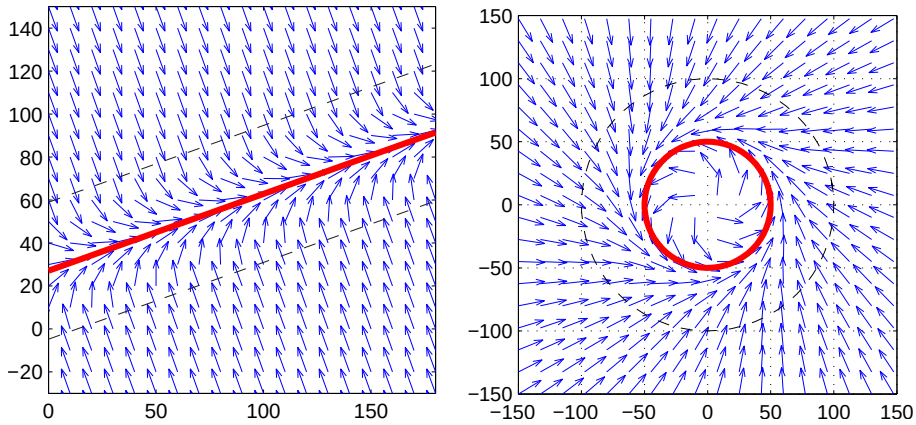


Figure 5.4: Straight line and orbit following vector fields (Nelson et al., 2006).

has orientation ξ and needs to turn to orientation given by ξ_c, then it uses angular speed $k_\omega(\xi_c - \xi)$ for a chosen $k_\omega > 0$.

Algorithm 3 shows our modified function that computes the orbit following vector field, its parameters and values returned. Let the orbit to follow be centred at a point $\mathbf{c} = (c_x, c_y)$ and have radius R. As in the straight line following algorithm, given the current position and orientation of the robot, $\mathbf{p}$ and ξ, respectively, a force vector with magnitude K_{TRVF} is calculated. In this work, the orbit following algorithm is intended to perform curves, so a waypoint $\mathbf{w_f}$ must be given to indicate where the orbit following must stop. As in the previous algorithm, the orientation of the force vector changes gradually as the robot approaches the orbit, and these changes are computed using the maximum linear speed of the robot v, and its rate of change depends on the constants k_o and k_ω.

This algorithm only generates a force vector different from zero if the robot did not cross $\overrightarrow{\mathbf{cw_f}}$ for a given waypoint $\mathbf{w_f}$. It is verified if it crossed by the positive sign of the cross product between the vector $\mathbf{q} = \mathbf{p} - \mathbf{c}$ (that is, the vector from the centre of the orbit to the position $\mathbf{p}$ of the robot) and the vector $\mathbf{w_f} - \mathbf{c}$. As the orbit is anti-clockwise, the orbit following vector field makes the cross product $\mathbf{q} \times (\mathbf{w_f} - \mathbf{c})$ decrease with time, being zero when $\mathbf{q}$ has the same orientation as $\mathbf{w_f} - \mathbf{c}$ and negative when the orientation of $\mathbf{q}$ is greater than the orientation of $\mathbf{w_f} - \mathbf{c}$. Force vector zero is returned when the robot is in a position beyond $\overrightarrow{\mathbf{cw_f}}$ or intersects this ray (line 4).

At line 5, γ is the heading from the centre of the orbit to the position of the robot. Nelson et al. (2006) use clockwise orientation and they assume null orientation when the robot is at the y-axis, thus $\gamma = \text{atan2}(q_x, q_y)$ is being used instead of $\text{atan2}(q_y, q_x)$ as in the original algorithm. Due to this, at line 12 $\pi/2 - \xi_c$ is utilised for computing

`Function` orbitPathFollowing $(K_{TRVF}, \mathbf{c}, R, \mathbf{p}, \xi, \mathbf{w_f}, k_o, k_\omega, v)$
Input : K_{TRVF}: force magnitude;
$\mathbf{c}, R$: centre and radius for orbit following;
$\mathbf{p} = (p_x, p_y), \xi$: current position and orientation of the robot;
$\mathbf{w_f} = (w_{f,x}, w_{f,y})$: final waypoint;
$k_o > 1$: constant for exponentiation in the vector field calculation;
k_ω: constant for proportional angular speed controller;
v: maximum linear speed.
Output: $\mathbf{F} = (F_x, F_y)$: force vector;
t': indicates the position along the path (positive when the robot did not cross $\overrightarrow{c\mathbf{w_f}}$).

$\mathbf{q} \leftarrow \mathbf{p} - \mathbf{c}$;
$t' \leftarrow \mathbf{q} \times (\mathbf{w_f} - \mathbf{c})$;
if $t' \leq 0$ **then return** $((0,0), t')$;
$\gamma \leftarrow \text{atan2}(q_x, q_y)$;
if $\|\mathbf{q}\| > 2R$ **then**
  $\xi_c \leftarrow \gamma - \frac{5\pi}{6} + \frac{v}{\|\mathbf{q}\|}\sin(\xi - \gamma)$;
else
  $P_1 \leftarrow \left(\frac{\|\mathbf{q}\| - R}{R}\right)^{k_o}$; **if** P_1 *is NaN* **then** $P_1 \leftarrow 0$;
  $P_2 \leftarrow (\|\mathbf{q}\| - R)^{k_o - 1}$; **if** P_2 *is NaN* **then** $P_2 \leftarrow 0$;
  $\xi_c \leftarrow \gamma - \frac{\pi}{2} - \frac{\pi}{3}P_1 - \frac{v}{k_\omega\|\mathbf{q}\|}\sin(\xi - \gamma) - \frac{k_o v \pi}{3R^{k_o}k_\omega}P_2\cos(\xi - \gamma)$;
$\mathbf{F} \leftarrow K_{TRVF}(\cos(\pi/2 - \xi_c), \sin(\pi/2 - \xi_c))$;
return $(\mathbf{F}, t')$;

Algorithm 3: Anti-clockwise orbit following vector field algorithm adapted from Nelson et al. (2006).

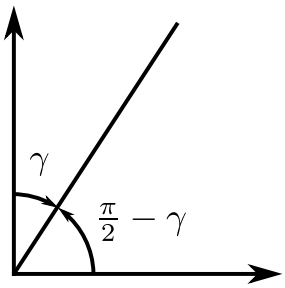


Figure 5.5: If a line has angle γ with the y-axis assuming clockwise orientation, then this same line has angle $\pi/2 - \gamma$ with the x-axis using anti-clockwise orientation, as the y-axis makes an angle of $\pi/2$ with the x-axis.

the result using the usual orientation, that is, anti-clockwise with null orientation at the x-axis, as justified by Figure 5.5.

Now the previously defined functions are used to introduce the TRVF algorithm. As explained in the theoretical section, the touch and run strategy uses K lanes. Each robot calculates four waypoints to follow in the lane located on the right side of the robot, assuming it is facing the target located at point $\mathbf{G}$. Figure 5.6 shows these four waypoints, which are based on the position of the robot $\mathbf{p}$, and must be followed in order. The four waypoints are calculated depending on the sector where they are. There are K sectors, one per lane, numbered from 1 to K. Let η be the angle of the vector $\mathbf{p} - \mathbf{G}$. Thus, a sector z of a robot is calculated by $z = \left\lfloor \frac{\eta}{\alpha} \right\rfloor + 1$ for the angle α defined in Section 4.5.3 and given by (4.15). Also in that section, the entering and exiting rays are defined – which robots use to enter and to exit in the curved trajectory towards the circular target area, respectively – presented in Figure 4.12a.

Let $\mathbf{E_1}$ be the point in the entering ray on the boundary of the circle centred at $\mathbf{G}$ with radius D, $\mathbf{E_2}$ be the point in the entering ray perpendicular to the point where the robot starts to turn in the touch and run strategy (as presented in Figure 4.12b), $\mathbf{E_3}$ be the point in the exiting ray perpendicular to the point where the robot ends the turn in the touch and run strategy, and $\mathbf{E_4}$ be the point in the exiting ray on the boundary of the circle centred at $\mathbf{G}$ with radius D. In Figure 5.6, these points are exemplified for sector 1. For a given sector z, the entering and exiting rays will have angle $z\alpha$ and $(z-1)\alpha$ with the x-axis, respectively. The position vector $\mathbf{w_1}$ is given by the sum of the vector $\mathbf{G}$, the vector going from $\mathbf{G}$ to $\mathbf{E_1}$ and the vector of modulus $I_d/2$ from $\mathbf{E_1}$ rotated $\pi/2$ anti-clockwise from the entering ray. Thus,

$$\begin{aligned} \mathbf{w_1} &= \mathbf{G} + (\mathbf{E_1} - \mathbf{G}) + \frac{I_d}{2}(\cos(z\alpha - \pi/2), \sin(z\alpha - \pi/2)) \\ &= \mathbf{G} + D(\cos(z\alpha), \sin(z\alpha)) + \frac{I_d}{2}(\sin(z\alpha), -\cos(z\alpha)). \end{aligned} \tag{5.5}$$

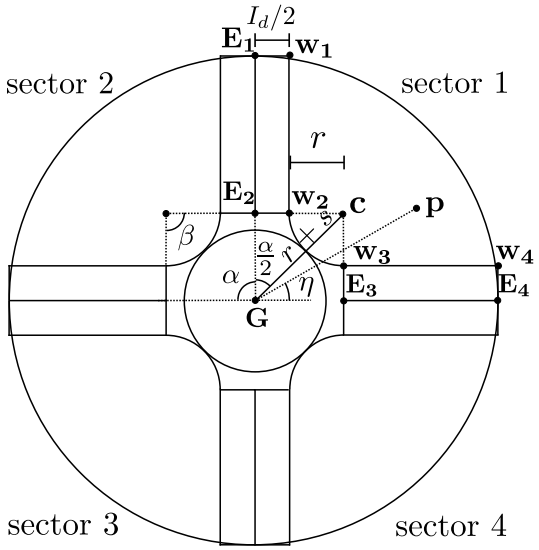


Figure 5.6: The robots follow the waypoints $\mathbf{w_1}$, $\mathbf{w_2}$, $\mathbf{w_3}$ and $\mathbf{w_4}$ depending on the sector where they are located. The centre $\mathbf{c}$ for the curved trajectory between waypoints $\mathbf{w_2}$ and $\mathbf{w_3}$ is distant by $r + s$ from the target centre $\mathbf{G}$. In this example, $K = 4$ and a robot is at position $\mathbf{p}$ and at sector 1.

The vector $\mathbf{w_2}$ is given similarly to $\mathbf{w_1}$ but using $\mathbf{E_2}$. The distance from $\mathbf{G}$ to $\mathbf{E_2}$ is $\sqrt{(r+s)^2 - (r + I_d/2)^2}$. This was calculated in the proof of Lemma 2 and shown as $\overline{AC}$ in the Figure 4.12b, but using d instead of I_d. The value of r can be calculated from (4.13) replacing d by I_d as well, that is,

$$r = \frac{s\sin(\alpha/2) - I_d/2}{1 - \sin(\alpha/2)}.$$

Then,

$$\begin{aligned}\mathbf{w_2} &= \mathbf{G} + (\mathbf{E_2} - \mathbf{G}) + \frac{I_d}{2}(\cos(z\alpha - \pi/2), \sin(z\alpha - \pi/2)) \\ &= \mathbf{G} + \sqrt{(r+s)^2 - (r + I_d/2)^2}(\cos(z\alpha), \\ &\quad \sin(z\alpha)) + \frac{I_d}{2}(\sin(z\alpha), -\cos(z\alpha)).\end{aligned} \tag{5.6}$$

A similar reasoning is used to calculate $\mathbf{w_3}$ and $\mathbf{w_4}$, however, for the exiting ray, the angle is $(z-1)\alpha$ instead of $z\alpha$. Thus,

$$\begin{aligned}\mathbf{w_3} &= \mathbf{G} + (\mathbf{E_3} - \mathbf{G}) + \frac{I_d}{2}(\cos((z-1)\alpha - \pi/2), \sin((z-1)\alpha - \pi/2)) \\ &= \mathbf{G} + \sqrt{(r+s)^2 - (r + I_d/2)^2}(\cos((z-1)\alpha), \sin((z-1)\alpha)) + \\ &\quad \frac{I_d}{2}(\sin((z-1)\alpha), -\cos((z-1)\alpha)),\end{aligned} \tag{5.7}$$

and

$$\begin{aligned}\mathbf{w_4} &= \mathbf{G} + (\mathbf{E_4} - \mathbf{G}) + \frac{I_d}{2}(\cos((z-1)\alpha - \pi/2), \sin((z-1)\alpha - \pi/2)) \\ &= \mathbf{G} + D(\cos((z-1)\alpha), \sin((z-1)\alpha)) + \frac{I_d}{2}(\sin((z-1)\alpha), \\ &\quad - \cos((z-1)\alpha)).\end{aligned} \tag{5.8}$$

For the sector z, the curve with radius r has centre $\mathbf{c}$, which is distant by $r + s$ from the target centre. The vector $\mathbf{c} - \mathbf{G}$ has orientation $z\alpha - \frac{\alpha}{2}$ (Figure 5.6), so

$$\mathbf{c} = \mathbf{G} + (r+s)\left(\cos\left(\left(z - \frac{1}{2}\right)\alpha\right), \sin\left(\left(z - \frac{1}{2}\right)\alpha\right)\right). \tag{5.9}$$

In order to keep track of the path following, a state machine is employed. Every robot has six states. The initial state is *going_to_target*. The robot remains in this state until it reaches the distance D from the target, then it changes to *going_to_entrance_straight_path*. In this state, the robot follows an orbit following vector field centred at the target centre $\mathbf{G}$ and radius D using Algorithm 3 with the parameter $\mathbf{w_f} = \mathbf{w_1}$. The robot changes to the state *on_entrance_straight_path* when the Algorithm 3 outputs $t' \leq 0$. This state indicates that the robot is following the straight-line vector field towards the target region until it reaches a position after the line orthogonal to the following line at the waypoint $\mathbf{w_2}$. This happens when the variable t' returned by the Algorithm 2 is greater than or equal to 1.

When this occurs, the robot changes to *on_entrance_curved_path* and is impelled by the sum of two forces: the orbit following force for the curved trajectory with centre at $\mathbf{c}$ and radius r and an attraction force towards the target centre $\mathbf{G}$. The second force was added because, next to the target region, as the number of robots grows, repulsive forces may push a robot on state *on_entrance_curved_path* far from the target. The orbit following force field by itself does not attract to the target region, so the other force is a counter-effect to this pushing. However, the attractive force magnitude must be higher than the orbit following force magnitude to avoid their vector sum being null. Consequently, a fixed norm $1.5K_{TRVF}$ is used for that

attractive force, sum these two vectors, then normalise the result to K_{TRVF}. For the Algorithm 3 in this state, $\mathbf{w_f} = \mathbf{w_3}$.

When the robot reaches the target region, it changes to *on_exit_curved_path*. In this state, the robot continues to follow the previous orbit following but an attractive force towards the waypoint $\mathbf{w_3}$ is summed by a similar normalised summation as described for the previous state. The transition to the next state, *on_exit_straight_path*, is done when Algorithm 3 returns $t' \leq 0$. Then, the robot follows the straight line following force field until it leaves the circle with radius D around the target centre, changing back to *going_to_target* and going to the next target region. A transition from the state *on_exit_curved_path* to *going_to_target* is also added because D can be not much greater than s, so the robot can leave the circle with radius D due to repulsive forces before making a transition to *on_exit_straight_path*. In that case, not doing that transition may expedite the leaving from the target because the robots going to the target region are attracted to the entrance path and do not often cause congestion on the exit path. To enforce that behaviour and reduce cluttering near the target area, transitions from *going_to_target* and *going_to_entrance_straight_path* to *on_entrance_curved_path* are forbidden.

As the robot is attracted to the next target when it is on *going_to_target*, depending on its position, it could be impelled to cross the previous target region again. Thus, a repulsive force from the circle with radius D and centre at the previous target centre is applied as if it was a huge obstacle. This makes the robots avoid the previous target region giving space to other robots entering or exiting. Figure 5.7a summarises these transitions and Figure 5.7b shows the vector field and the expected state for a robot at the displayed positions assuming no influence of the repulsive force of the other robots.

Finally, Algorithm 4 presents the TRVF algorithm. The condition at line 2 can be checked by a global boolean variable initialised as false before the robot goes to the next target. Without this condition, the robots would change lanes if they get pushed to another one, resulting in more congestion. At lines 9 and 32, it is assumed any chosen repulsive force function.

5.3 Ad Hoc Follower and Mixed Teams

Assume the same scenario described at the beginning of this chapter but this time also consider that there are robots that do not know which algorithm is being used by the individuals in the swarm. Let N be the total number of individuals and $M < N$ the number of robots not knowing the algorithm. These M robots are ad hoc agents in that scenario because, as described in Chapter 1 and Section 3.3, they have the same goal as the others – to get efficiently in the target area – but have no knowledge about them. Inspired by the multi-agent terminology, I call them *ad hoc robots*. Thus,

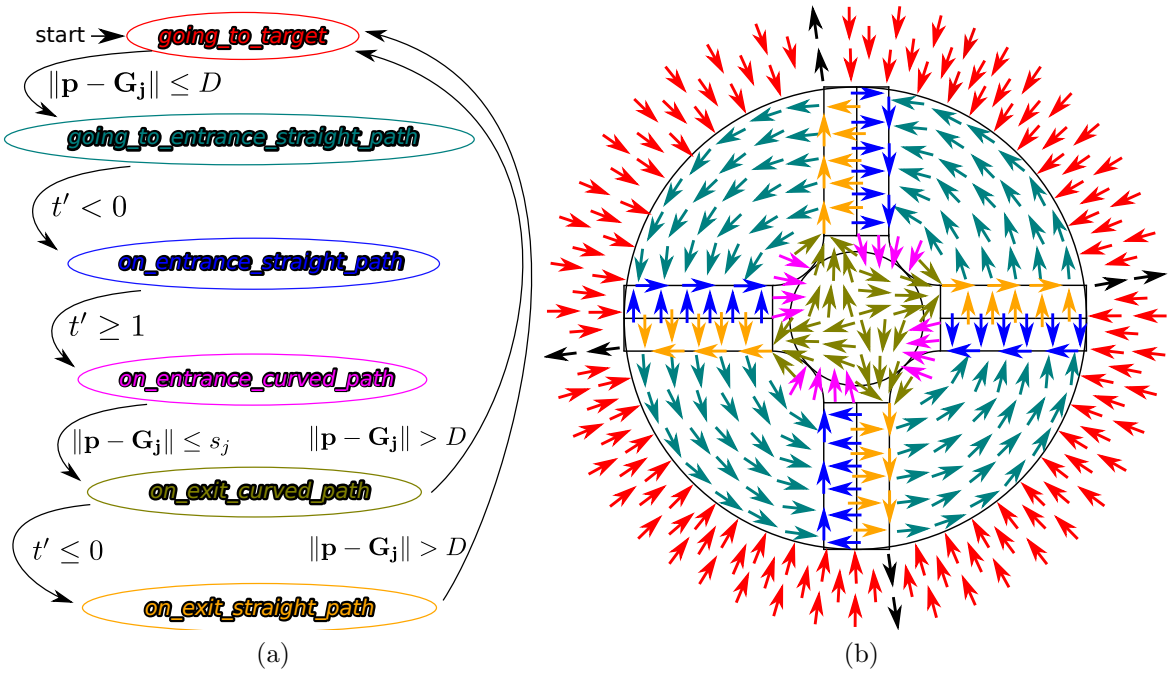


Figure 5.7: Vector field and state machine for the TRVF algorithm. The colours in the vector field represent the expected robot state at the shown vector positions. The black vector indicates the effect of the repulsive force from the target region when the robot is more distant than D from the target already reached. The inner and outer circles have a radius of s and D, respectively. Here, $K = 4$. (a) State machine. (b) Vector field.

Input : K_{TRVF}: force magnitude;
K: number of lanes;
D: the length of the corridor;
$\mathbf{G}_1 = (G_{1,x}, G_{1,y}), \ldots, \mathbf{G}_n = (G_{n,x}, G_{n,y})$: a list of $n \geq 2$ circular target region centres;
$s_1, \ldots, s_n$: a list of n circular target region radii;
j: the current target index;
I_d: influence of the robot;
$k_s > 1, k_o > 1$: constant for exponentiation in the straight line following and orbit following vector fields, respectively;
k_ω: constant for proportional angular speed controller;
v: maximum linear speed.

Output: $\mathbf{F} = (F_x, F_y)$: force vector;

Get position of the robot $\mathbf{p} = (p_x, p_y)$, its orientation ξ, and let $\mathbf{G} = \mathbf{G}_j = (G_x, G_y)$ and $s = s_j$;
if $\mathbf{w_1}, \ldots, \mathbf{w_4}$, *and* $\mathbf{c}$ *were not calculated for target index* j **then**
  $z \leftarrow \left\lfloor \frac{\operatorname{atan2}(p_y - G_y, p_x - G_x)}{2\pi/K} \right\rfloor + 1$;
  Use Eqs. 5.5–5.9 to calculate $\mathbf{w_1}, \ldots, \mathbf{w_4}, r$ and $\mathbf{c}$;
if *state = going_to_target* **then**
  if $\|\mathbf{p} - \mathbf{G}\| \leq D$ **then** state $\leftarrow$ *going_to_entrance_straight_path*;
  else
    $\mathbf{F} \leftarrow K_{TRVF} \frac{\mathbf{G}-\mathbf{p}}{\|\mathbf{G}-\mathbf{p}\|}$;
    $\mathbf{F_R} \leftarrow$ repulsiveForceFromTarget$(\mathbf{G}_{j-1}, D)$;
    return $K_{TRVF} \frac{\mathbf{F}+\mathbf{F_R}}{\|\mathbf{F}+\mathbf{F_R}\|}$;
if *state = going_to_entrance_straight_path* **then**
  $(\mathbf{F}, t') \leftarrow orbitPathFollowing(K_{TRVF}, \mathbf{G}, D, \mathbf{p}, \xi, \mathbf{w_1}, k_o, k_\omega, v)$;
  if $t' \leq 0$ **then** state $\leftarrow$ *on_entrance_straight_path*;
  else return $\mathbf{F}$;

Algorithm 4: TRVF algorithm (continues on Algorithm 5).

if *state = on_entrance_straight_path* **then**
  $(\mathbf{F}, t') \leftarrow straightPathFollowing(K_{TRVF}, \mathbf{p}, \xi, \mathbf{w_1}, \mathbf{w_2}, I_d, k_s, k_\omega, v)$;
  if $t' \geq 1$ **then** state $\leftarrow$ *on_entrance_curved_path*;
  else return $\mathbf{F}$;
if *state = on_entrance_curved_path* **then**
  if $\|\mathbf{G} - \mathbf{p}\| \leq s$ **then**
    Increment j, then let $\mathbf{G} = \mathbf{G}_j = (G_x, G_y)$ and $s = s_j$;
    state $\leftarrow$ *on_exit_curved_path*;
  else
    $(\mathbf{F_1}, t') \leftarrow orbitPathFollowing(K_{TRVF}, \mathbf{c}, r, \mathbf{p}, \xi, \mathbf{w_3}, k_o, k_\omega, v)$;
    $\mathbf{F_2} \leftarrow 1.5 K_{TRVF} \frac{\mathbf{G}-\mathbf{p}}{\|\mathbf{G}-\mathbf{p}\|}$;
    return $K_{TRVF} \frac{\mathbf{F_1}+\mathbf{F_2}}{\|\mathbf{F_1}+\mathbf{F_2}\|}$;
if *state = on_exit_curved_path or state = on_exit_straight_path* **then**
  if $\|\mathbf{p} - \mathbf{G}_{j-1}\| > D$ **then**
    state $\leftarrow$ *going_to_target*;
    $\mathbf{F} \leftarrow K_{TRVF} \frac{\mathbf{G}-\mathbf{p}}{\|\mathbf{G}-\mathbf{p}\|}$;
    $\mathbf{F_R} \leftarrow$ repulsiveForceFromTarget$(\mathbf{G}_{j-1}, D)$;
    return $K_{TRVF} \frac{\mathbf{F}+\mathbf{F_R}}{\|\mathbf{F}+\mathbf{F_R}\|}$;
  else
    if *state = on_exit_curved_path* **then**
      $(\mathbf{F_1}, t') \leftarrow orbitPathFollowing(K_{TRVF}, \mathbf{c}, r, \mathbf{p}, \xi, \mathbf{w_3}, k_o, k_\omega, v)$;
      if $t' \leq 0$ **then**
        $\mathbf{F_2} \leftarrow 1.5 K_{TRVF} \frac{\mathbf{w_3}-\mathbf{p}}{\|\mathbf{w_3}-\mathbf{p}\|}$;
        return $K_{TRVF} \frac{\mathbf{F_1}+\mathbf{F_2}}{\|\mathbf{F_1}+\mathbf{F_2}\|}$;
      else state $\leftarrow$ *on_exit_straight_path*;
    if *state = on_exit_straight_path* **then**
      $(\mathbf{F}, t') \leftarrow straightPathFollowing(K_{TRVF}, \mathbf{p}, \xi, \mathbf{w_3}, \mathbf{w_4}, I_d, k_s, k_\omega, v)$;
      return $\mathbf{F}$;

Algorithm 5: TRVF algorithm (continuation).

two distinct groups of robots are split from N: M ad hoc robots and $N - M$ *aware* robots (i.e., which know the swarm congestion control algorithm and execute it).

Contrary to the usual approach in the multi-agent systems' literature, where the ad hoc agents have to use processing time to learn about the other agents, I initially propose a follower algorithm for ad hoc robots to overcome this problem that depends on a distributed alternative algorithm: the Ad Hoc Follower (AHF). As shown later, the experiments with this approach show that only using an alternative algorithm without following any robot has a better outcome. Thus, our approach is to ad hoc robots execute a fixed alternative algorithm. If this alternative algorithm is the same used by the aware robots, there will be no impact in the swarm resulting from mixing them. However, if they are different, this results in Mixed Teams (MT), and they may affect negatively. The impact of different algorithms is analysed in Section 5.4.1.3. Heed that both AHF and MT do not need learning about other robots, and each individual can execute them distributively and by local sensing.

A robot executing the AHF follows any robot that reaches its sensing area and is in a direction around the one it would follow to the target area. As in the previous algorithms, the robots have an influence radius I_d for sensing other robots. Let $\mathbf{p}$ be the position of a ad hoc robot and $\mathbf{q}_i$, $i \in \{1, \ldots, V\}$, be the position vectors in relation to $\mathbf{p}$ of the V robots inside the influence circle of the ad hoc robot. If $V = 0$, that is, there is no robot inside its influence circle, the robot executes an alternative algorithm Alg, which is supposed to return the force vector for accessing or leaving the common target area. Let $\mathbf{F}_{Alg}$ be the vector output by Alg. If $V > 0$, denote i^* a number in $\{1, \ldots, V\}$ representing the index of the robot's position $\mathbf{q}_i$ inside the influence circle having the smallest angle in relation to the vector $\mathbf{F}_{Alg}$, i.e.,

$$i^* = \underset{i \in \{1,\ldots,V\}}{\operatorname{argmax}} \frac{\mathbf{q}_i \cdot \mathbf{F}_{Alg}}{\|\mathbf{q}_i\| \|\mathbf{F}_{Alg}\|}, \tag{5.10}$$

since the smallest angle has the largest cosine between those vectors.

The ad hoc robot follows the robot at $\mathbf{q}_{i^*}$ if its angle with $\mathbf{F}_{Alg}$ is less than a fixed γ_{ad}. Assume $\gamma_{ad} \leq \pi/2$ to avoid deadlocks due to circular trajectories and robots being pushed away from the target area for long periods. Ties are broken by the first processed value. This algorithm outputs a force vector with fixed magnitude K_{AHF}. Thus, the final output $\mathbf{F}_{\mathbf{AHF}}$ is

$$\mathbf{F}_{\mathbf{AHF}} = K_{AHF} \frac{\mathbf{F}_{Alg}}{\|\mathbf{F}_{Alg}\|},$$

if no robot is being followed. Otherwise,

$$\mathbf{F}_{\mathbf{AHF}} = K_{AHF} \frac{\mathbf{q}_{i^*}}{\|\mathbf{q}_{i^*}\|}.$$

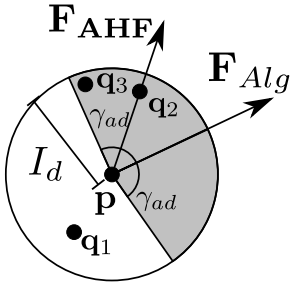


Figure 5.8: Example of output vector for AHF based on the variables. Robots are the black dots, and the area in grey represents the positions of robots that may be followed.

Figure 5.8 depicts an example of these vectors and variables. In it, three robots – at $\mathbf{q}_1$, $\mathbf{q}_2$ and $\mathbf{q}_3$ – are in the influence circle with radius I_d around the mass centre of the ad hoc robot located at $\mathbf{p}$. The robot located at $\mathbf{q}_1$ is outside the area in grey and will not be followed. However, a robot inside it may be followed. This area is calculated using the force vector $\mathbf{F}_{Alg}$ given by the alternative algorithm and the angle γ_{ad}. Hereafter denote it γ_{ad}-area. Among the robots inside this area, only the one having a position with respect to $\mathbf{p}$ with the smallest angle between the vector $\mathbf{F}_{Alg}$ is followed – in this example, the robot at location $\mathbf{q}_2$.

Finally, Algorithm 6 states the above steps for calculating the output force vector, given the mentioned parameters. It assumes a sensing device returning the position $\mathbf{q}_i$ of each locally perceived robot $i \in \{1, \ldots, V\}$ relative to the position $\mathbf{p}$. The followed robot may or may not be another ad hoc robot. This algorithm must be executed in the robot's main loop while getting its current position $\mathbf{p}$. Note that when finding the maximum value in (5.10), the maximum cosine value is initialised with the minimum cosine allowed inside the area given by the angle γ_{ad} in line 6. In addition, observe that $\mathbf{F}_{Alg}$ depends on an alternative algorithm.

Surprisingly, experiments in Section 5.4.1.3 with different values of γ_{ad} for the AHF show that using $\gamma_{ad} = 0$, that is, not following any robot and only executing the alternative algorithm, has better results on the simulation time. Thus, the Mixed Teams (MT) are defined as a team of aware robots using its control algorithm and another of ad hoc robots using an alternative algorithm, which is the same as using AHF with $\gamma_{ad} = 0$. Observe that when $\gamma_{ad} = 0$, the if statement in lines 4–11 and the variable V in line 2 are unneeded. Algorithm 7 presents the outcome of using $\gamma_{ad} = 0$ for clarity.

Interestingly, Section 5.4.1.3 shows that, among the presented algorithms, the best alternative algorithm for less than 30% of ad hoc non-holonomic robots to the total number of individuals and less than 60% of ad hoc holonomic robots is to go in the direction of the target. Hereafter, the absence of a control algorithm, that is, when

Input : K_{AHF}: force magnitude;
I_d: influence of the robot;
Alg: alternative algorithm;
γ_{ad}: maximum angle on the left and right of the direction given by the alternative algorithm to allow following another robot;
$\mathbf{p}$: current robot's position.
Output: $\mathbf{F_{AHF}}$: force vector;

$\mathbf{F}_{Alg} \leftarrow$ vector from Alg;
$V \leftarrow$ number of robots inside the influence circle of radius I_d ;
$\mathbf{F_{AHF}} \leftarrow K_{AHF}\frac{\mathbf{F}_{Alg}}{\|\mathbf{F}_{Alg}\|}$;
if $V > 0$ **then**
  Get all robots positions $\mathbf{q}_1, \ldots, \mathbf{q}_V$ in respect to $\mathbf{p}$;
  $maxCos \leftarrow \cos(\gamma_{ad})$;
  for $i \leftarrow 1$ **to** V **do**
    if $\frac{\mathbf{q}_i \cdot \mathbf{F}_{Alg}}{\|\mathbf{q}_i\|\|\mathbf{F}_{Alg}\|} > maxCos$ **then**
      $maxCos \leftarrow \frac{\mathbf{q}_i \cdot \mathbf{F}_{Alg}}{\|\mathbf{q}_i\|\|\mathbf{F}_{Alg}\|}$;
      $\mathbf{F_{AHF}} \leftarrow K_{AHF}\frac{\mathbf{q}_i}{\|\mathbf{q}_i\|}$;
  end
return $\mathbf{F_{AHF}}$;

Algorithm 6: AHF algorithm.

Input : K_{AHF}: force magnitude;
Alg: alternative algorithm;

Output: $\mathbf{F_{AHF}}$: force vector;

$\mathbf{F}_{Alg} \leftarrow$ vector from Alg;
$\mathbf{F_{AHF}} \leftarrow K_{AHF}\frac{\mathbf{F}_{Alg}}{\|\mathbf{F}_{Alg}\|}$;
return $\mathbf{F_{AHF}}$;

Algorithm 7: AHF algorithm when $\gamma_{ad} = 0$ used by ad hoc robots in MT.

the robots go straightly to the target area while avoiding others by repulsive forces, is denoted as *no coordination* (NC).

5.4 Experiments and Results

The algorithms were tested in realistic simulations on Stage (Gerkey, Vaughan, and Howard, 2003) for holonomic and non-holonomic robots to show that it can be applied to different types of robots and the divergent outcomes when using them. The robots measure $0.44 \times 0.44 \times 0.44$ m^3. When testing non-holonomic robots, the control equations given by Luca and Oriolo (1994) are used. Our implementation considered the following equation for the repulsive forces (Siegwart, Nourbakhsh, and Scaramuzza, 2011):

$$\mathbf{F_R} = \begin{cases} -K_r \left(\dfrac{1}{d_{qp}} - \dfrac{1}{I} \right) \dfrac{\mathbf{q}-\mathbf{p}}{d_{qp}^3}, & \text{if } d_{qp} < I, \\ 0, & \text{otherwise,} \end{cases} \tag{5.11}$$

where $K_r > 0$ is a constant, $\mathbf{p}$ the current position of the robot, $\mathbf{q}$ the position of the neighbour, $d_{qp} = \|\mathbf{q}-\mathbf{p}\|$ the Euclidean distance between $\mathbf{q}$ and $\mathbf{p}$, and I the influence radius.

Experiments were run with the robots starting in a random position distant between 13 and 21 m from the target centre. After reaching the common target, robots will go towards the next one, either to the left or right of that target. It is decided randomly, according to a uniform probability (roughly half of the robots would go to the left and half to the right). The new targets are aligned with the common target on the x-axis but far away.

Two kinds of experiments were performed, one for testing the algorithms SQF, TRVF and AHF and another to compare them with the state-of-the-art algorithms. Hyperlinks to the video of executions are available in the captions of each corresponding figure. They are in real-time so that the reader can compare the time and screenshots presented in the figures with those in the supplied videos. (The source codes of the experimented algorithms are in Passos (2022b) and Passos (2022a).) Additionally, Appendix D presents more figures, complementing the findings in this chapter.

5.4.1 Experiments with the Proposed Algorithms Separately

SQF, TRVF and AHF algorithms are first individually analysed here. The default values of the parameters used for the algorithms are shown in Tables 5.1 and 5.2. For each experiment, 40 executions were run, and the average and the confidence interval such that $\rho = 0.01$ are reported in the graphs. An appropriate statistical test

with such ρ is performed each time statistical significance or "significantly better" are mentioned. Since robots may rarely not reach the target in some simulations due to local minima from potential fields, they were stopped after 60 minutes for SQF and TRVF algorithms. This value was chosen because the total simulation time obtained at the runs where all robots reached the target was less than 45 minutes. However, for AHF, this occurred for less than 450 minutes, so it was increased to 600 minutes.

Parameter	*Notation*	*Value*	*Used by algorithm(s)*
Circular target area radius	s	3 m	All
Working radius of the algorithm around the target	D	13 m	All
Coefficient for repulsive forces	K_r	0.5	All
Default influence radius	I_d	3.0 m	All
SQF force magnitude	K_{SQF}	2.5	SQF
Minimum influence radius	I_{min}	1.0 m	SQF
TRVF force magnitude	K_{TRVF}	2.5	TRVF
Number of lanes	K	5	TRVF
Constant for exponentiation in the straight line following	k_s	1.1	TRVF
Constant for exponentiation in the orbit following vector fields	k_o	1.1	TRVF
Constant for proportional angular speed controller	k_ω	3	TRVF
Maximum linear speed	v	1 m/s	TRVF
PCC force magnitude	K_{PCC}	2.5	PCC
Radius of the free region	r_σ	3.7 m	PCC, PCC-EE
Outer radius of the danger region	r_γ	5.2 m	PCC, EE, PCC-EE
Radius of the α-area	δ	3 m	PCC, PCC-EE
Communication radius	δ_{comm}	3 m	PCC, PCC-EE
Number of iterations before sending a message	T_{comm}	25	PCC, PCC-EE
Number of iterations for testing if a waiting robot will change state	T_η	40	PCC, PCC-EE

Table 5.1: Default values for simulation parameters with the notation used in this work and algorithms that use it. It continues in Table 5.2.

The next sections present the experiments and results for SQF, TRVF and AHF

Parameter	*Notation*	*Value*	*Used by algorithm(s)*
Angle of the α-area to switch to *waiting* state	α_w	$23\pi/36$ rad	PCC, PCC-EE
Angle of the α-area to switch to *locked* state	α_l	$\pi/4$ rad	PCC, PCC-EE
Probability of impatience	ρ_I	0.3	PCC, PCC-EE
EE force magnitude	K_{EE}	2.5	EE
Angle of the entry region	ω	$2\pi/3$ rad	EE, PCC-EE
PCC-EE force magnitude	$K_{PCC\text{-}EE}$	2.5	PCC-EE
AHF force magnitude	K_{AHF}	2.5	AHF
Alternative algorithm	Alg	NC	AHF
Angle of the γ_{ad}-area to follow another robot	γ_{ad}	0 rad	AHF

Table 5.2: Continuation of Table 5.1.

algorithms varying certain parameters, and then a comparison between all algorithms is presented.

5.4.1.1 SQF

Sample of the Simulations. Figure 5.9 shows an execution of the SQF algorithm, with 100 non-holonomic robots with default values. Figure 5.9a shows the initial positions of the robots around the target area (small circle in the centre). The larger circle around the target shows the distance D where the rotational field will take place. Hence, robots inside the larger circle perform a rotational movement towards the corridor and soon reach the state in Figure 5.9b, where the first robots reach the corridor. These robots will directly move towards the target area (Figure 5.9c), and then leave the region following a different rotational field, in yellow in Figures 5.9d and 5.10e. Eventually, all robots can reach the target. Robots in black are at a distance greater than D and are now going towards their next target, randomly chosen between left and right sides.

Comparison with Theoretical Throughput. The throughput for a growing number of robots is analysed with holonomic and non-holonomic robots. Figure 5.11 displays the results for the experiments compared with the least bound of (4.11) (that is, when $\theta = \pi/6$) using the mean distance between each robot and its closest neighbour and the mean linear speed in all experiments for each number of robots. These values do not follow a normal distribution. Thus, instead of plotting the

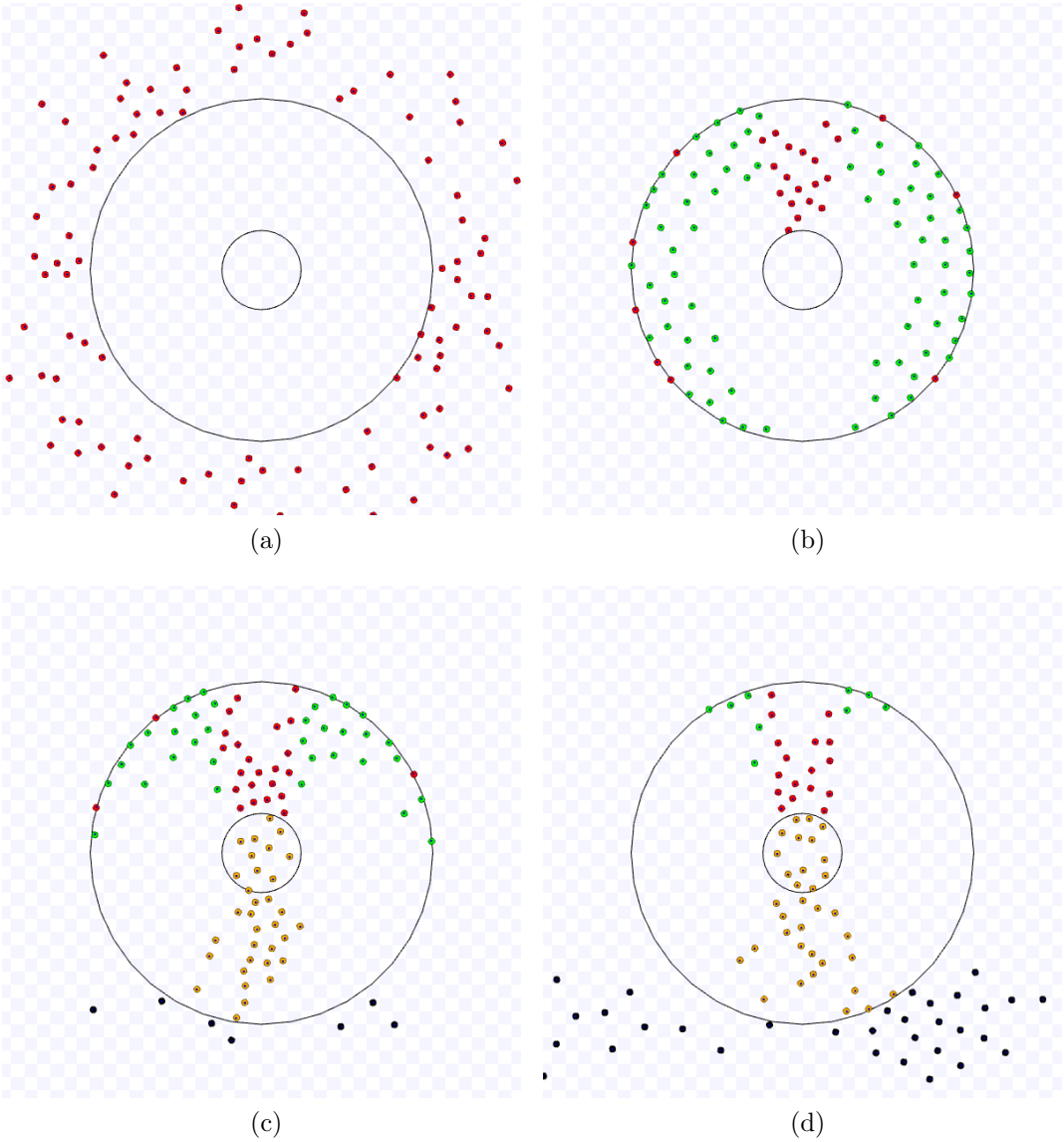
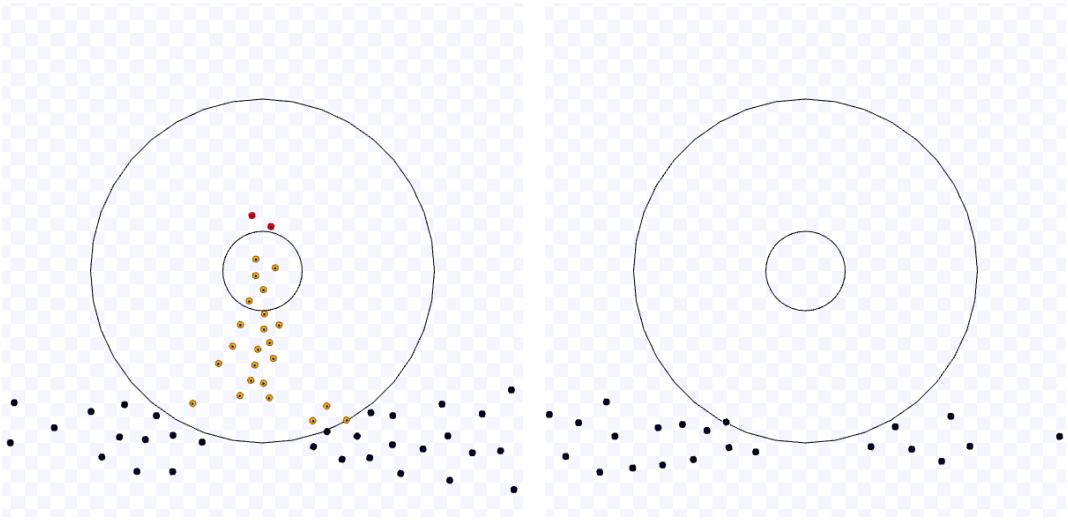


Figure 5.9: Screenshots of the SQF algorithm, with 100 non-holonomic robots with default values. The red, green and yellow robots are in state *going_to_target*, *going_to_corridor* and *leaving_target*, respectively. Black robots are going to their next target. This continues in Figure 5.10. Available on `https://youtu.be/3-d7Y7eViW4`, accessed on 20 December 2022. (a) 0 s: Initial positions. (b) After 49.8 s. (c) After 142.8 s. (d) After 214.3 s.



(e) (f)

Figure 5.10: Continuation of Figure 5.9. (e) After 285.7 s. (f) 357.2 s: ending of the simulation.

confidence interval with $\rho = 0.01$, these means shifted to above and below by one standard deviation are used to calculate the interval shown in Figure 5.11. In other words, for every number of robots N, let $\overline{v_N}$ and $\overline{d_N}$ be the mean of the linear speed and the distance between the robots from all experiments with the same number of robots and σ^{v_N} and σ^{d_N} their standard deviations. The middle value of the theoretical throughput is obtained from the least bound of (4.11) by using the $\overline{v_N}$ and $\overline{d_N}$ in place of v and d for each experiment with the same number of robots. The upper bar value is the least bound of (4.11) applying $\overline{v_N} + \sigma^{v_N}$ and $\overline{d_N} - \sigma^{d_N}$, and the lower bar value uses $\overline{v_N} - \sigma^{v_N}$ and $\overline{d_N} + \sigma^{d_N}$. The inversion in the signs is because the speed and distance are directly and inversely proportional to the throughput, respectively. Moreover, for the graph labelled "Experiments", the mean throughput was calculated by averaging the result of Definition 2 in each experiment with the same number of robots, and the bars correspond to the confidence interval with $\rho = 0.01$. Thus, by using $\overline{v_N} \pm \sigma^{v_N}$ and $\overline{d_N} \mp \sigma^{d_N}$, the range of the ratio $\frac{v}{d}$ in (4.11) is larger than the range of the confidence interval obtained from the experiments.

Observe in Figure 5.11 that the results obtained from the experiments are still below the upper bound obtained by the mean values but inside the one standard deviation interval. The theoretical throughput for holonomic robots maintains a steady mean value as the number of robots grows. Nevertheless, for non-holonomic robots, the mean value reaches a maximum of 140 robots and decreases as more robots are in the swarm. In the non-holonomic case, the mean velocity and the mean

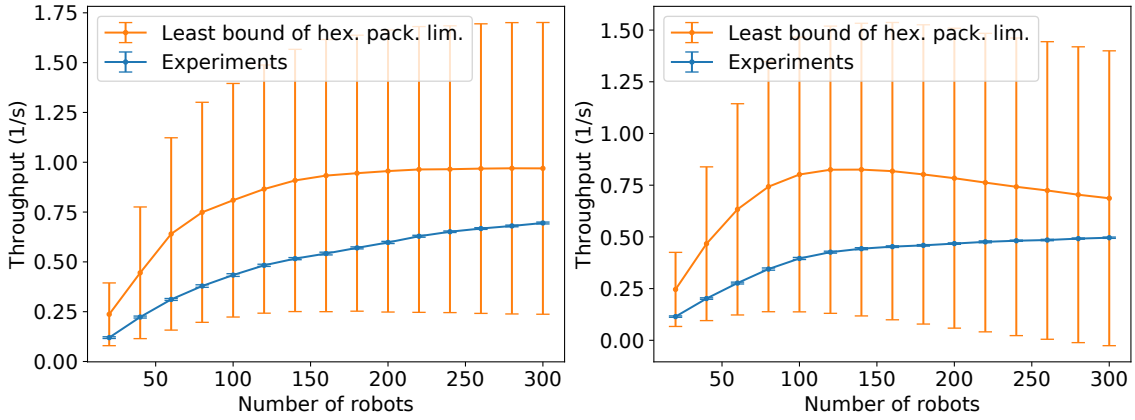


Figure 5.11: Throughput of the SQF algorithm by the number of robots from 20 to 300 in steps of 20 for the experiments and the least bound (4.11) with $\theta = \pi/6$, using the mean distance between the robots and mean linear speed from experiments – bars represent the shift of the means by one standard deviation to above and below. (a) Holonomic. (b) Non-holonomic.

distance of the robots vary more for non-holonomic robots due to the restrictions in their movements. (This is discussed in more detail in Section 6.2.) Although the high variation in the distance between the robots and linear speed contributes to the difference, the SQF algorithm forms a corridor only in the upper part of the circle with a radius of D around the target. Hence, the robots still need time to get to the corridor. Also, the positions of the robots in the corridor are not too compacted.

5.4.1.2 TRVF

For the repulsive force away from the previous target region, one similar to (5.11) is used but the distance is considered from the robot position and the circle with radius D centred at the target position, and the influence radius is D, that is,

$$\mathbf{F_{TR}} = \begin{cases} -K_r \left(\frac{1}{d_{gp}} - \frac{1}{D}\right) \frac{1}{d_{gp}^2} \frac{\mathbf{G}' - \mathbf{p}}{\|\mathbf{G}' - \mathbf{p}\|}, & \text{if } d_{gp} < D, \\ 0, & \text{otherwise,} \end{cases} \tag{5.12}$$

where $K_r > 0$ is a constant, $\mathbf{p}$ the current position of the robot, $\mathbf{G}'$ the previous target centre position, and $d_{gp} = \|\mathbf{G}' - \mathbf{p}\| - D$ is the distance from the robot position to the circle.

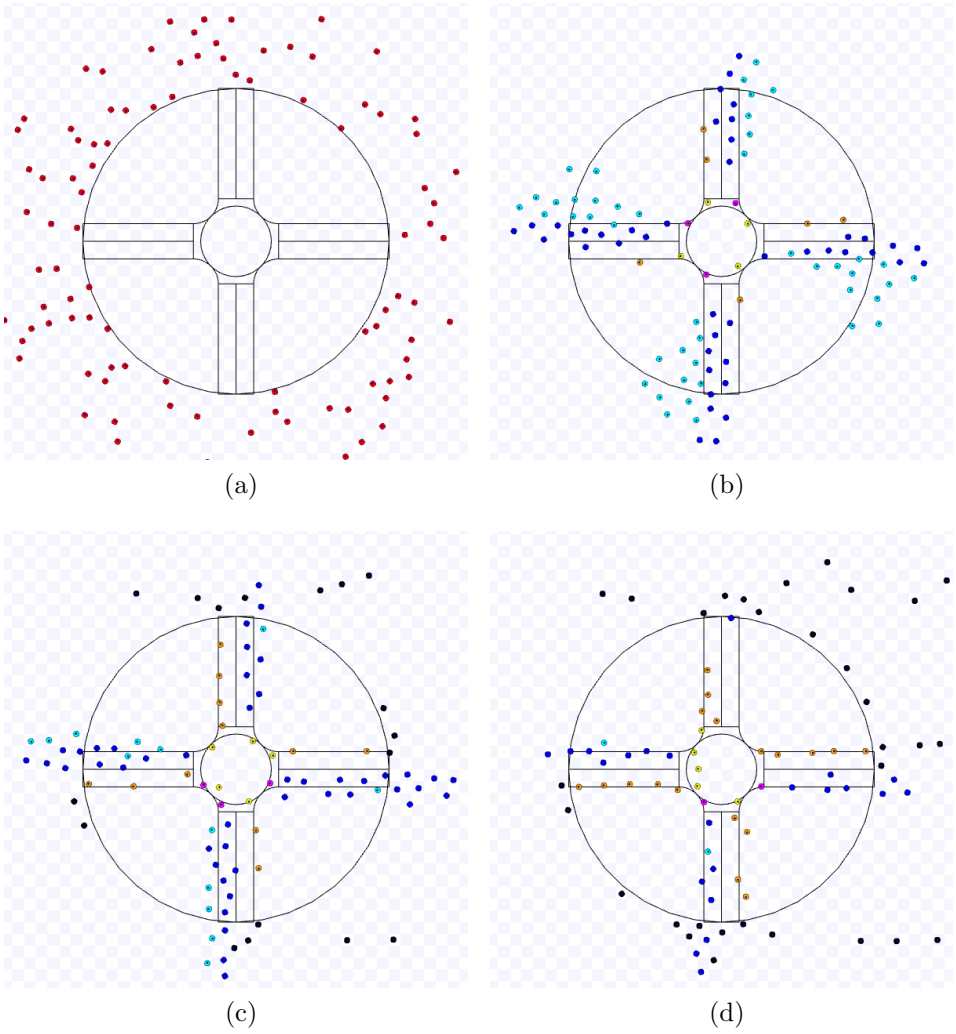
Sample of the Simulations. Figure 5.12 displays an execution of the TRVF algorithm, with 100 non-holonomic robots and four lanes with default values. Figure

5.12a shows the initial positions of the robots around the target area (small circle in the centre). The larger circle around the target shows the distance D, and the robots will avoid it when they go to the next target using (5.12). Each robot goes to the entering lane next to its position and follows the force field (Figure 5.12b). The robots avoid crossing the target region and follow the border of the circle with radius D when going to the next target in Figures 5.12c, 5.12d and its continuation on Figure 5.13 so that all robots can reach the target. With this algorithm, the cluttering is distributed through the entrances and the target region according to the number of lanes, while the leaving lanes become free. However, without the TRVF algorithm, the clutter would concentrate only next to the target region.

Comparison with Theoretical Throughput. As done for the SQF algorithm, the throughput for a growing number of robots is analysed by considering holonomic and non-holonomic robots. Figure 5.14 displays the results of the experiments in comparison with the theoretical maximum throughput value. The theoretical maximum throughput was obtained by (4.19) for $K = 5$ with the mean distance between the robots and mean speed in all experiments for each number of robots. (Tests with different values of K are presented in Section D.1.) As before, to plot the offset from the mean values, the standard deviation is used instead of the confidence interval because these values do not follow a normal distribution. Thus, the upper and lower bar values in the asymptotic plots of Figure 5.14 are obtained by a similar process described for the analogous graph for the SQF algorithm (Figure 5.11) but using (4.19). Identically to the experiments of the SQF algorithm, the graphs labelled "Experiments" are related to the result of Definition 2, and the bars correspond to the confidence interval with $\rho = 0.01$. Hence, as earlier, the range in the asymptotic graph is larger than in the experiments. From Figure 5.14, the experimental throughput is still below the upper bound obtained by the mean values but inside the one standard deviation interval. The difference between the experimental data and the theoretical value obtained from the mean values occurs because, by using variable linear speed, the robots are not constantly paced towards the target. Additionally, next to the curve, the robots deviate from the trajectory to avoid bumping into each other.

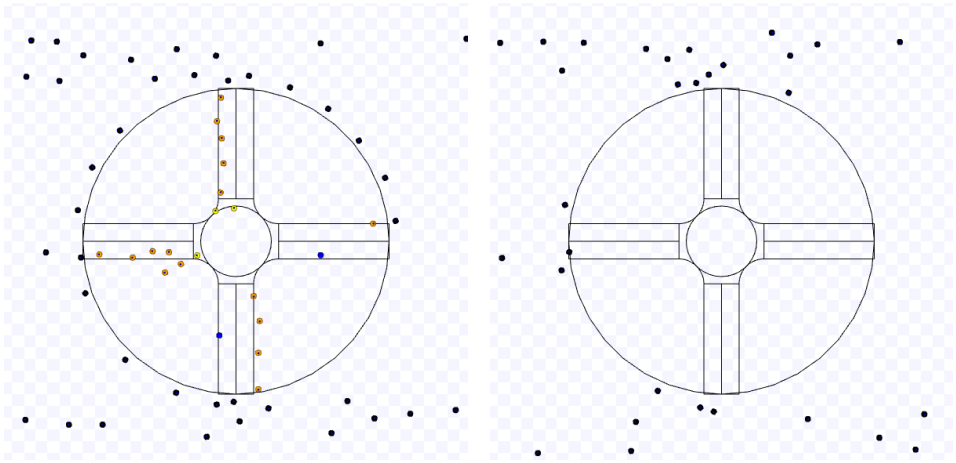
5.4.1.3 AHF and MT

Samples of the Simulations. Simulations were run using different percentages of ad hoc robots to the total number of robots. The colours used by the aware robots are the same as in the previous section. Figures 5.15–5.18 present an execution of AHF using the $\gamma_{ad} = \frac{\pi}{2}$, $Alg =$ NC, a total of 100 holonomic robots and 10% and 50% of ad hoc robots (in grey). The aware robots are running the SQF algorithm with the default values. Figure 5.15a shows the starting position of the robots, and the ad hoc robots are scattered. As they are only 10% of the total number of robots in



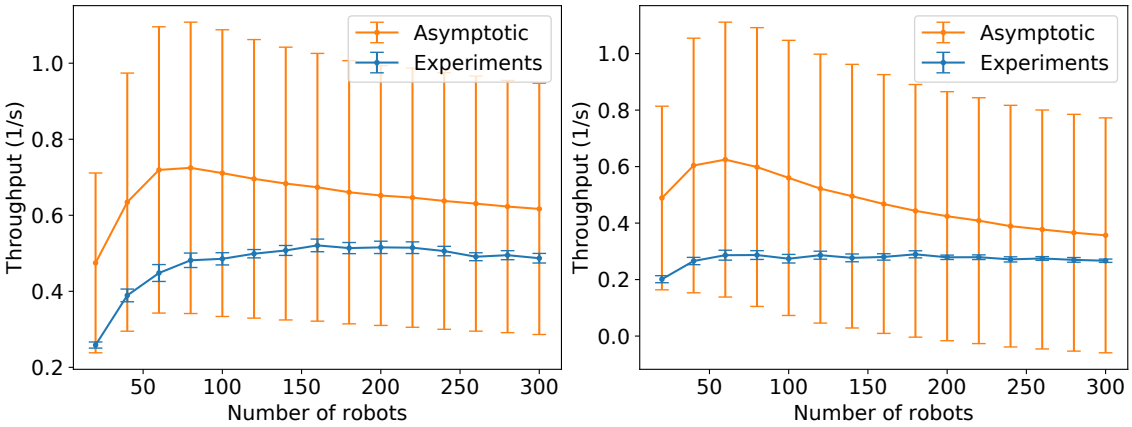
(a) (b) (c) (d)

Figure 5.12: Screenshots of the TRVF algorithm, for 100 non-holonomic robots, four lanes and default values. The colour of the robots are the same for their states shown in Section 5.2, that is, red, cyan, blue, magenta, yellow and orange for *going_to_target*, *going_to_entrance_straight_path*, *on_entrance_straight_path*, *on_entrance_curved_path*, *on_exit_curved_path* and *on_exit_straight_path* states, respectively. Black robots are going to their next target. This continues in Figure 5.13. Available on `https://youtu.be/MRzXS_9I2Ls`, accessed on 20 December 2022. (a) 0 s: Initial positions. (b) After 89.5 s. (c) After 179 s. (d) After 268.5 s.



(e) (f)

Figure 5.13: Continuation of Figure 5.12. (e) After 358 s. (f) 447.5 s: ending of the simulation.



(a) (b)

Figure 5.14: Throughput of the TRVF algorithm by the number of robots from 20 to 300 in steps of 20 for the experiments, and asymptotic throughput using $K = 5$, the mean distance between the robots and mean linear speed from experiments – bars represent the shift of the means by one standard deviation to above and below. (a) Holonomic. (b) Non-holonomic.

this example, they are likely to follow robots knowing the swarm control algorithm. However, a few go directly to the target without encountering any robot in its γ_{ad}-area. There are three such ad hoc robots in Figure 5.15b (marked with circles and arrows): one of those robots is in the bottom area, approaching the target area, and two are outside the corridor on its left-hand side. A few seconds later, one of these two will follow a robot using the SQF next to the target area, and the other will go after the former ad hoc robot, as seen in Figure 5.15c (with circles and arrows). Because the ad hoc robots use NC as the alternative algorithm, after reaching the target area, one goes directly to the next target on the right instead of following a robot through the SQF force field to leave (marked with a circle and an arrow in Figure 5.15d). This also occurs because no other robot was in their γ_{ad}-area oriented by the next target direction. In Figure 5.16e, the ad hoc robots finish the task and follow the others until the end of the experiment (Figure 5.16f).

Figure 5.17a similarly illustrates the starting position of the robots. Now that there are 50% ad hoc robots (in grey) in this example, besides going directly to the target, they may as well follow other ad hoc robots which are going directly to the target, for example, the clusters with grey robots at the bottom of Figure 5.17b. Robots from this cluster either follow the robots leaving the target area while their direction is inside their γ_{ad}-area or other ad hoc robots running by the left-hand side of the target area (Figure 5.17c). However, as various robots come from the target area using the rotational leaving force field of SQF or following a robot doing the same, some will struggle to go to the target (Figure 5.17d). Figure 5.18e shows when the ad hoc robots can go to the target after the robots on the leaving route of the SQF algorithm are outside their γ_{ad}-area. For this experiment, a robot using the SQF was the last to finish (marked with a circle and an arrow in Figure 5.18f). However, in different executions, a robot using AHF could be the last robot.

Figures 5.19–5.22 display executions with 10% and 50% ad hoc robots but using TRVF instead of SQF. Figure 5.19a shows the initial position of the robots, with the ad hoc robots uniformly distributed among the others. As occurred for SQF, even though an ad hoc robot is probable to find another following the TRVF (90% of the total robots), it can go directly to the target without following any other. In this case, that occurs because there is no robot between the ad hoc robot and the target area in the top left quadrant around the target centre in Figure 5.19b (marked with a circle and an arrow). It will try to leave the working area, heading to the next target on the left-hand side. However, it will follow a succession of other robots through the entry lane until leaving the working area (with a circle and an arrow in Figure 5.19c) as they are successively in its γ_{ad}-area. For this number of ad hoc robots, the robots frequently go after robots leaving through the exit lane after reaching the target area (Figure 5.19d). Also, the robots executing AHF often finish before the robots using TRVF (Figure 5.20e).

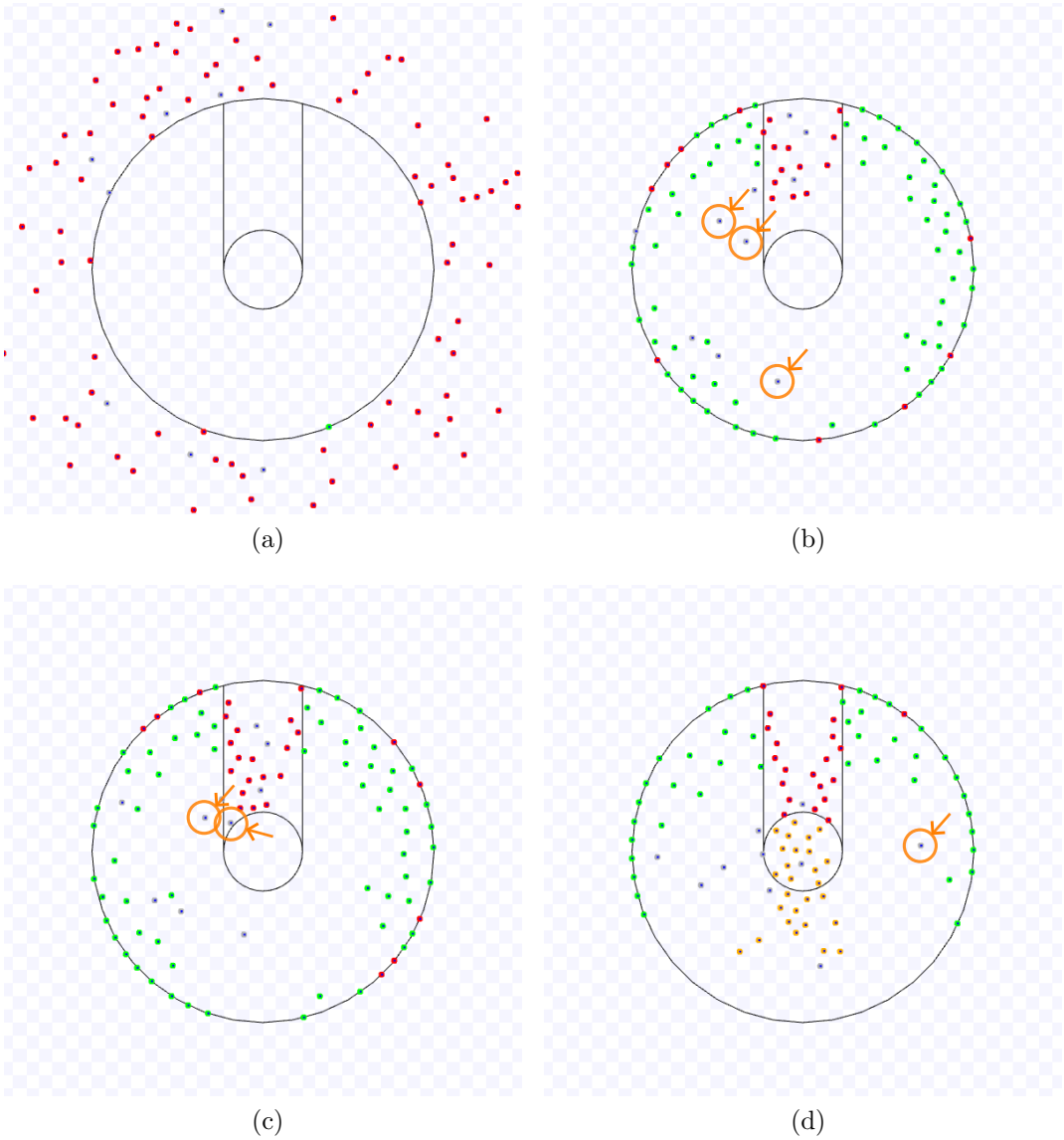
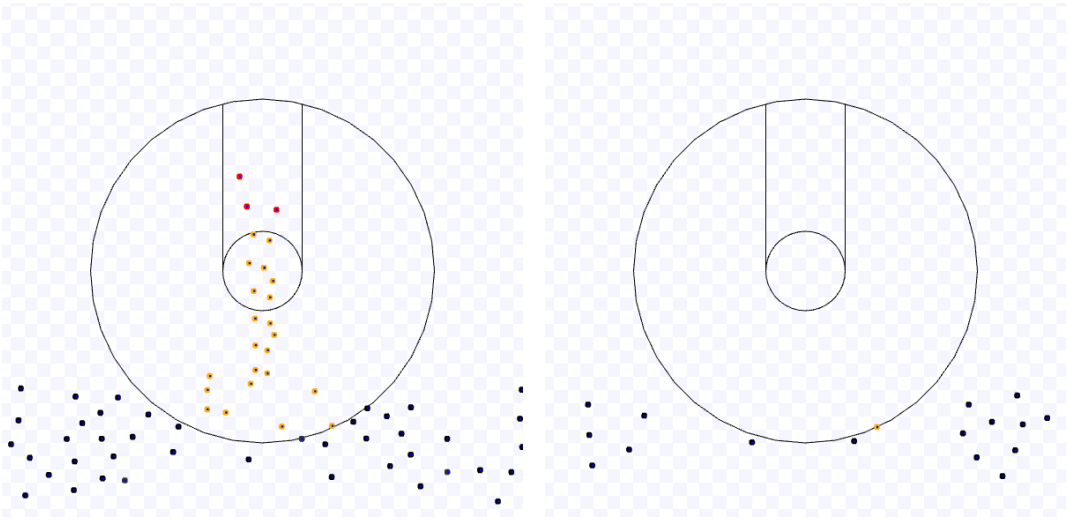


Figure 5.15: Screenshots of 10% of the robots using AHF and the SQF algorithm executed by the aware robots. This continues in Figure 5.16. Available on `https://youtu.be/llHNPOaXgFE`, accessed on 27 April 2023. (a) 0 s: Initial positions. (b) After 33 s. (c) After 42 s. (d) After 89 s.



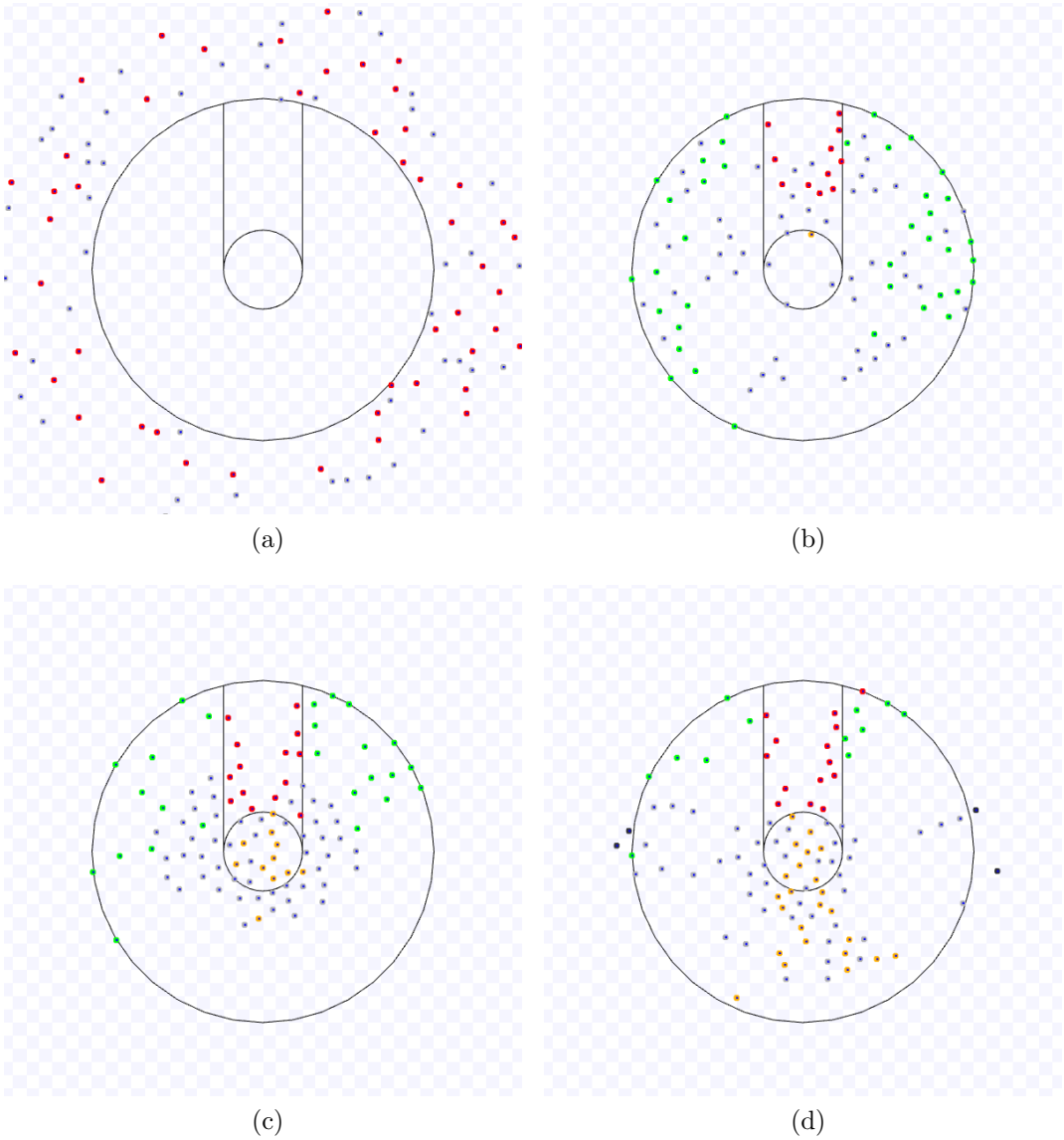
(e) (f)

Figure 5.16: Continuation of Figure 5.15. (e) After 224.3 s. (f) 309.5 s: ending of the simulation.

Figure 5.21a shows the initial position of the robots for the 50% case. As half of the robots use AHF, more robots go directly towards the target (Figure 5.21b). They will try to go directly to the next target, but the other robots are coming through the lanes, and then they will follow any robot inside its γ_{ad}-area. Some will pass through the middle of the target area before, directly or indirectly, following another robot going by the lanes (Figure 5.21c). This causes congestion, interfering with the robots executing TRVF. After the cluttering diminishes (Figure 5.21d), most of the robots with TRVF can access the target area. In Figure 5.22e, the few robots using AHF go after the others through the lanes before the end of this experiment (Figure 5.22f).

Parameter γ_{ad}. In order to compare the value of γ_{ad}, experiments were run to measure the total simulation time with this angle varying from 0 to $\frac{\pi}{2}$ rad in increments of $\frac{\pi}{12}$ and 10%, 50% and 90% of ad hoc robots for a total of 100 and 200 robots with holonomic and non-holonomic robots. Heed that $\gamma_{ad} = 0$ means that the ad hoc robot will not follow any robot. In these experiments, NC, SQF and TRVF are used as alternative algorithms.

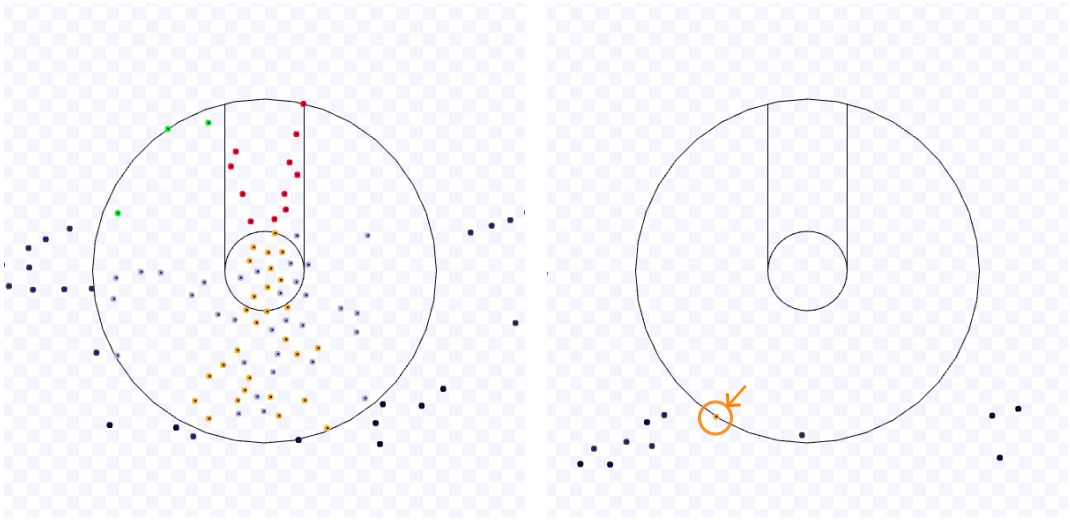
Figures 5.23–5.24 present the results assuming NC as alternative algorithm and the swarm control algorithm is SQF and Figures 5.25–5.26 for the same alternative algorithm, but the TRVF algorithm as swarm control algorithm. Figures 5.27–5.28 present the results for SQF as the alternative algorithm and TRVF as swarm control algorithm. In Figures 5.29–5.30, TRVF is the alternative algorithm and SQF is the swarm control algorithm.



(a) (b)

(c) (d)

Figure 5.17: Screenshots of 50% of the robots using AHF and the SQF algorithm executed by the aware robots. This continues in Figure 5.18. Available on `https://youtu.be/6tZndLOOFEQ`, accessed on 27 April 2023. (a) 0 s: Initial positions. (b) After 44 s. (c) After 76 s. (d) After 107 s.



(e) (f)

Figure 5.18: Continuation of Figure 5.17. (e) After 134.3 s. (f) 279.9 s: ending of the simulation.

For all plots, the minimum average time is at $\gamma_{ad} = 0$ or, when the minimum average is not at $\gamma_{ad} = 0$, it is not different than the result using $\gamma_{ad} = 0$ with statistical significance. In general, the ad hoc robots have to change their orientation whenever they find another using an algorithm going to a different direction. As that algorithm gives force vectors with different directions on those positions, and the γ_{ad}-area is local, the robots may change their trajectory, making the route to and off the target longer than it should be if they were constantly executing one of the algorithms. Thus, using AHF is not significantly better than using MT. Therefore, the default value for parameter γ_{ad} is 0, that is, MT will be used as default.

Parameter *Alg*. The usage of SQF and TRVF as alternative algorithms was compared by varying the number of robots and the ratio of ad hoc robots to the number of robots $\left(\frac{M}{N}\right)$. Figure 5.31 illustrates an example of the comparison of the total simulation time per total number of robots of MT using as alternative algorithms the SQF and NC with 10% of ad hoc robots to the total number of individuals. In these experiments, TRVF is the swarm control algorithm. (Section D.2 have similar figures for percentages ranging from 20% to 90% in steps of 10%.) These figures are summarised in Figure 5.32, which shows the sum of the average total simulation time over each experimented number of robots by the percentage of ad hoc robots for MT. The bars correspond to the respective sum of the confidence interval ranges with $\rho = 0.01$.

SQF as the alternative algorithm takes more or equal time on average than using

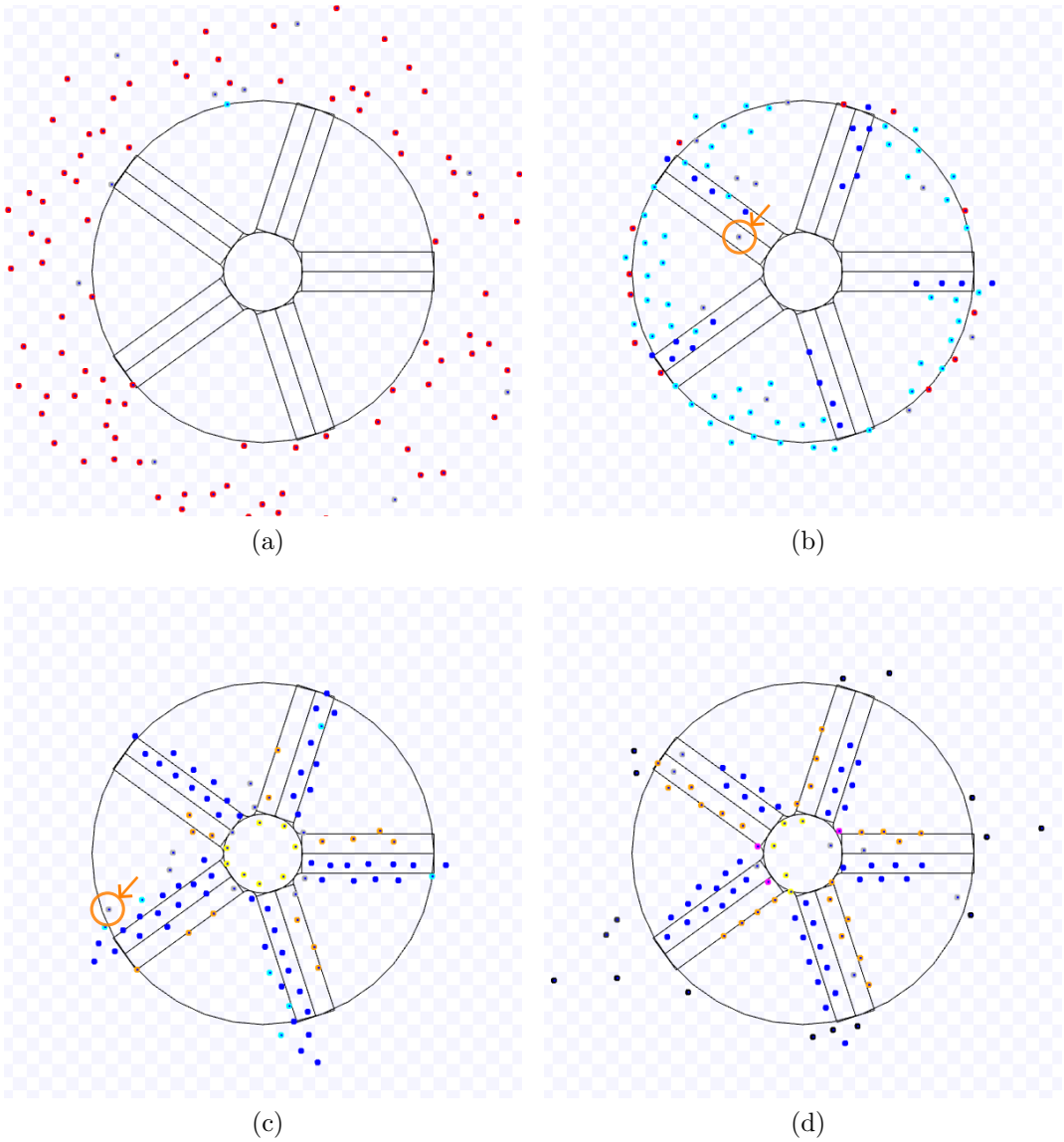
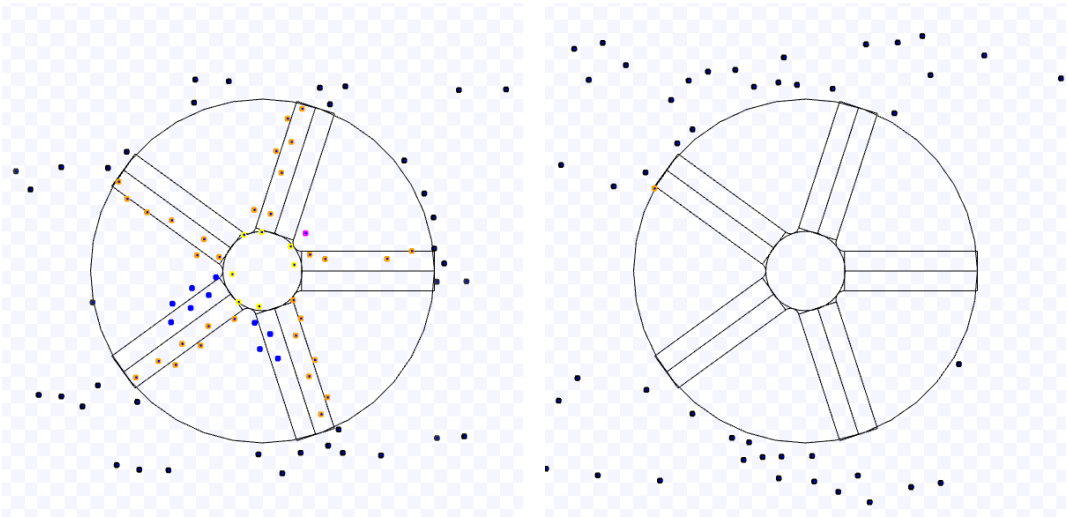


Figure 5.19: Screenshots of 10% of the robots using AHF and the TRVF algorithm executed by the aware robots. This continues in Figure 5.20. Available on `https://youtu.be/3pF4AqwOR6o`, accessed on 27 April 2023. (a) 0 s: Initial positions. (b) After 31 s. (c) After 88.5 s. (d) After 126.6 s.



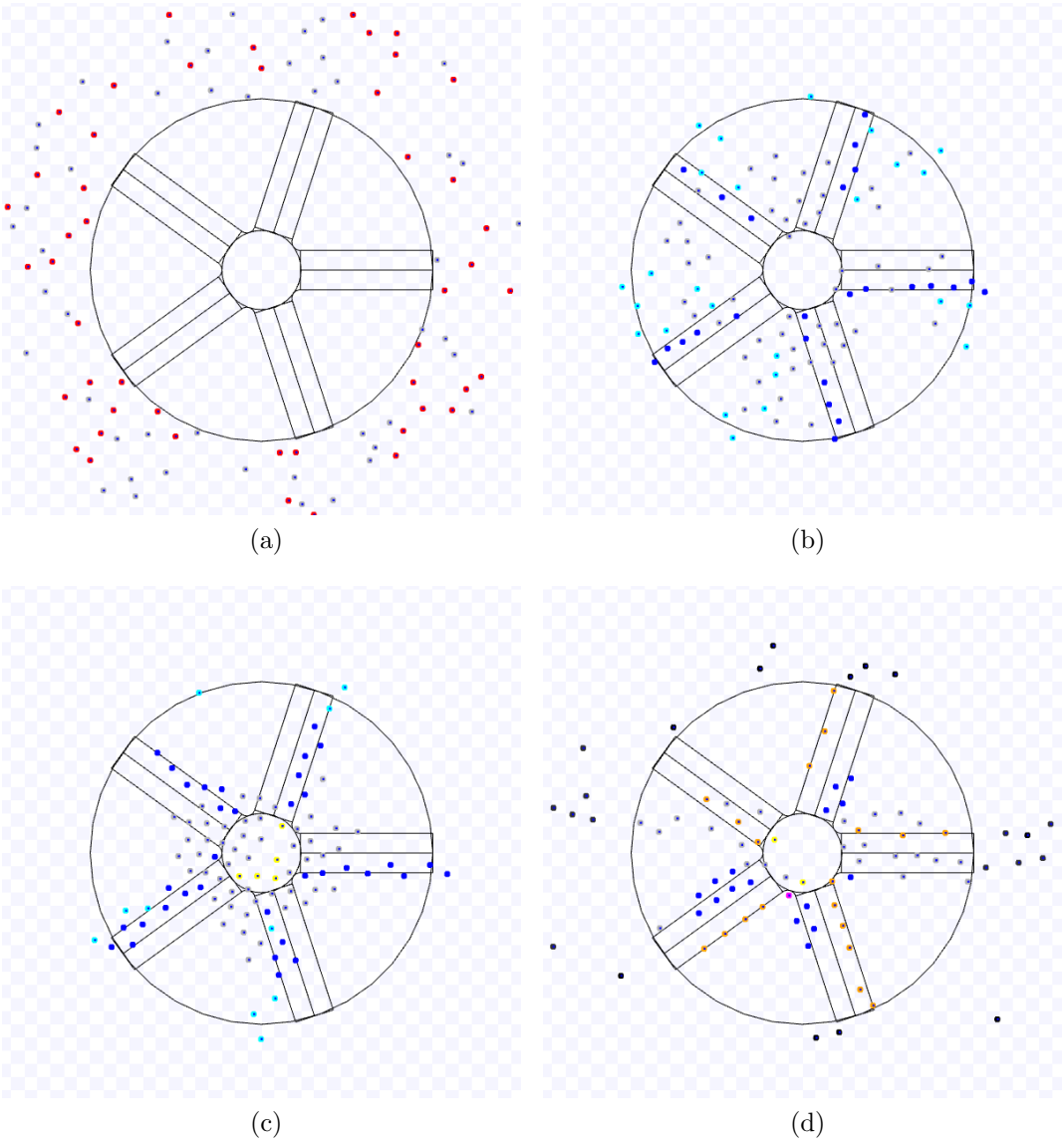
(e) (f)

Figure 5.20: Continuation of Figure 5.19. (e) After 174 s. (f) 264.6 s: ending of the simulation.

NC for percentages from 10% to 20% of ad hoc robots using MT for both type of robots and 30% to 70% for holonomic robots with statistical significance (Figure 5.32). Although the SQF is faster, when used as the alternative algorithm for a low percentage of robots, they follow a trajectory longer than they would if NC is in use. As fewer robots are going directly to the target region and there is less congestion, NC is better for low percentages. The variance is higher for SQF because the SQF robots leaving the target area may push the robots using TRVF away from the working area in the bottom lane for a long time.

Additionally, the usage of holonomic robots takes more time and varies more than non-holonomic robots for the SQF algorithm. Holonomic robots are faster to be repelled by another robot due to their lack of restrictions on moving in any direction. Because they can be repelled for longer periods, they tend to take more time going away from the working area.

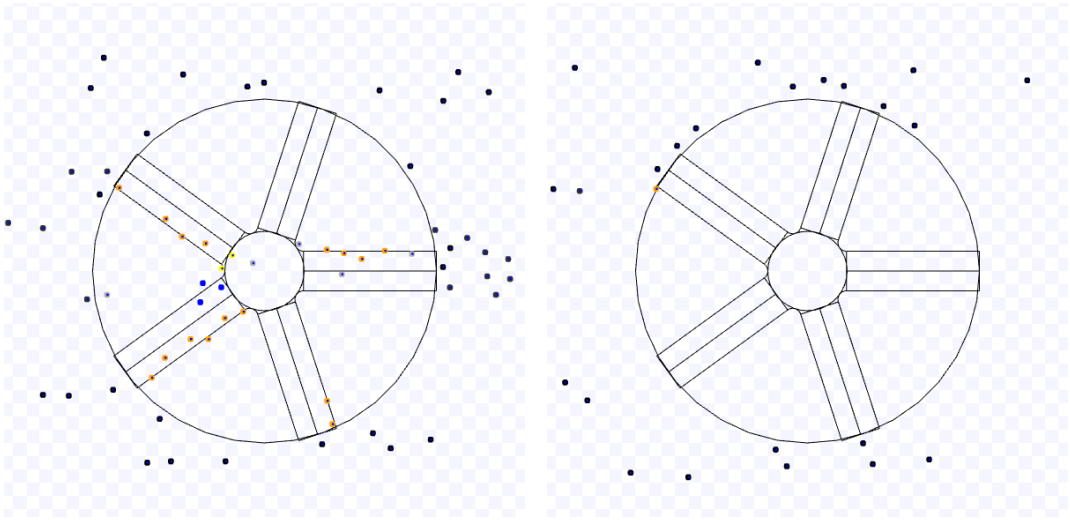
Figure 5.33 shows a sample of the comparison of NC and TRVF as the alternative algorithms with 10% of ad hoc robots to the total number of robots for holonomic and non-holonomic robots. In these experiments, SQF is the swarm control algorithm. Similar figures for percentages ranging from 20% to 90% in steps of 10% are in Section D.2. As before, these figures are summarised in Figure 5.34, which presents the sum over the number of robots of the average total simulation time by the ratio of ad hoc robots. NC as the alternative algorithm takes less or equal time than using TRVF on average for percentages from 10% to 80% for both types of robots with statistical



(a) (b)

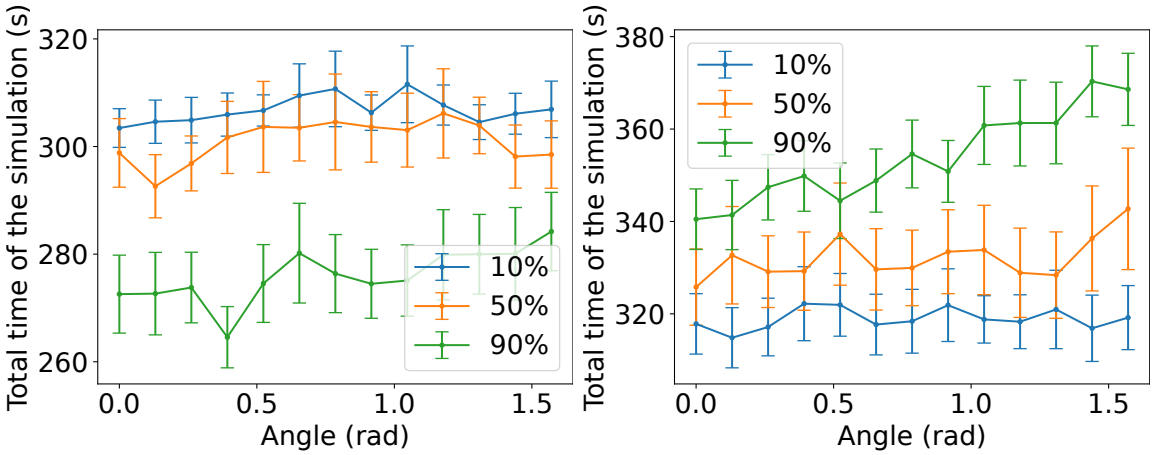
(c) (d)

Figure 5.21: Screenshots of 50% of the robots using AHF and the TRVF algorithm executed by the aware robots. This continues in Figure 5.22. Available on `https://youtu.be/rjWWuVebqK8`, accessed on 27 April 2023. (a) 0 s: Initial positions. (b) After 45 s. (c) After 67 s. (d) After 160 s.



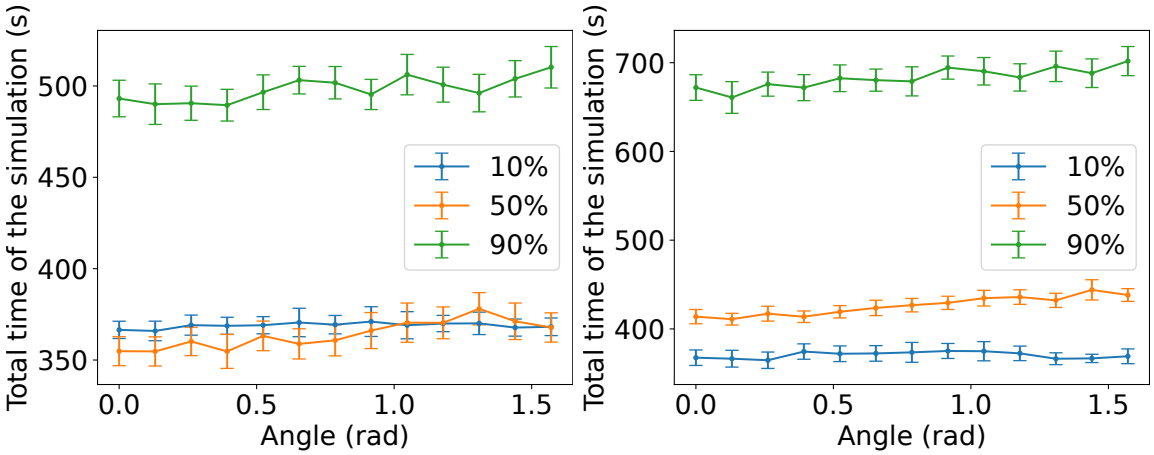
(e) (f)

Figure 5.22: Continuation of Figure 5.21. (e) After 199.7 s. (f) 272.5 s: ending of the simulation.



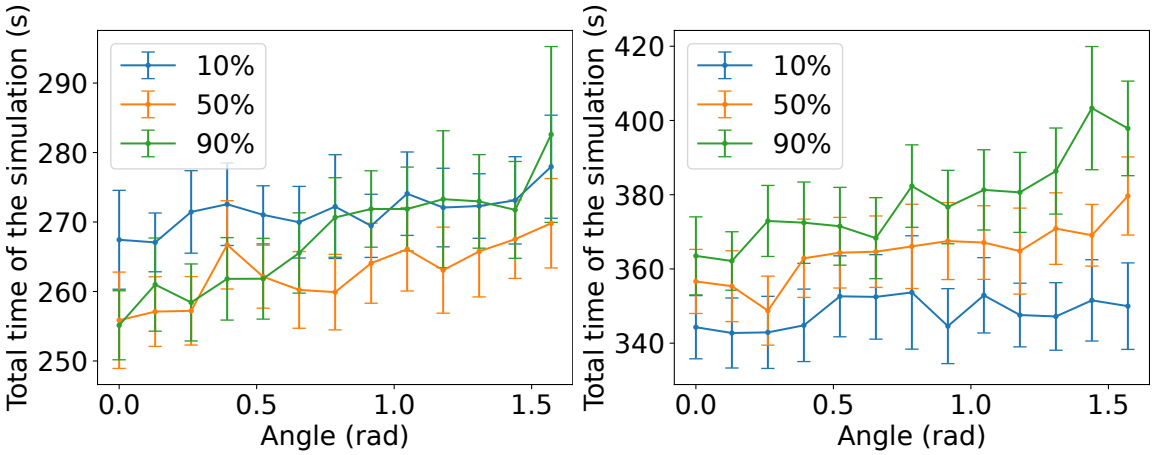
(a) (b)

Figure 5.23: Comparison of the total simulation time by γ_{ad} values for the experiments with AHF with NC as alternative algorithm for 10%, 50%, 90% of the robots executing it and SQF being used by the others with a total of 100 robots for (a) holonomic and (b) non-holonomic robots.



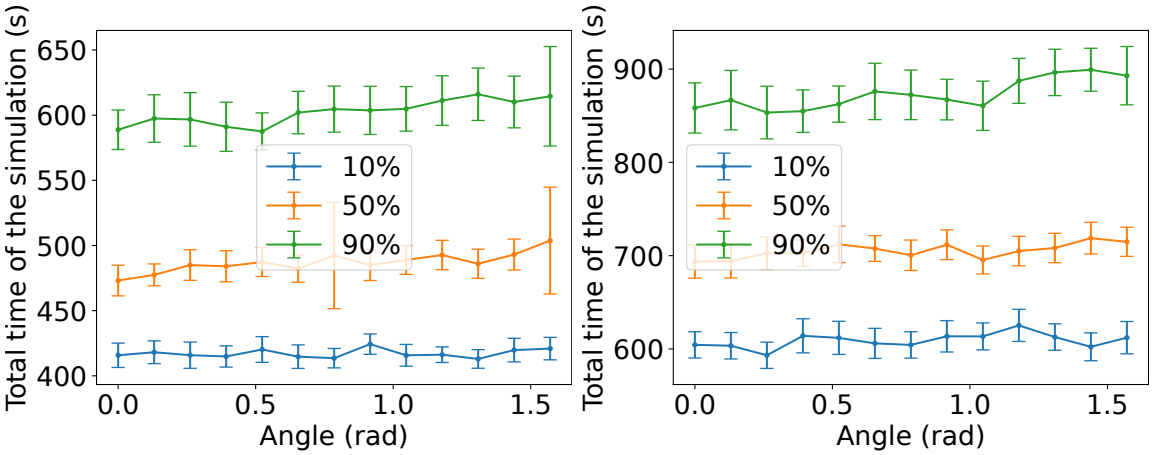
(a) (b)

Figure 5.24: Comparison of the total simulation time by γ_{ad} values for the experiments with AHF with NC as alternative algorithm for 10%, 50% and 90% of the robots executing it and SQF being used by the others with a total of 200 robots for (a) holonomic and (b) non-holonomic robots.



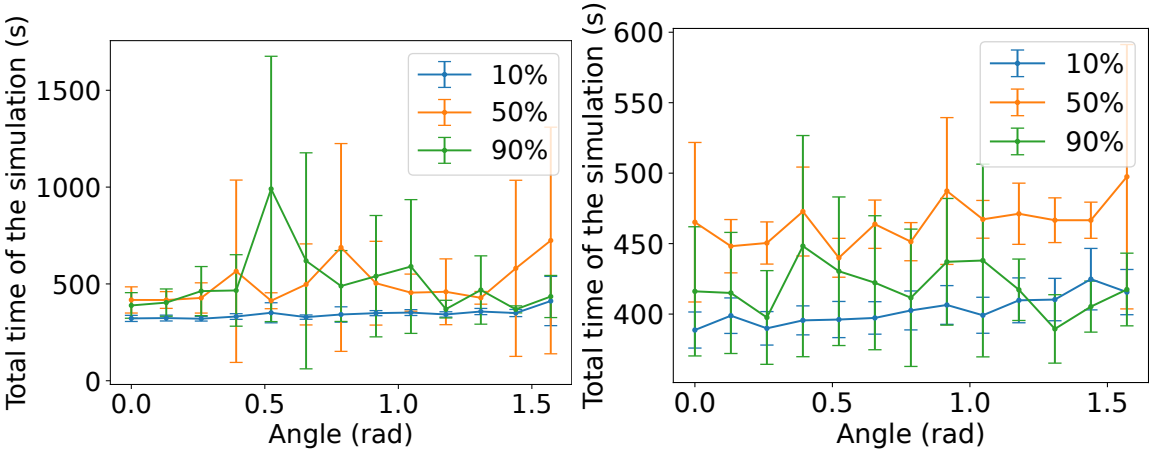
(a) (b)

Figure 5.25: Comparison of the total simulation time by γ_{ad} values for the experiments with AHF with NC as alternative algorithm for 10%, 50% and 90% of the robots executing it and TRVF being used by the others with a total of 100 robots for (a) holonomic and (b) non-holonomic robots.



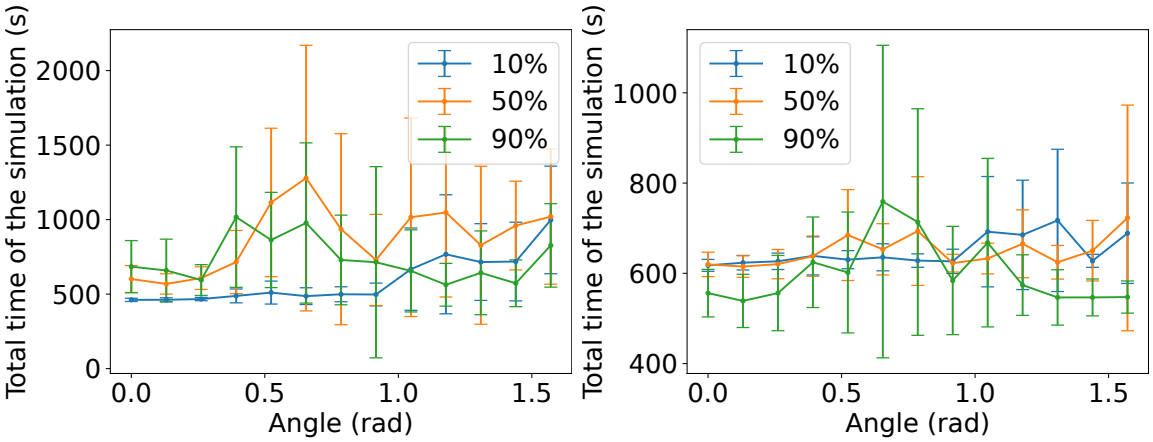
(a) (b)

Figure 5.26: Comparison of the total simulation time by γ_{ad} values for the experiments with AHF with NC as alternative algorithm for 10%, 50% and 90% of the robots executing it and TRVF being used by the others with a total of 200 robots for (a) holonomic and (b) non-holonomic robots.



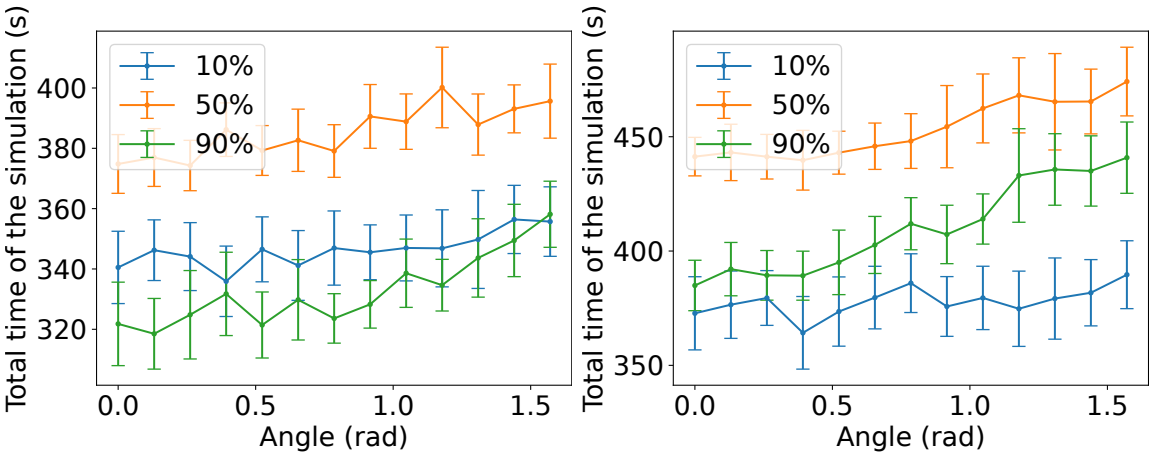
(a) (b)

Figure 5.27: Comparison of the total simulation time by γ_{ad} values for the experiments with AHF with SQF as alternative algorithm for 10%, 50% and 90% of the robots executing it and TRVF being used by the others with a total of 100 robots for (a) holonomic and (b) non-holonomic robots.



(a) (b)

Figure 5.28: Comparison of the total simulation time by γ_{ad} values for the experiments with AHF with SQF as alternative algorithm for 10%, 50% and 90% of the robots executing it and TRVF being used by the others with a total of 200 robots for (a) holonomic and (b) non-holonomic robots.



(a) (b)

Figure 5.29: Comparison of the total simulation time by γ_{ad} values for the experiments with AHF with TRVF as alternative algorithm for 10%, 50% and 90% of the robots executing it and SQF being used by the others with a total of 100 robots for (a) holonomic and (b) non-holonomic robots.

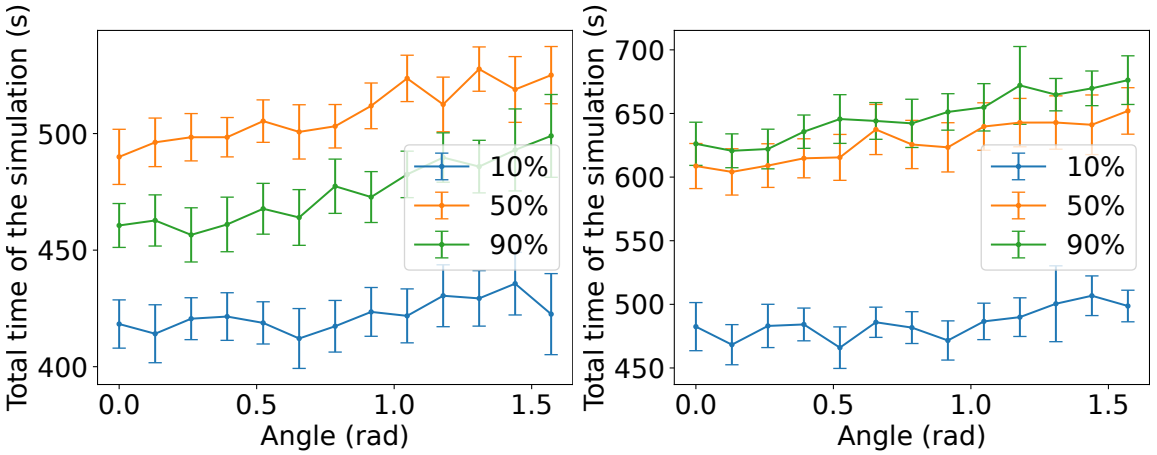


Figure 5.30: Comparison of the total simulation time by γ_{ad} values for the experiments with AHF with TRVF as alternative algorithm for 10%, 50% and 90% of the robots executing it and SQF being used by the others with a total of 200 robots for (a) holonomic and (b) non-holonomic robots.

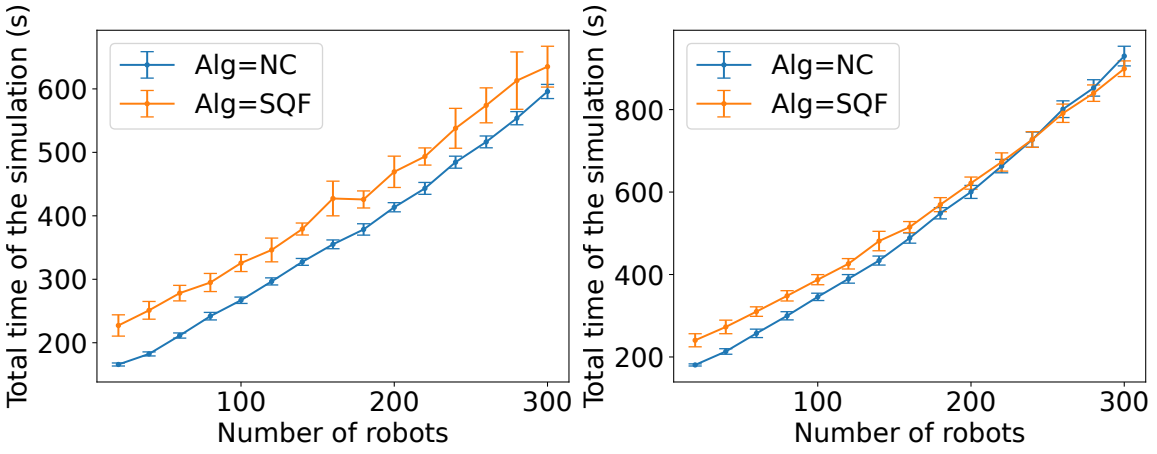


Figure 5.31: Comparison of the total simulation time by the number of robots for the experiments with MT using TRVF as the swarm control algorithm and SQF and NC as the alternative algorithm by 10% of the robots for (a) holonomic and (b) non-holonomic robots.

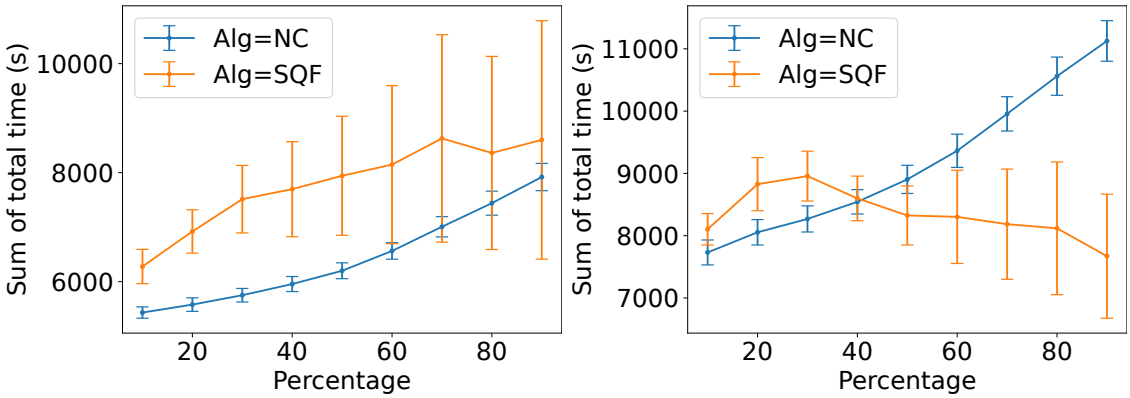


Figure 5.32: Comparison of the total simulation time sum over the number of robots by the percentage of ad hoc robots using NC and SQF as alternative algorithms. The swarm control algorithm of the aware robots is TRVF. (a) Holonomic robots. (b) Non-holonomic robots.

significance.

Surprisingly, although NC is the slowest algorithm for handling the common target problem for numerous robots (Marcolino et al., 2017), it was the best alternative algorithm in most cases with statistical significance. Therefore, I selected it as the default alternative algorithm in our experiments. In addition, I expect that MT would be used for small ratios of ad hoc robots when assuming it would be made of auxiliary robots in an ad hoc teamwork scenario. Remember, however, that SQF and TRVF still outperform NC for this problem as the swarm control algorithm (Passos, Duquesne, and Marcolino, 2023), and our result only relates to the choice of *Alg* for MT. Nevertheless, if the proportion of ad hoc robots and an estimated number of robots are known beforehand, the alternative algorithm can be chosen based on the graphs.

Comparison of Different Ratios of Ad Hoc Robots and Control Algorithms without Ad Hoc Robots. Finally, to show the effect of the ratio of ad hoc robots on the total simulation time, Figures 5.35 and 5.36 show a sample of the simulation time per number of robots for holonomic and non-holonomic robots for 10% of ad hoc robots in MT (that is, robots using AHF with default values in Table 5.2). SQF and TRVF are the congestion control algorithms in Figures 5.35 and 5.36, respectively. (Plots for percentages varying from 20% to 90% in steps of 10% are shown in Section D.4.) Figure 5.37 shows the sum ranging over the number of robots of the average total simulation time by the percentage of ad hoc robots in MT. The bars correspond to the respective sum of the confidence interval ranges with $\rho = 0.01$.

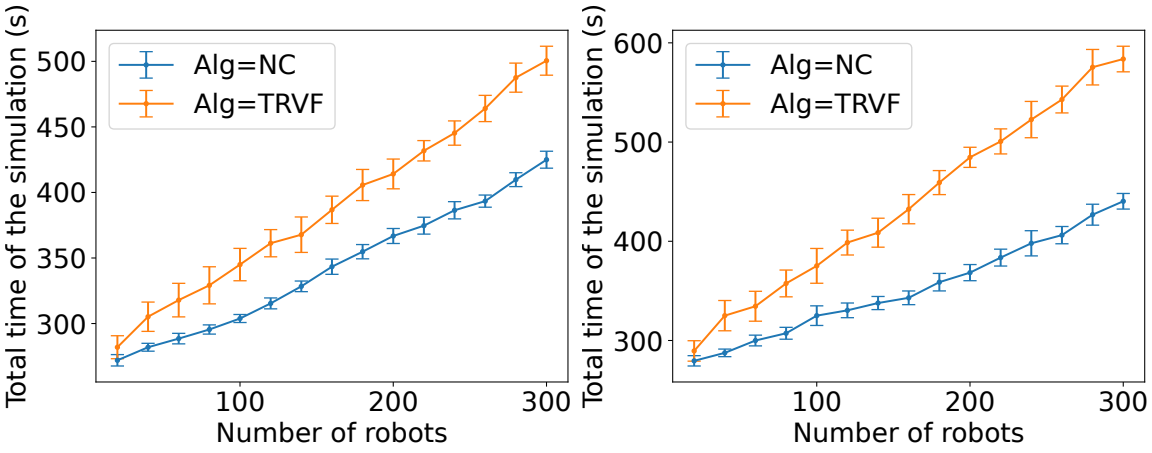


Figure 5.33: Comparison of the total simulation time by the number of robots for the experiments with MT using SQF as the swarm control algorithm and TRVF and NC as the alternative algorithm by 10% of the robots for (a) holonomic and (b) non-holonomic robots.

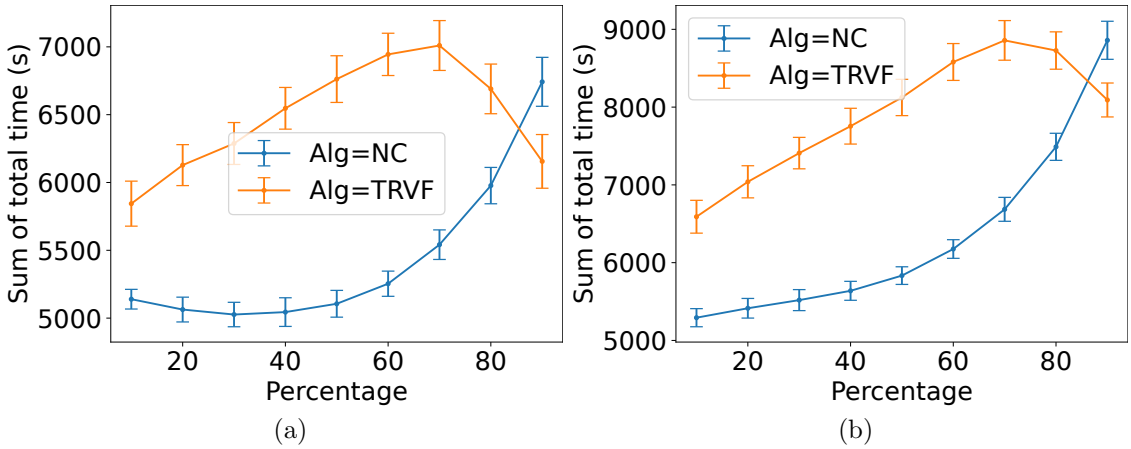


Figure 5.34: Comparison of the total simulation time sum over the number of robots by the percentage of ad hoc robots using NC and TRVF as alternative algorithms. The swarm control algorithm of the aware robots is SQF. (a) Holonomic robots. (b) Non-holonomic robots.

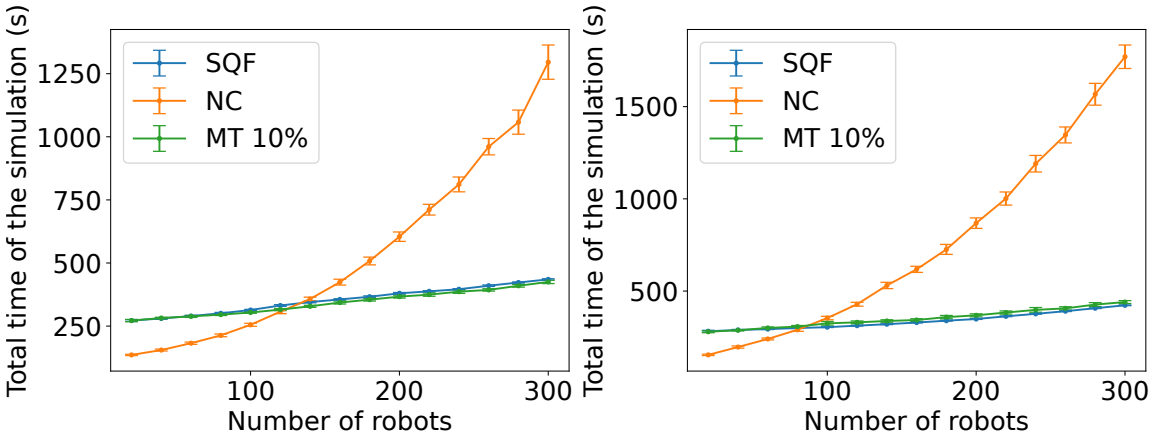


Figure 5.35: Comparison of the total simulation time by the number of robots for the experiments with MT using SQF as the swarm control algorithm and NC as the alternative algorithm by 10% of the robots for (a) holonomic and (b) non-holonomic robots.

In these figures, the results for the algorithm used by the aware robots without any ad hoc robot and for NC are plotted as references.

From these figures, it is inferred that the rise in the total simulation time is influenced by the ratio of ad hoc robots. As this ratio increases, the total simulation time for all number of robots approaches the total time for NC. As noticed before, for the SQF with holonomic robots, low percentages of robots using NC follow a path to and out from the target shorter than SQF. Also, because they have no restriction in movement compared to the non-holonomic robots, there is less congestion, and they can outperform the SQF for such values. In fact, Section 6.2 will show that the total simulation time increases as a linear combination of the alternative algorithm time and the time of the swarm congestion control algorithm. (The effect of one ad hoc robot in MT is presented in Section D.3.)

5.4.2 Comparison with State-of-the-Art Algorithms

In this section, the algorithms are compared regarding (i) the throughput, (ii) the time the swarm reaches the target area, (iii) the time for leaving it averaged by the number of robots, and (iv) the total simulation time. Only the runs that succeed in terminating within 60 minutes are considered in Section 5.4.2.1 and 20 minutes in Section 5.4.2.2. This termination time is lower in the later section because 100 robots were used, while in the former, up to 300. The robots running algorithms PCC, EE

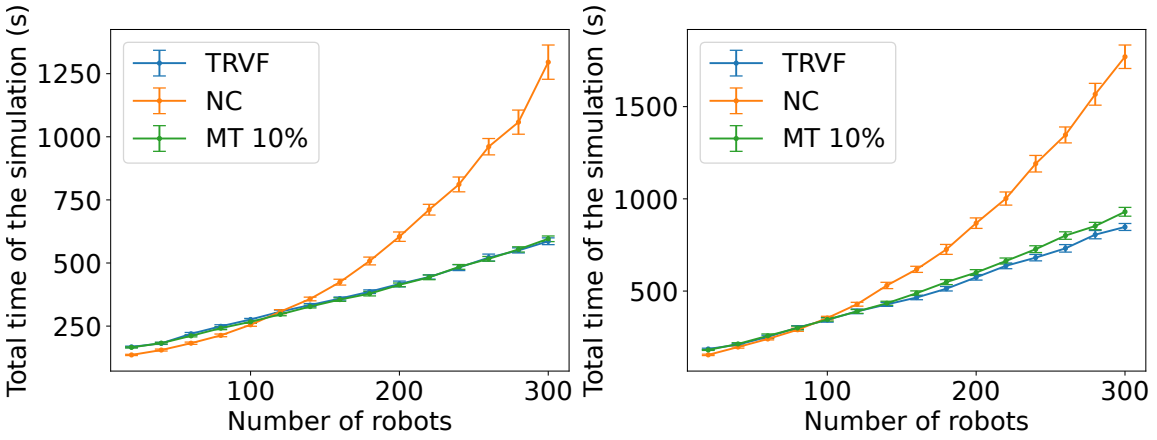


Figure 5.36: Comparison of the total simulation time by the number of robots for the experiments with MT using TRVF as the swarm control algorithm and NC as the alternative algorithm by 10% of the robots for (a) holonomic and (b) non-holonomic robots.

and PCC-EE for the experiments of Section 5.4.2.2 entered deadlock in executions longer than 20 minutes for 100 robots.

The total simulation time is obtained when the last robot in the experiment arrives at the outer circle with a radius of D after everyone has reached the target area. The reaching time is measured from the last robot in the swarm that gets to the target area. The leaving time is taken from the moment a robot reaches the target area until it arrives at the outer circle. As done by Marcolino et al. (2017), to measure the effectiveness of the algorithms on the crowd reduction for leaving the target area, the leaving time of every robot are summed and divided by the number of robots in the experiment.

Additionally, tests were performed with the target area with constant radius (as in Table 5.1) in Section 5.4.2.1 and varying small values in Section 5.4.2.2.

5.4.2.1 Comparison for Constant Target Size

Comparison of the Reaching Time and Throughput. Figures 5.38 and 5.39 show the comparison for a varying number of robots of the target area throughput and the reaching time, respectively. For both types of robots, throughput and time increase with the number of robots. Additionally, observe in the comparisons that higher throughput reflects a lower arrival time.

Moreover, in the throughput graph, for holonomic robots from 240 robots, the

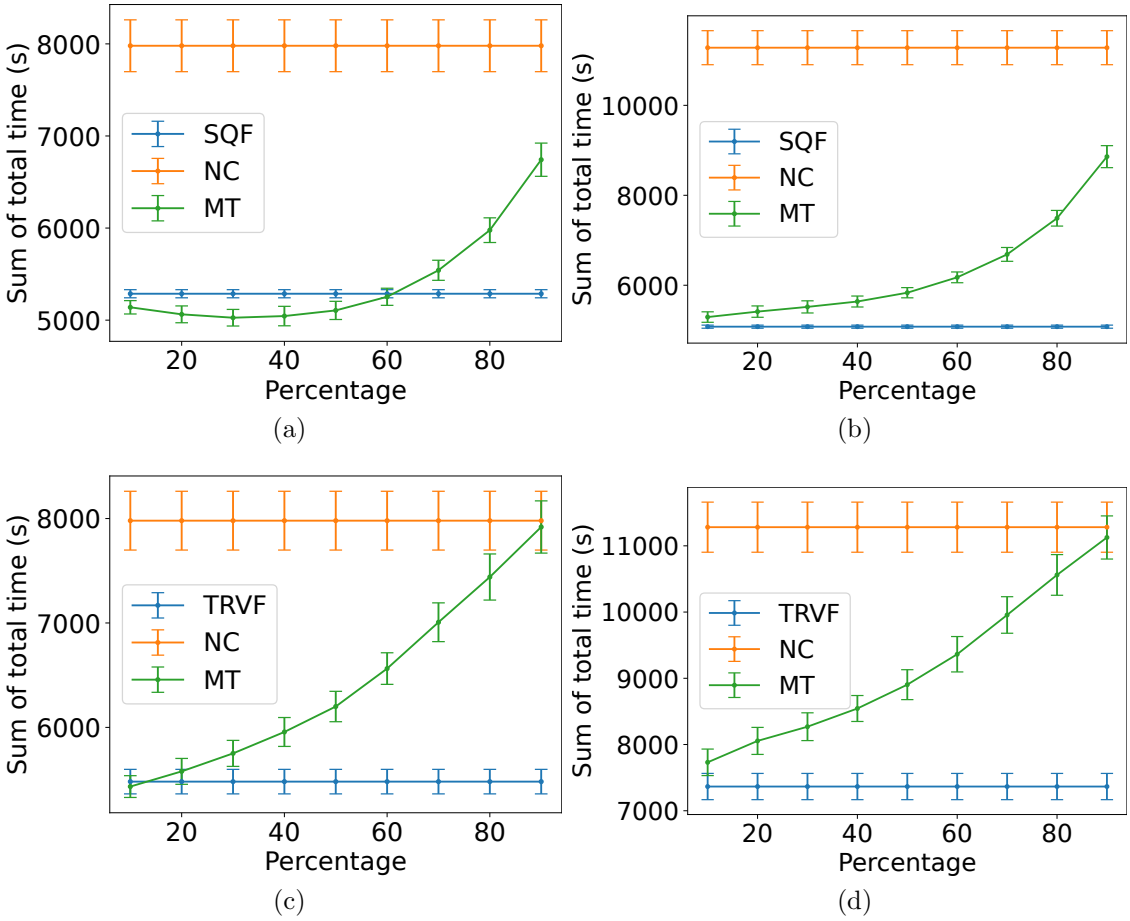


Figure 5.37: Comparison of the total simulation time sum over the number of robots by the percentage of ad hoc robots in MT. For reference, the total simulation time for NC and the swarm control algorithm without any ad hoc robot is shown. SQF is the control algorithm for (a) holonomic and (b) non-holonomic robots and TRVF for (c) holonomic and (d) non-holonomic robots.

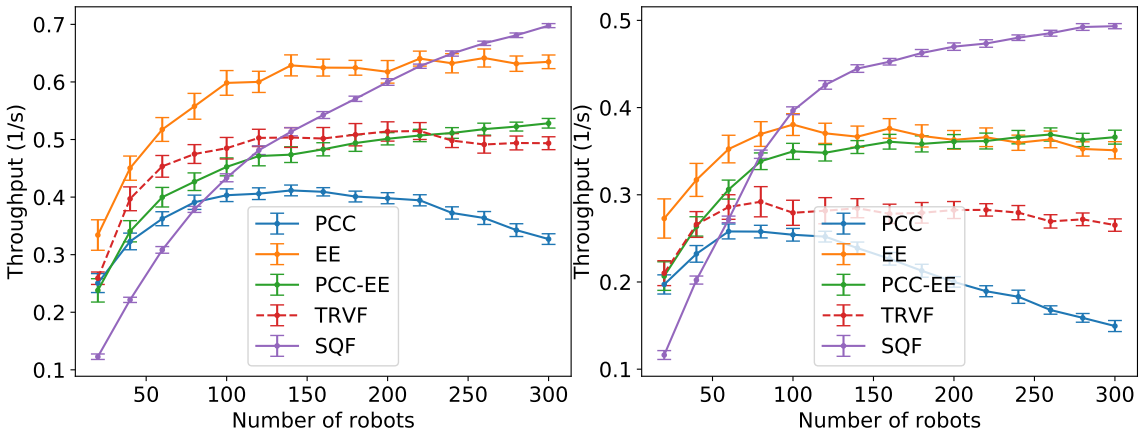


Figure 5.38: Throughput comparison of the algorithms for a number of robots from 20 to 300 in steps of 20. (a) Holonomic. (b) Non-holonomic.

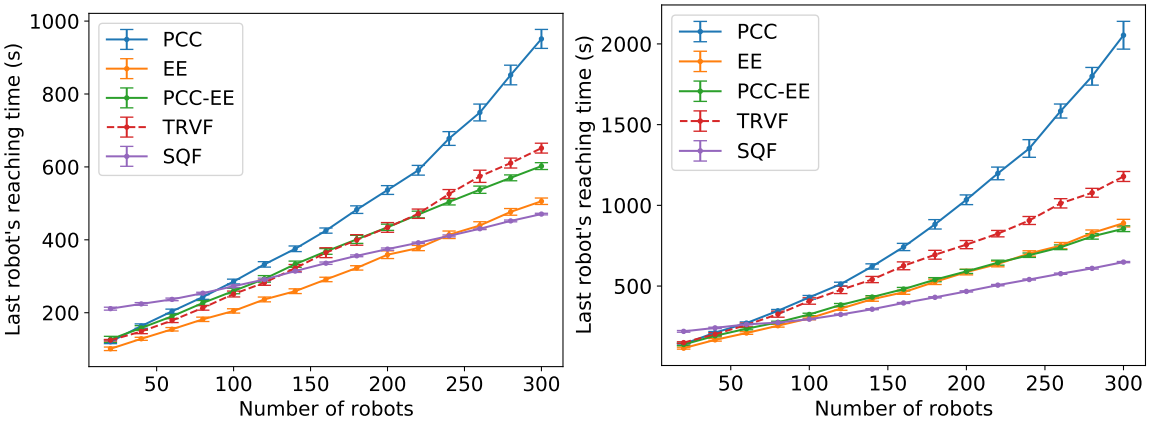


Figure 5.39: Comparison of the time to reach the target of the algorithms for a number of robots from 20 to 300 in steps of 20. (a) Holonomic. (b) Non-holonomic.

SQF algorithm is significantly better than all other algorithms and, from 100 robots, for non-holonomic ones. In the reaching time graph, this only occurs from 280 for holonomic and 120 for non-holonomic ones with statistical significance by t-test. However, below these values, EE is still significantly better in term of throughput. Excluding the EE algorithm, the TRVF algorithm is better than all the remaining algorithms from 40 to 120 individuals for holonomic robots but, for non-holonomic robots, it is only significantly better than SQF and PCC algorithms just for 40 and 60 robots.

Furthermore, unpaired t-tests with $\rho = 0.01$ returned that the reaching time intervals of SQF and EE algorithms for 240 and 260 holonomic robots – Figure 5.39a – have the same mean. These tests showed that the reaching time intervals for 100 non-holonomic robots of SQF and EE algorithms – Figure 5.41b – also have the same mean.

Comparing only TRVF and SQF by the target reaching time, the former is better than the latter until 120 individuals for holonomic robots and up to 40 for the non-holonomic case with statistical significance. Unpaired t-tests returned that SQF and TRVF reaching time mean intervals have different means except for 140 holonomic and 60 non-holonomic robots. As the SQF algorithm organises a queue above the target region, the robots must initially follow the way until reaching that queue. After that, the robots flow through it. On the other hand, in the TRVF, the robots go directly to the target region. However, as the number of robots increases, more congestion happens near this region because of the repulsive forces caused by the robots doing the curve.

Observe as well that, although SQF has higher throughput than TRVF from some number of robots, the comparison of their corresponding inspiration strategies, hexagonal packing and touch and run, had a different result with respect to the asymptotic throughput in Figure 4.18. For calculating asymptotic throughput, the maximum linear speed and the minimum distance between robots are considered constants. Using only asymptotic throughput, one can evaluate which one is the best strategy in a scenario with those fixed quantities. For robots using artificial fields, getting an explicit asymptotic throughput equation is difficult due to the changeability of the speed and the distance between the robots, but this does not prevent using experimental throughput for comparisons. Accordingly, this changeability and the effect of the other robots in the trajectory yield the SQF being significantly better than TRVF, although the touch and run is better than hexagonal packing. Even so, the analytically calculated throughputs are still upper bounds on the ones obtained from simulations, as observed in Figures 5.11 and 5.14.

Comparison of the Average Leaving Time. Figure 5.40 displays the comparison for the number of robots versus the average leaving time. In Figure 5.40a, the TRVF algorithm is significantly better than all the others until 240 individuals

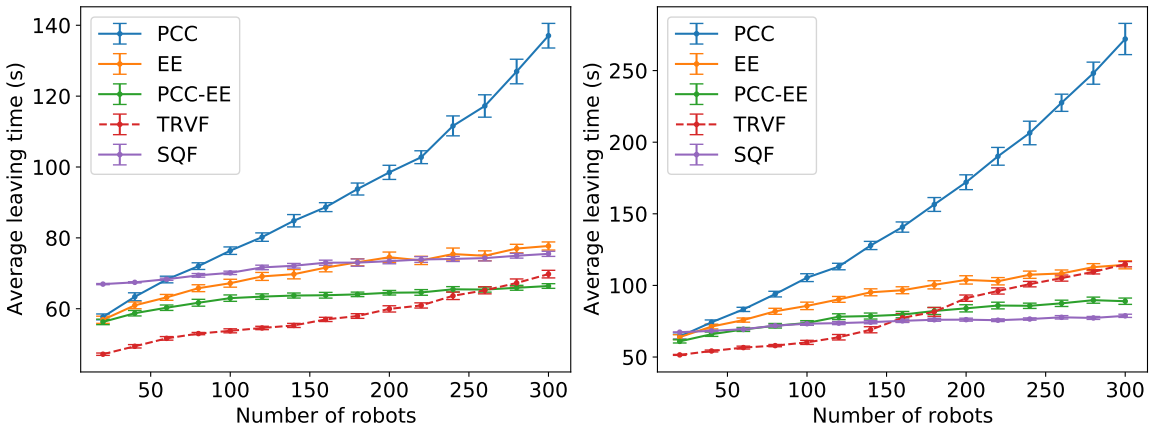


Figure 5.40: Comparison of the algorithms for a number of robots from 20 to 300 in steps of 20 versus the sum of the time for leaving the target area of all robots in the experiment divided by the number of robots. (a) Holonomic. (b) Non-holonomic.

for holonomic robots and 140 for non-holonomic ones. For holonomic robots, from 280 individuals, PCC-EE has less average leaving time than the other, while for non-holonomic robots, SQF is better from 180 individuals with statistical significance.

As the number of robots increases, the TRVF algorithm has higher average leaving time because more robots gather next to the target area, trying to go away. For non-holonomic robots, this number is lower as they demand more time to avoid other robots when moving under non-holonomic constraints.

For holonomic robots, robots using the SQF algorithm take more time to leave the target due to the curved path that they must follow caused by the rotational force field to leave the target area. By contrast, the robots with PCC-EE follow almost a straight line from the target to leave that area, as it frees regions for leaving (as seen in Figure 5.43b). However, in the non-holonomic case, the robots need to make more turns or reduce linear speed to avoid others next to the target until they reach one of those regions where they can move in straight lines. With SQF, the rotational speed variation for leaving is low, and the robots can maintain linear speed most of the time, as few robots are moving in the opposite direction.

Comparison of the Total Simulation Time. Figure 5.41 shows the comparison for the total simulation time. Also, note that the average time to leave the target is less than the time to reach it – less than 15% – justifying the low difference in the shape of the graphs for the total simulation time and the reaching time. Apart from the shifting in the values on these graphs, SQF is significantly better, regarding the simulation time, only starting from 280 holonomic robots (the same for the reaching

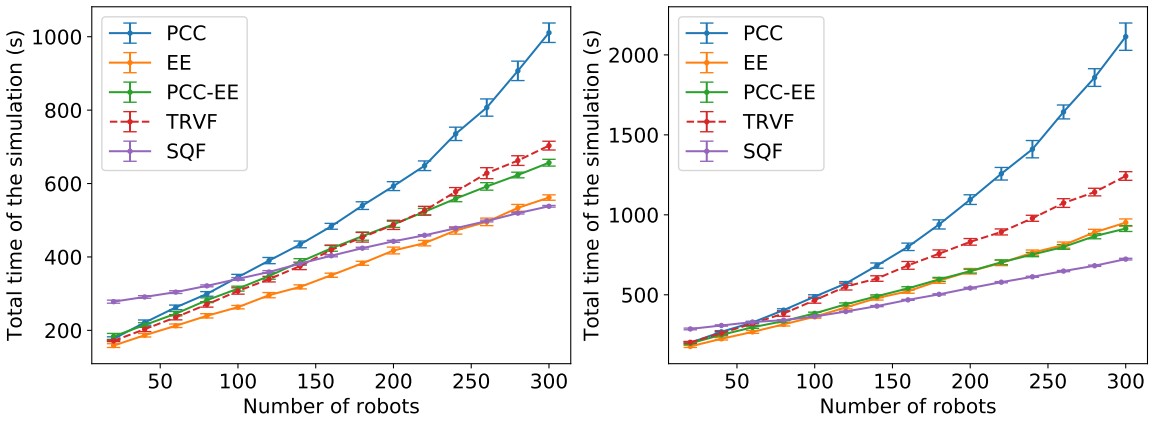


Figure 5.41: Total simulation time comparison of the algorithms for a number of robots from 20 to 300 in steps of 20. (a) Holonomic. (b) Non-holonomic.

time). Unpaired t-tests with $\rho = 0.01$ returned that the simulation time intervals of SQF and EE algorithms for 240 and 260 holonomic robots – Figure 5.41a – have the same mean. For non-holonomic robots, the total time of SQF is less than the obtained from EE for 120 robots, as occurred for the reaching time. Likewise, unpaired t-tests showed that the simulation time intervals for 100 non-holonomic robots of SQF and EE algorithms – Figure 5.41b – have the same mean (as happened for reaching time).

5.4.2.2 Comparison for Varying Target Sizes

The throughput and total simulation time of a varying target size using 100 robots are now analysed to show that the SQF algorithm outperforms the other algorithms as the target size gets smaller. Only total simulation time is shown here because the leaving time is small compared to the reaching time, as shown before. In this experiment, the TRVF algorithm was not used because the restriction the lower and upper values of K in (4.14) (Proposition 7) forbids the usage of the small values of target area radius for the influence radius used here.

First, it is shown that current algorithms fail to complete executions for small target sizes. For a given target area radius s, it was respectively used $s + 0.7$ m and $s+2.2$ m for the radius of free and danger regions to PCC, EE and PCC-EE algorithms (Marcolino et al., 2017). Figure 5.42 shows the percentage of failed simulations for different target sizes. For experiments with non-holonomic robots, PCC fails for all target sizes. EE and PCC-EE failed to terminate every run for a target size below 0.6 m, but the number of unfinished experiments decreases until the radius is less than 0.9 m. The SQF can complete all executions for the displayed target sizes. For the

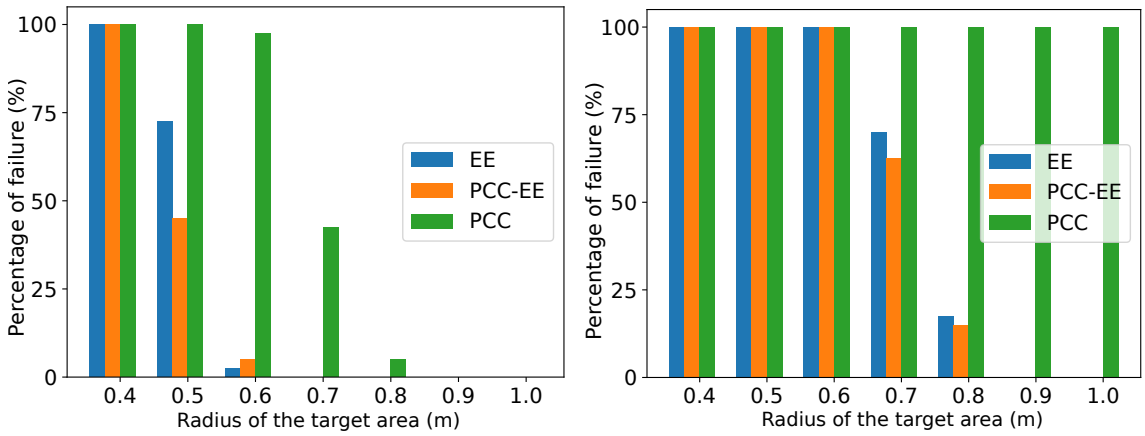
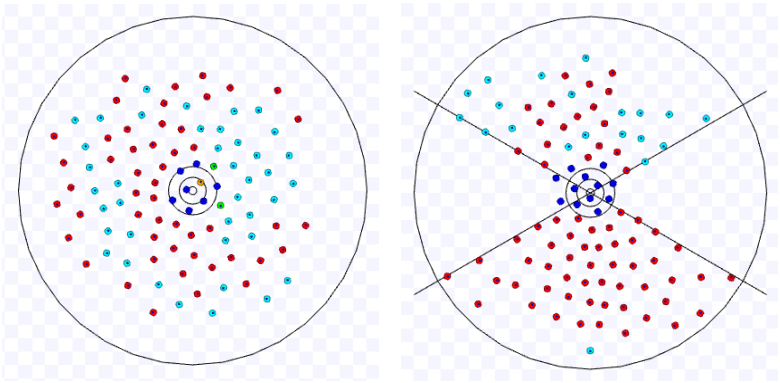


Figure 5.42: The number of runs that fail to complete in a simulation time less than 20 minutes in relation to the target size and algorithm for holonomic and non-holonomic robots. SQF had zero percentage of failure in all runs. (a) Holonomic. (b) Non-holonomic.

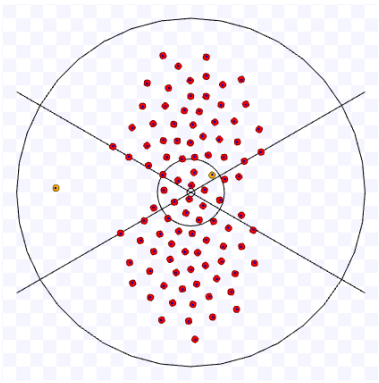
holonomic robots experiments, from $s = 0.9$ m all algorithms complete the execution within the time limit and the decrease in the number of failed executions per radius size is greater than the non-holonomic robots experiments.

When the radius of the target area is small, more robots tend to concentrate around it. The attractive force towards the target centre is not enough to counteract the repulsive forces from the nearby robots. This does not create a zero force vector but reduces the attraction to the target centre notably, and the robots slowly and erratically circle around the target. Figure 5.43 shows examples of this situation.

Figures 5.44 and 5.45 display the throughput and time to complete by the radius of the target area. The missing points are caused by failed executions, as shown above. The confidence intervals are calculated taking in account only the successful runs. As shown in Figure 5.42b, PCC failed for all radius values when non-holonomic robots are utilised, so it is not presented in Figures 5.44b and 5.45b. SQF significantly outperforms all other algorithms for any target size in both non-holonomic and holonomic cases.



(a) (b)



(c)

Figure 5.43: Executions with non-holonomic robots and $s = 0.3$ m showing situations where the robots cannot proceed within the time limit of 20 minutes. Here the colours have the same meaning as by Marcolino et al. (2017). The robots circle around the target area for too long time. (a) PCC. Available on `https://youtu.be/kLOLOENvnqU`, accessed on 20 December 2022. (b) PCC-EE. Available on `https://youtu.be/UvzSqFyXB7E`, accessed on 20 December 2022. (c) EE. Available on `https://youtu.be/BuTcsBNGCag`, accessed on 20 December 2022.

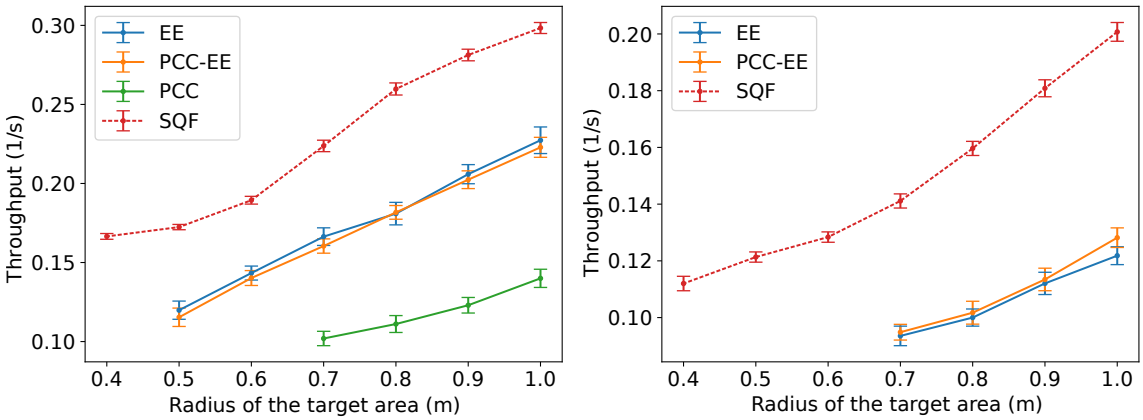


Figure 5.44: Throughput comparison of the algorithms for a varying circular target area radius. (a) Holonomic. (b) Non-holonomic.

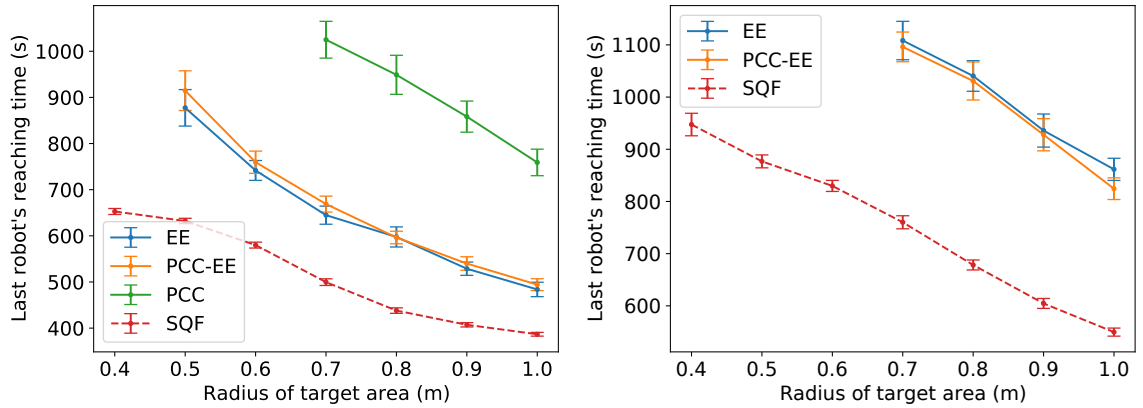


Figure 5.45: Simulation time comparison of the algorithms for a varying circular target area radius. (a) Holonomic. (b) Non-holonomic.

Chapter 6

Estimated Task completion time estimations

Besides the target area throughput, it is important to measure the task completion time, which includes not only the time to arrive at the target but also the time to leave the working area. In Chapter 5, the algorithms SQF, TRVF and the algorithms used for comparison had lower time to leave the working area than to arrive at the target, justifying the usage of target area throughput. However, there are algorithms in which the time to leave the working area affects the swarm more, such as MT and NC.

In this chapter, as before, consider the task scenario where a system with N robots in a plane must reach a common target located at $\mathbf{G}$. Upon reaching the target, the robots move to different destinations, which may not be similar. Suppose that the target area is a circle of radius s, and a robot reaches the target if its centre of mass is at a distance below or equal to the radius s from the centre of the target. In addition, there is no minimum amount of time to stay at the target.

In this scenario, the coordination algorithms, if applied, are only employed inside a working area, that is, a circle of radius $D > s$ around the target. If the robots have two or more targets, they are apart by at least $2D$. The robots are initially in a random position with a distance from the target centre in $[D, D + E)$ m for a fixed E. Once the robots have reached the target area, they will proceed to the next one, either to the left or right of the shared target. Approximately half of the robots will go to the left-hand side and the rest to the right, according to a uniform probability. The task completion time is obtained from the last robot to leave the working area, that is, the maximum time to enter the target area plus the time to leave the working area for every robot in the swarm. As the initial position of the robots is random in an experiment and influences the variation of this maximum time, it is considered the expected task completion time from various experiments.

Assume a controller based on the following equations for a robot $i \in \{1, \ldots, N\}$ at position $\mathbf{r}_i(t) = (r_{ix}(t), r_{iy}(t))$ at time t: $\dot{\mathbf{r}}_i(t) = \frac{d\mathbf{r}_i(t)}{dt}, \ddot{\mathbf{r}}_i(t) = \frac{d\dot{\mathbf{r}}_i(t)}{dt}$, $\mathbf{F_A} = \mathbf{F_A}(\mathbf{r}_i(t))$, $\mathbf{F_R} = \sum_{\substack{j=1 \\ i \neq j}}^{N} \mathbf{F_R}(\mathbf{r}_i(t), \mathbf{r}_j(t))$, and

$$\ddot{\mathbf{r}}_i(t) = K_{res} \frac{\mathbf{F_A} + \mathbf{F_R}}{\|\mathbf{F_A} + \mathbf{F_R}\|} - K_{dp} \dot{\mathbf{r}}_i(t), \tag{6.1}$$

where $\mathbf{F_A}$ depends on the algorithm used by each robot and has a fixed magnitude K_a, and $\mathbf{F_R}$ is the repulsive force applied to avoid bumping other robots. Each robot adds the attractive force with the repulsive forces and then constrain the result to a fixed vector modulus K_{res} to homogenise the acceleration among the individuals in the swarm. The constant $K_{dp} > 0$ is employed to control alterations in velocity. It prevents the robots slipping due to acceleration changes.

Also, the repulsive force is

$$\mathbf{F_R}(\mathbf{r}_i(t), \mathbf{r}_j(t)) = \begin{cases} K_r \left(\frac{1}{\|\mathbf{r}_i(t) - \mathbf{r}_j(t)\|} - \frac{1}{I_d} \right) \frac{\mathbf{r}_i(t) - \mathbf{r}_j(t)}{\|\mathbf{r}_i(t) - \mathbf{r}_j(t)\|^3}, & \text{if } \|\mathbf{r}_i(t) - \mathbf{r}_j(t)\| < I_d \\ 0, & \text{otherwise,} \end{cases}$$

where K_r is a fixed multiplicative constant for the repulsive force field, and I_d is the default influence radius, that is, the maximum distance from its mass centre a robot considers anything sensed as an obstacle to avoid (Siegwart, Nourbakhsh, and Scaramuzza, 2011).

With this $\mathbf{F_R}$, Equation 6.1 depends on the neighbourhood of the robot, which, although limited, is dynamic. Even if there is a fixed maximum number of neighbours, a global analysis of $\mathbf{r}_i(t)$ would depend on all the other $N - 1$ robots for each instant t. Thus, getting a closed-form expression for the task completion time in terms of N directly from Equation 6.1 is difficult.

Instead of deriving a function $\mathbf{r}_i(t)$, inverting it to get t for each robot i, get the maximum t for every robot i and then calculate its expected value, a theoretical macroscopic analysis inspired by simulations is performed to obtain the expected task completion time $\mathbb{E}[t]$. For doing this, besides the target area radius s and the working radius D, assume that the following variables are given: the mean distance between the centre of mass of a robot and the others $\bar{d}$ inside the influence radius I_d for all robots and the mean linear speed of all robots $\bar{v}$.

Furthermore, in the expected task completion time expressions shown below, constants are defined for each algorithm to subsume the overall behaviour of the robots, their interactions and the congestion inspired by the experiments. Such abstraction is analogous to the friction constant in Physics, which summarises the microscopical effect of surfaces. In this field of study, the friction constant for commonly used materials was initially calculated from experiments with them. Similarly, the defined constants are fitted by the experimental data in Section 6.2.

s	D	E	K_{res}	K_a	K_{dp}	K_r	I_d
3 m	13 m	8 m	2.5	2.5	10	0.5	3 m

Table 6.1: Default values for parameters used in the examples of Chapter 6 for all algorithms.

6.1 Analysis of algorithms

In this section, four algorithms are analysed about the expected task completion time. In the examples in the figures that follow and in the performed experiments, it was used the values in Table 6.1 and the robots are holonomic. In the experiments, the new targets are far from the first target region at the same x coordinate but distant by 999999 m on its left and right sides. Note, however, that the theoretical results also apply for arbitrary values and holonomic or non-holonomic robots, as shown in Section 6.2.

In the following sections, the algorithms of Chapter 5 are restated in equation form and based on the local motion controllers for showing the difficulty of solving the task completion time from the individual local control equations. I start this presentation with the no coordination (NC) (Marcolino and Chaimowicz, 2009) (as explained before, without any congestion control algorithm, only attractive potential field to the target area and repulsive potential field for avoiding other robots) as a simple example to show that even from this reduced model it is hard to infer the expected task completion time per number of robots.

6.1.1 No Coordination Algorithm

When the robots are not coordinated, they only follow the target region, avoiding the others by repulsive force. The equation for the $\mathbf{F_A}$ in Equation 6.1 is the attractive force to the target centre $\mathbf{G}$ given by

$$\mathbf{F_A}(\mathbf{r}_i(t)) = \frac{K_a(\mathbf{G} - \mathbf{r}_i(t))}{\|\mathbf{G} - \mathbf{r}_i(t)\|}. \tag{6.2}$$

Although this is the simplest attractive force of all algorithms presented here, Equation 6.1 also considers the repulsive force of all robots in the environment. Thus, as the number of robots grows, the analytical solution for the differential equations is still complicated to infer.

As an illustration, Figures 6.1 and 6.2 show an experiment of NC. Red robots are going to the target region, yellow ones have arrived at it and are trying to leave it, and black robots have exited from the working area and are going to the next target. The robots start in random positions outside the working area (Figure 6.1a). The

first robot to reach the working area takes an expected time inversely proportional to the number of individuals in the swarm (marked with a circle and an arrow in Figure 6.1b). Each robot goes toward the target area until the first reaches it . Eventually, the first robots to reach the common target area will be stuck in it and will try to leave while they are slowed down due to the congestion caused by the other robots that did not arrive at the target yet (Figure 6.1c). The robots trying to leave the target area will slowly push the other robots until they move out of the cluttered region with a shape similar to a circle. If a robot is heading in the same direction as the first to overcome this area, it may follow the space created by the other robots avoiding the first. This may cause more robots to do the same, and a queue of robots is formed (Figure 6.1d). The length of this queue and the number of robots in it are random. New queues may appear while the number of robots in the cluttered area diminishes. When no robots are trying to arrive at the target area, the last queues of robots will finally reach the outside of the working area (Figure 6.2e) until the last robot leaves this area (with a circle and an arrow in Figure 6.2f).

Figures 6.3 and 6.4 illustrate an example with a larger target area radius. The robots have more space to access the target. In this example, the capacity of the target area is greater than the number of robots, so they can use the free space to travel to the next targets in both sides (Figure 6.3c). Due to this free space, the number of robots going to the target (in red) is not enough to force the robots leaving it (in yellow) to go to the target centre (Figure 6.3d). Thus, the time until the cluttering near the target area disappears is lower than in the previous example, and how the robots leave the target area is also different (Figure 6.4e) because of the free space in the target area and the less resistance caused by the robots going to the target area.

By analysing such examples, an approximation of the expected task completion time for the NC algorithm can be obtained. The next estimation presents my result.

Estimation 1. *The estimated expected time for N robots starting at a random position with a distance from the target centre in $[D, D+E)$ to arrive at the common target area and leave the working area without coordination is*

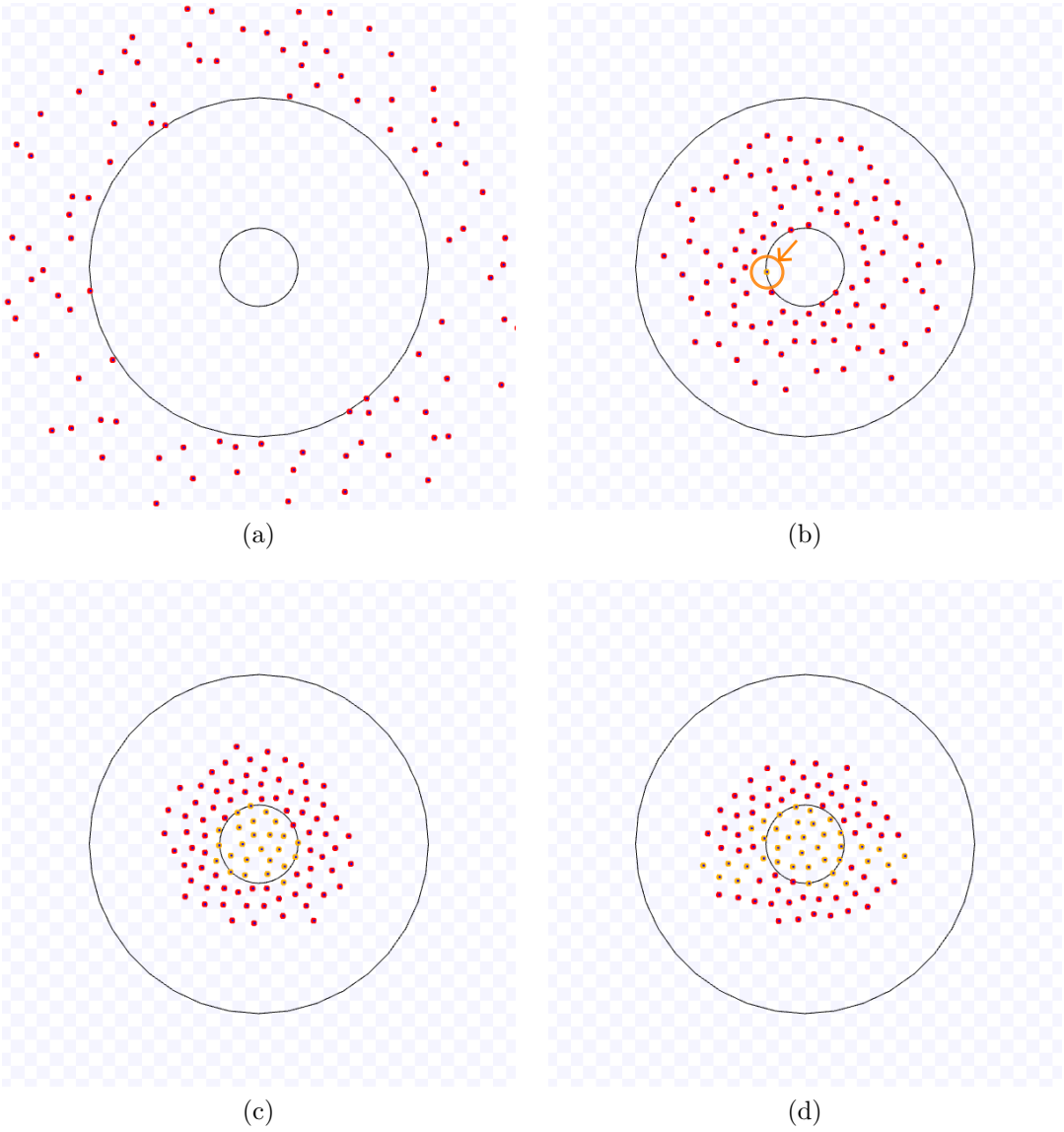
$$\mathbb{E}[t_{NC}] \approx \frac{E}{\bar{v}(N+1)} + \frac{2D-s}{\bar{v}} + wait_1(N). \tag{6.3}$$

for

$$wait_1(N) = \begin{cases} C_{NC1}N, & \text{if } N \leq \frac{2\left(s+\bar{d}/2\right)^2}{\left(\bar{d}/2\right)^2} \\ C_{NC2}\left(\frac{N^{1.5}}{1.5\sqrt{\pi}} - N\right), & \text{otherwise} \end{cases}$$

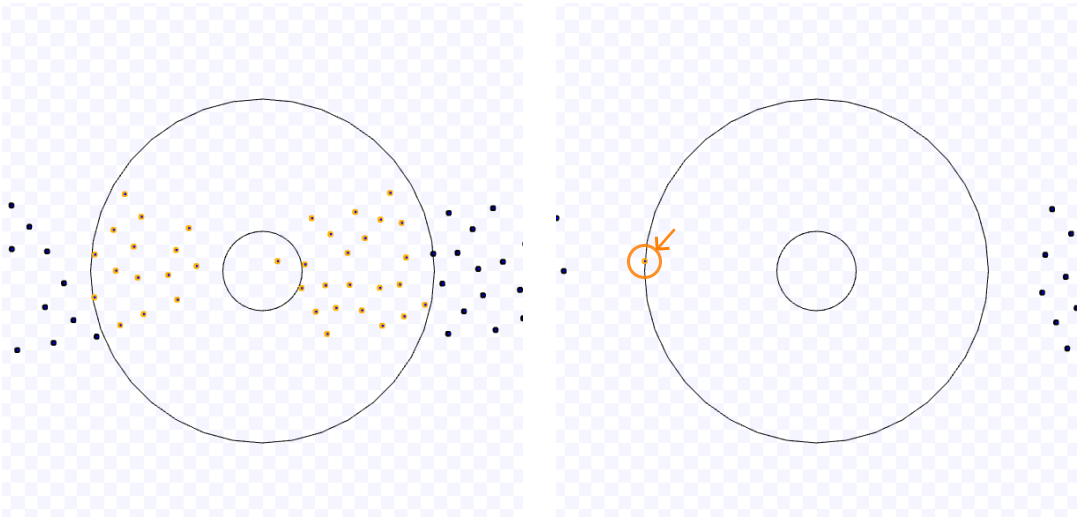
and constants C_{NC1} and C_{NC2}.

Explanation. The equation for estimating the expected time for NC has five parts: the expected time for (a) the first robot arriving at the working area from its starting



(a) (b) (c) (d)

Figure 6.1: Example of an experiment of the NC algorithm with $N = 100$ and default values. (a) Start of the experiment. (b) The first robot to reach the target area. (c) More robots reach the target region, fill this area and try to leave. (d) Queues of robots formed to leave the cluttered area. Full simulation available on `https://youtu.be/qgLxFk0OGyo`, accessed on 14 October 2022.



(e) (f)

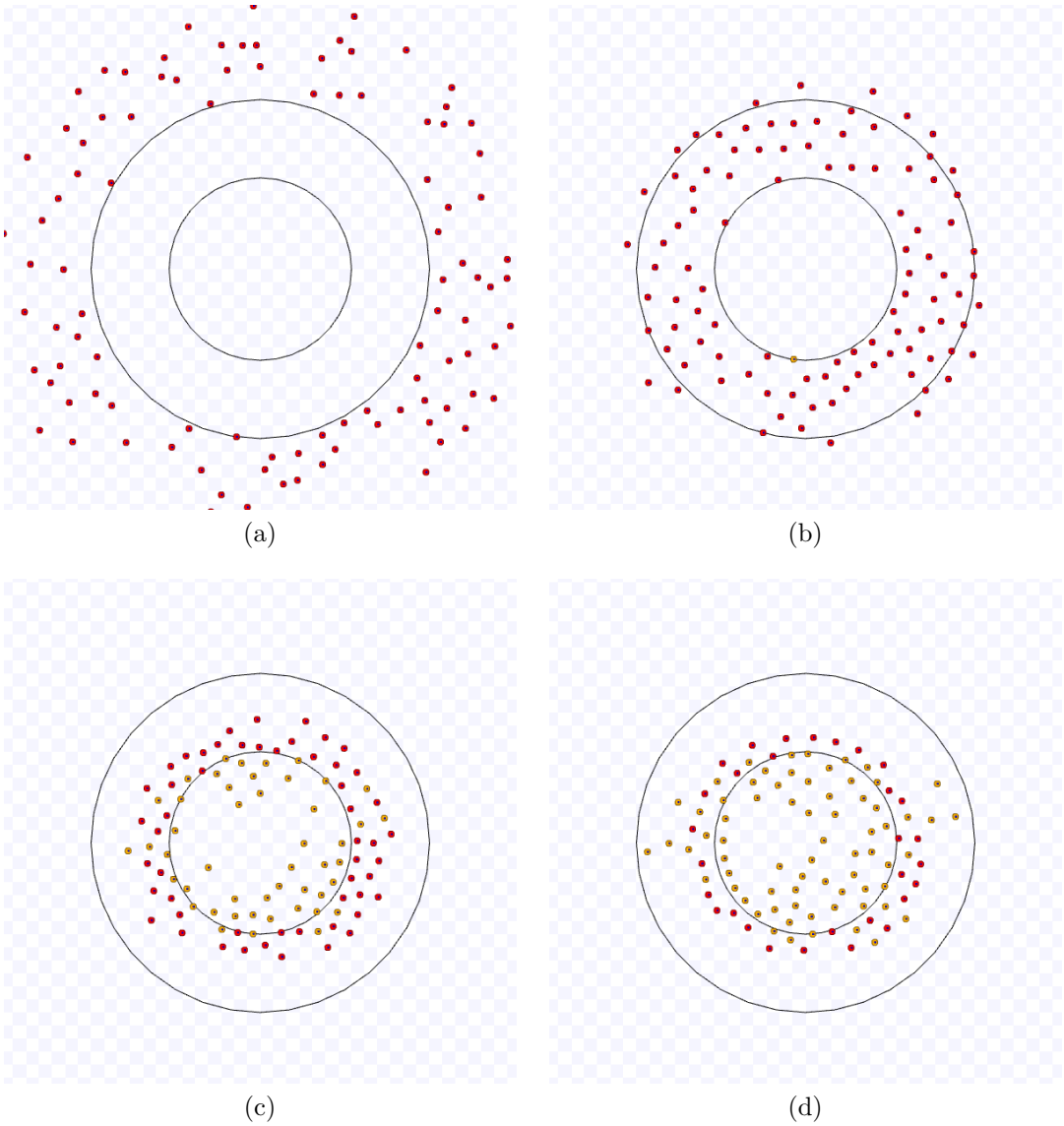
Figure 6.2: Continuation of Figure 6.1. (e) Last queues near the end of the experiment. (f) The last robot to leave the working area.

position, (b) the first robot that entered the working area reaching the target area (e.g., the time at Figure 6.1b), (c) the first robots filling the target area (e.g., the time from Figure 6.1b to 6.1c), (d) the cluttered area disappearing by the queues of robots leaving the target (e.g., the time of Figures 6.1c–6.2e and 6.3c–6.4e) and (e) the last robot that reached the target leaving the working area (e.g., the time from Figure 6.2e to 6.2f).

Part (a) is obtained from the expected value of the minimum starting distance to the working area for the N robots. Due to the task scenario description, this value uniformly ranges in $[0, E)$. Consequently, the expected minimum distance is $E/(N+1)$, obtained from the probability distribution function of the minimum value (Casella and Berger, 2002, p. 229). Then, the expected time for the first robot to arrive at the working area is $E/(\bar{v}(N+1))$.

Parts (b) and (e) share the same expression, obtained from the expected time for a robot on average speed going from the end of the working area to the target area and vice-versa: $(D-s)/\bar{v}$. Part (c) is given by the expected time for a robot going from the border of the target area to the target centre due to the congestion caused by the other robots going to the target area: $s/\bar{v}$.

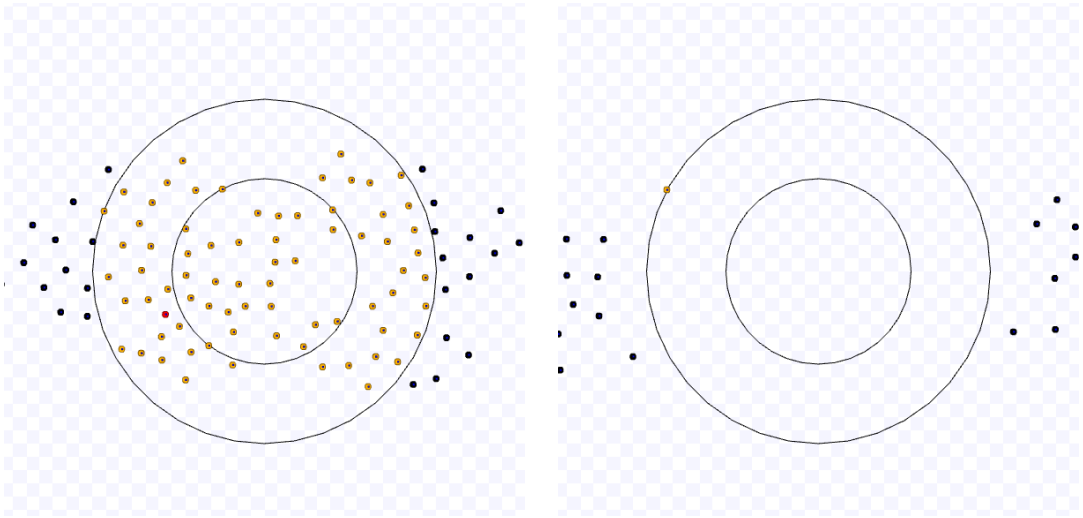
For calculating the time (d), there are two cases. When the number of robots that fit inside the target area is greater than or equal to the number of robots that do not fit, the robots which arrived at the target area have less resistance to leave through the space between the robots which did not yet. The capacity of the target



(a) (b)

(c) (d)

Figure 6.3: Example of an experiment of the NC algorithm with $N = 100$ and default values except for $s = 7$ m. (a) Start of the experiment. (b) The first robot to reach the target area. (c) Free space in the middle of the target area. (d) A few robots going to the target area and more space inside the target area. Full simulation available on `https://youtu.be/x0oO5DuAtwQ`, accessed on 12 May 2023.



(e) (f)

Figure 6.4: Continuation of Figure 6.3. (e) Robots leave the target area with less resistance. (f) End of the experiment.

area is approximately the number of circles of radius $\bar{d}/2$ – half the mean distance between the robots – inside the target area with radius s increased by $\bar{d}/2$, that is, $\frac{\left(s+\bar{d}/2\right)^2}{\left(\bar{d}/2\right)^2}$. Thus, the number of robots exceeding the capacity of the target area is less than or equal to its capacity if $N - \frac{\left(s+\bar{d}/2\right)^2}{\left(\bar{d}/2\right)^2} \leq \frac{\left(s+\bar{d}/2\right)^2}{\left(\bar{d}/2\right)^2} \Leftrightarrow N \leq \frac{2\left(s+\bar{d}/2\right)^2}{\left(\bar{d}/2\right)^2}$. From experiments, the waiting time can be approximated by $C_{NC1}N$ for a constant C_{NC1} that abstracts the influence of other factors in this time, assuming that it does not depend on time (e.g., any scaling in the speed or the average number of robots per queue, and the influence in the overall movement by the type of the robot – holonomic or non-holonomic).

If the number of robots that fit inside the target area is less than the number of robots that do not fit, the resistance for leaving the target area is higher than before, and part (d) is calculated differently. Due to the erratic formation of the queues of the robots leaving the target area, the time is approximated by assuming a fixed mean distance $\bar{d}$ between the robots and mean speed $\bar{v}$ moving from the cluttered area and taking the same time to move as the erratic queues. As seen in the example from Figure 6.5, the robots outside the target area go towards the centre, while those leaving it form queues following the other robots for each direction to overcome the congestion. The red robots going to the target tend to concentrate inside a circle containing the cluttered area. On the other hand, the yellow robots leave that area

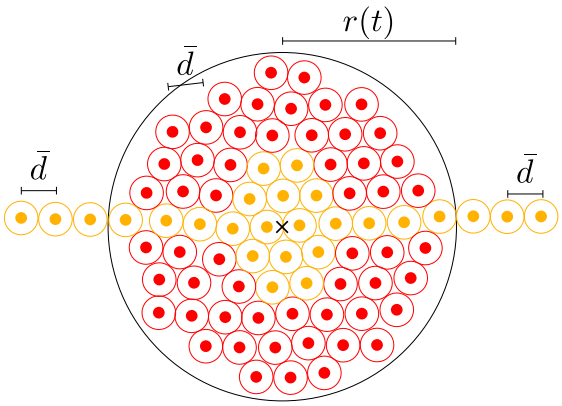


Figure 6.5: Number of robots at a given time, $N(t)$, occupying a circle of radius r with robots distant by each other by $\bar{d}$. Red and yellow dots and circles represent robots going to and leaving the target area, respectively.

one by one for each queue, diminishing the number of robots in the clutter.

Hence, the cluttered area is approximated by a circle whose radius $r(t)$ vanishes as the robots run through these queues. Let $N(t)$ be the number of robots in that circle at time t. Figure 6.5 illustrates this circle with robots. The cross mark in the middle indicates the target area centre. Suppose that a robot next to that position has to leave that circle. Then, it must run a distance of at most $r(t)$.

This radius $r(t)$ can be estimated from the area occupied by $N(t)$ robots. For this estimation, consider each robot occupying a squared area of $\bar{d}^2$ m^2. The area of the circle containing the robots is approximately the area occupied by $N(t)$ squares with area $\bar{d}^2$, that is,

$$\pi r(t)^2 \approx \bar{d}^2 N(t) \Rightarrow r(t)^2 \approx \frac{\bar{d}^2 N(t)}{\pi} \Rightarrow r(t) \approx \sqrt{\frac{\bar{d}^2 N(t)}{\pi}} \Rightarrow r(t) \approx \bar{d}\frac{\sqrt{N(t)}}{\sqrt{\pi}}.$$

A robot next to the target centre has to move through a distance equivalent to the number of robots fitting $r(t)$, excluding that one. Thus, the number of robots to move through is given by $r(t)/\bar{d} - 1$. As one robot is decreased proportionally to the number of robots that it would have to move through and the number of queues, the rate of decrease is given by

$$\frac{dN(t)}{dt} = \frac{-1}{C_{NC2}\left(\frac{r(t)}{\bar{d}} - 1\right)} = -\frac{1}{C_{NC2}\left(\frac{r(t)}{\bar{d}} - 1\right)} \approx -\frac{C_{NC2}^{-1}}{\frac{\bar{d}\frac{\sqrt{N(t)}}{\sqrt{\pi}}}{\bar{d}} - 1} = -\frac{C_{NC2}^{-1}}{\frac{\sqrt{N(t)}}{\sqrt{\pi}} - 1}.$$

C_{NC2} subsumes the effect of other factors in this rate, such as how the robots move

and variations in the speed, similar to C_{NC1} above. Hence,

$$\left(\frac{\sqrt{N(t)}}{\sqrt{\pi}} - 1\right) dN(t) \approx -C_{NC2}^{-1} dt \Leftrightarrow \int \left(\frac{\sqrt{N(t)}}{\sqrt{\pi}} - 1\right) dN(t) \approx -\int C_{NC2}^{-1} dt \Leftrightarrow$$

$$\frac{N(t)^{1.5}}{1.5\sqrt{\pi}} - N(t) + n_0 \approx -C_{NC2}^{-1} t,$$

for a constant n_0. As for $t = 0$, $N(0) = N$, $n_0 = -\frac{N^{1.5}}{1.5\sqrt{\pi}} + N$, implying that when $N(t) = 0$,

$$t \approx C_{NC2} \left(\frac{N^{1.5}}{1.5\sqrt{\pi}} - N\right).$$

The desired result is obtained by summing parts (a)-(e). □

6.1.2 Single Queue Former Algorithm

As presented before in Section 5.1, the SQF algorithm makes one queue shaped as a rectangular corridor that goes towards the target. This corridor has a width equal to the circular target diameter, $2s$, and a length equal to the working radius, D. The robots are permited to enter the target region only by this queue. Potential fields are deployed to form this queue and to guide the robots to the target area exit efficiently. The corridor starts from the current target centre $\mathbf{G} = (G_x, G_y)$. Without any loss of generality, the corridor has vertices located at $(G_x + s, G_y + D)$, $(G_x - s, G_y + D)$, $(G_x - s, G_y)$ and $(G_x + s, G_y)$.

A rotational force field is applied to robots in the working area in order to enter the common target area through the corridor. The field rotation centre is at the target centre. The robots are submitted to an attractive force towards the target once they reach the corridor. After they arrive at the target area, another rotational force field is applied to them, whose centre is either at the leftmost point of the working area, $\mathbf{P} = (P_x, P_y) = (G_x - D, G_y)$, or at the rightmost point, $\mathbf{Q} = (Q_x, Q_y) = (G_x + D, G_y)$, depending on the position of the next target.

To control the attractive forces, each robot has the states *going to the target* (GT), *leaving the target* (LT) and *going to the corridor* (GC), which respectively means the robot is going straightly to the target region, it is leaving it, and it is going to the corridor. The starting state of the robots is GT since they start outside the working area. Consider the conditions named W, O and A, represented as well-formed formulas:

$$W = \|\mathbf{r}_i(t) - \mathbf{G}\| \leq D,$$
$$O = r_{iy}(t) < G_y \vee |r_{ix}(t) - G_x| > s,$$
$$A = \exists t' \leq t : \|\mathbf{r}_i(t') - \mathbf{G}\| \leq s,$$

W	*O*	*A*	*State*
T	T	T	LT
T	T	F	GC
T	F	T	LT
T	F	F	GT
F	T	T	GT
F	T	F	GT
F	F	T	GT
F	F	F	GT, Start

Table 6.2: Truth table relating the possible values of conditions W, O and A with the states. "Start" means the condition met at $t = 0$.

where $\mathbf{r}_i(t) = (r_{ix}(t), r_{iy}(t))$ is the position of the robot i at time t. In other terms, W is true when the robot is inside the working area, O, when it is outside the corridor unlimited from above and A, when it has already arrived at the target area. Note that W and O do not depend on past time as A. Thus, the states of the robot can be represented as formulas using these conditions, which are mutually exclusive:

$$GT = \neg W \vee \neg O \wedge \neg A,$$
$$GC = W \wedge O \wedge \neg A,$$
$$LT = W \wedge A.$$

To help to check that these states are mutually exclusive, see that $GT \wedge GC$, $GT \wedge LT$ and $GC \wedge LT$ are false. Another way is by their corresponding truth values in Table 6.2.

By using these conditions, the attractive force is represented as follows. Robots outside the corridor follow a force according to Chapter 5

$$\mathbf{F_A}(\mathbf{r}_i(t)) = \begin{cases} K_a \frac{(r_{iy}(t)-G_y, -r_{ix}(t)+G_x)}{\|\mathbf{r}_i(t)-\mathbf{G}\|}, & \text{if } GC \wedge r_{ix}(t) - G_x \leq 0, \\ K_a \frac{(-r_{iy}(t)+G_y, r_{ix}(t)-G_x)}{\|\mathbf{r}_i(t)-\mathbf{G}\|}, & \text{if } GC \wedge r_{ix}(t) - G_x > 0, \end{cases} \tag{6.4}$$

where K_a is a constant for setting the force magnitude. In other words, for the left-hand side of the circular working area, a clockwise rotational field is applied, and for the right-hand side, an anti-clockwise one.

However, if the robots are in the corridor, they go towards the target area following the same attractive force equation as in Equation 6.2, that is,

$$\mathbf{F_A}(\mathbf{r}_i(t)) = \frac{K_a(\mathbf{G} - \mathbf{r}_i(t))}{\|\mathbf{G} - \mathbf{r}_i(t)\|}, \text{ if } GT. \tag{6.5}$$

Robots exiting nearby the target are constrained by another rotational field. This field also is only applied inside the circular working area:

$$\mathbf{F_A}(\mathbf{r}_i(t)) = \begin{cases} K_a \frac{(r_{iy}(t)-P_y, -r_{ix}(t)+P_x)}{\|\mathbf{r}_i(t)-\mathbf{P}\|}, & \text{if } LT \wedge G'_x \leq G_x, \\ K_a \frac{(-r_{iy}(t)+Q_y, r_{ix}(t)-Q_x)}{\|\mathbf{r}_i(t)-\mathbf{Q}\|}, & \text{if } LT \wedge G'_x > G_x, \end{cases} \tag{6.6}$$

where $\mathbf{G}' = (G'_x, G'_y)$ is the new target location. In other terms, for a robot with a new target located on the left-hand side of the previous target centre, a clockwise rotational field is applied and, on the right-hand side, an anti-clockwise one.

Finally, in addition to the default influence radius I_d, another constant is used by the robots when calculating repulsive forces: I_{min}, with $I_{min} < I_d$, the minimum influence radius allowed. For robots inside the corridor or exiting the target region, the influence radius is I_{min}. Now consider a robot outside the corridor but on the upper half of the circular working area. Let d' be the distance between the robot and a vertical line in the middle of the corridor. Its influence radius I varies in relation to d' and is set to $I = I_{min} + d'$ only for $0 \leq d' \leq I_d - I_{min}$. This range for d' guarantees that $I_{min} \leq I \leq I_d$. For the other robots, the influence radius is I_d. Thus, the influence radius and repulsive force are given by

$$I(\mathbf{r}_i(t)) = \begin{cases} I_{min}, & \text{if } W \wedge (GT \vee LT), \\ I_{min} + |r_{ix}(t) - G_x|, & \text{if } GC \wedge r_{iy}(t) > G_y \wedge |r_{ix}(t) - G_x| \\ & \quad < I_d - I_{min}, \\ I_d, & \text{otherwise}, \end{cases} \tag{6.7}$$

$$\mathbf{F_R}(\mathbf{r}_i(t), \mathbf{r}_j(t)) = \begin{cases} K_r \left(\frac{1}{\|\mathbf{r}_i(t)-\mathbf{r}_j(t)\|} - \frac{1}{I(\mathbf{r}_i(t))} \right) \frac{\mathbf{r}_i(t)-\mathbf{r}_j(t)}{\|\mathbf{r}_i(t)-\mathbf{r}_j(t)\|^3}, & \text{if } \|\mathbf{r}_i(t) - \mathbf{r}_j(t)\| \\ & \quad \leq I(\mathbf{r}_i(t)), \\ 0, & \text{otherwise.} \end{cases}$$

Observe that plugging those attractive and repulsive forces in Equations 6.4–6.7 into the movement controller (Equation 6.1) yields an intricate differential equation with cases that not only depend on the position of an individual robot for attraction but also the robots in the neighbourhood due to the repulsive force and past time. Moreover, the influence radius calculated in Equation 6.7 also has different cases, expanding the chain of related expressions to calculate the solutions of the differential equations for each robot. Due to this complexity, the inference of the SQF algorithm task completion time function from the controller equations becomes more difficult than when the robots use NC. As a result, the estimated task completion time function equation is calculated by observing the behaviour of the robots in experiments, similarly to the previous section.

As an example, Figures 6.6 and 6.7 illustrate an experiment of the SQF algorithm with the default value of its parameter $I_{min} = 1$ m. Red, green and yellow robots are in

the state *going to the target*, *going to the corridor* and *leaving the target*, respectively. Black robots have exited the working area and are going to the next target. The starting positions are shown in Figure 6.6a. The robots go in the direction of the target and change to state *going to the corridor* as they reach the working area (Figure 6.6b). The longest time for all the robots to enter the corridor is approximately the time for the bottom-most robot to go to it over the working area border (marked with a circle and an arrow in Figure 6.6c). Its time includes the waiting of all robots ahead due to those occupying the corridor while they move towards the target area. After that, all robots can access the corridor (Figure 6.6d). After the last robot reaches the target area (marked with a circle and an arrow in Figure 6.7e), all robots follow the exit force field and are on different sides depending on their next target (Figure 6.7f).

Depending on the target size, the area for leaving the target may have robots still going to the entrance corridor. In this circumstance, the robots going to the target area must wait before they reach a location without any robot leaving the working area because these robots keep passing through them. Figures 6.8 and 6.9 present an example of such a situation. Figure 6.8a illustrates the initial configuration. Figure 6.8b presents the first robot to reach the target area through the corridor (marked with a circle and an arrow). Figure 6.8c shows a robot that has arrived in the target area without going through the corridor and is leaving the target area. As the space between the target area and the working area border is small, the other robots push the robots near the target, and some may go to the target area without passing through the corridor. By the time the first robot leaves the working area (marked with a circle and an arrow in Figure 6.8d), there are robots going to the corridor in green and robots reaching the working area in red in the middle of the way induced by the potential field to leave the target area. These robots will take more time to go to the corridor than when s was smaller. Thus, they create a barrier reducing the area for the robots to leave, and the congestion also blocks the robots going to the corridor, thus, increasing their time to leave (Figure 6.9e). Figure 6.9f shows the end of this experiment. Observe that most of the robots are not in this figure. The robots on it are examples of robots that took more time to go to the corridor because they were on the leaving route when the other robots were leaving, as described above. Hence, these robots in Figure 6.9f were the last to overcome the robots exiting through the leaving area.

As before, an approximation of the expected task completion time for the SQF is presented as follows.

Estimation 2. *The estimated expected time for N robots starting at a random position with a distance from the target centre in $[D, D + E)$ to arrive at the common target area and leave the working area using the SQF algorithm is*

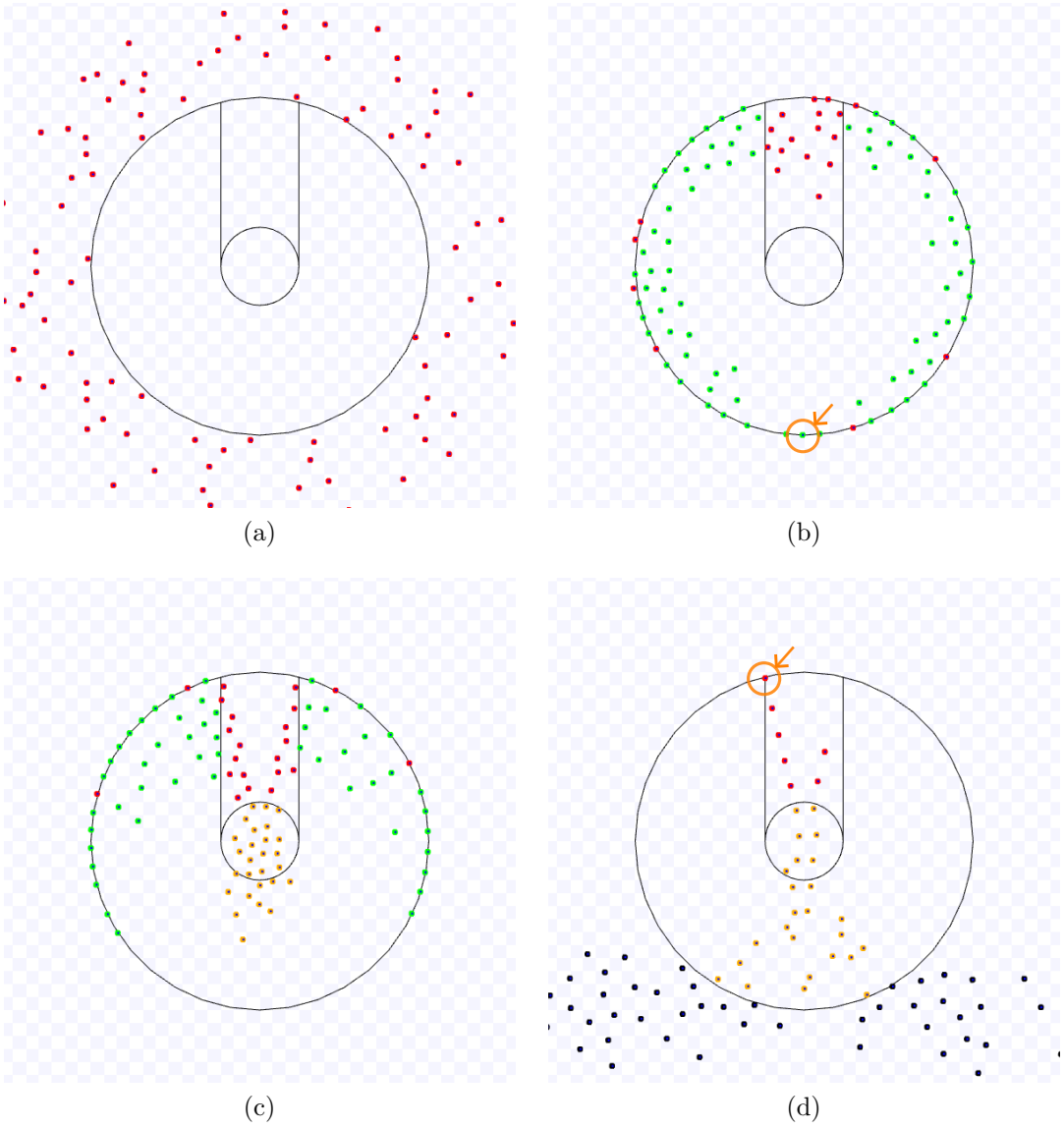
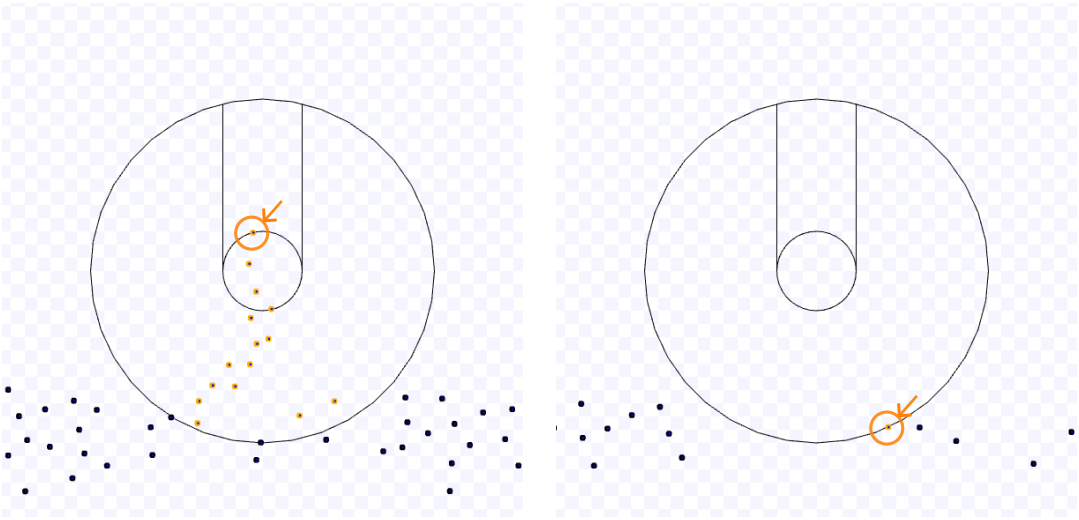


Figure 6.6: Example of an experiment of the SQF algorithm with default values and $N = 100$. (a) Start of the experiment. (b) Bottom-most robot to reach the working area. (c) Robots waiting on both sides of the corridor. (d) The last robot to reach the corridor. Full simulation available on `https://youtu.be/EdkCnUVw4Bg`, accessed on 24 March 2023.



(e) (f)

Figure 6.7: Continuation of Figure 6.6. (e) The last robot to reach the target area. (f) The last robot to leave the working area.

$$\mathbb{E}[t_{SQF}] \approx \frac{\sqrt{D^2 - s^2}}{\bar{v}} \left(\arcsin\left(\frac{s}{D}\right) + \arccos\left(\frac{\sqrt{D^2 - s^2}}{2D}\right)\right) + \frac{D - s}{\bar{v}} + \frac{N(\pi - \arcsin(s/D))D}{(N+2)\bar{v}} + \frac{NE}{\bar{v}(N+1)} + wait_2(N), \tag{6.8}$$

for

$$wait_2(N) = \begin{cases} C_{SQF1}N, & \text{if } N \leq \frac{1}{2d^2}\left(2D^2 \arccos\left(\frac{s}{D}\right) + 2s^2 + 2(D-s)^2 \arccos\left(\frac{D-s}{2D}\right)\right. \\ & \quad \left. + 4D^2 \arcsin\left(\frac{D-s}{2D}\right) - (D-s)\sqrt{4D^2 - (D-s)^2}\right), \\ C_{SQF2}N^2, & \text{otherwise} \end{cases}$$

and constants C_{SQF1} *and* C_{SQF2}.

Explanation. The estimation of the expected time to complete the SQF algorithm is better explained by analysing the experiment backwardly from the last robot to complete the task. Its calculation has five parts: the expected time for (a) the last robot leaving the working area from the target area (e.g., the time from Figure 6.7e to 6.7f), (b) the last robot going from the topmost edge of the corridor to the target area (e.g., from Figure 6.6d to 6.7e), (c) robots waiting to access the corridor (e.g., Figures 6.6c, 6.8d and 6.9e), (d) the bottom-most robot travelling from where it first

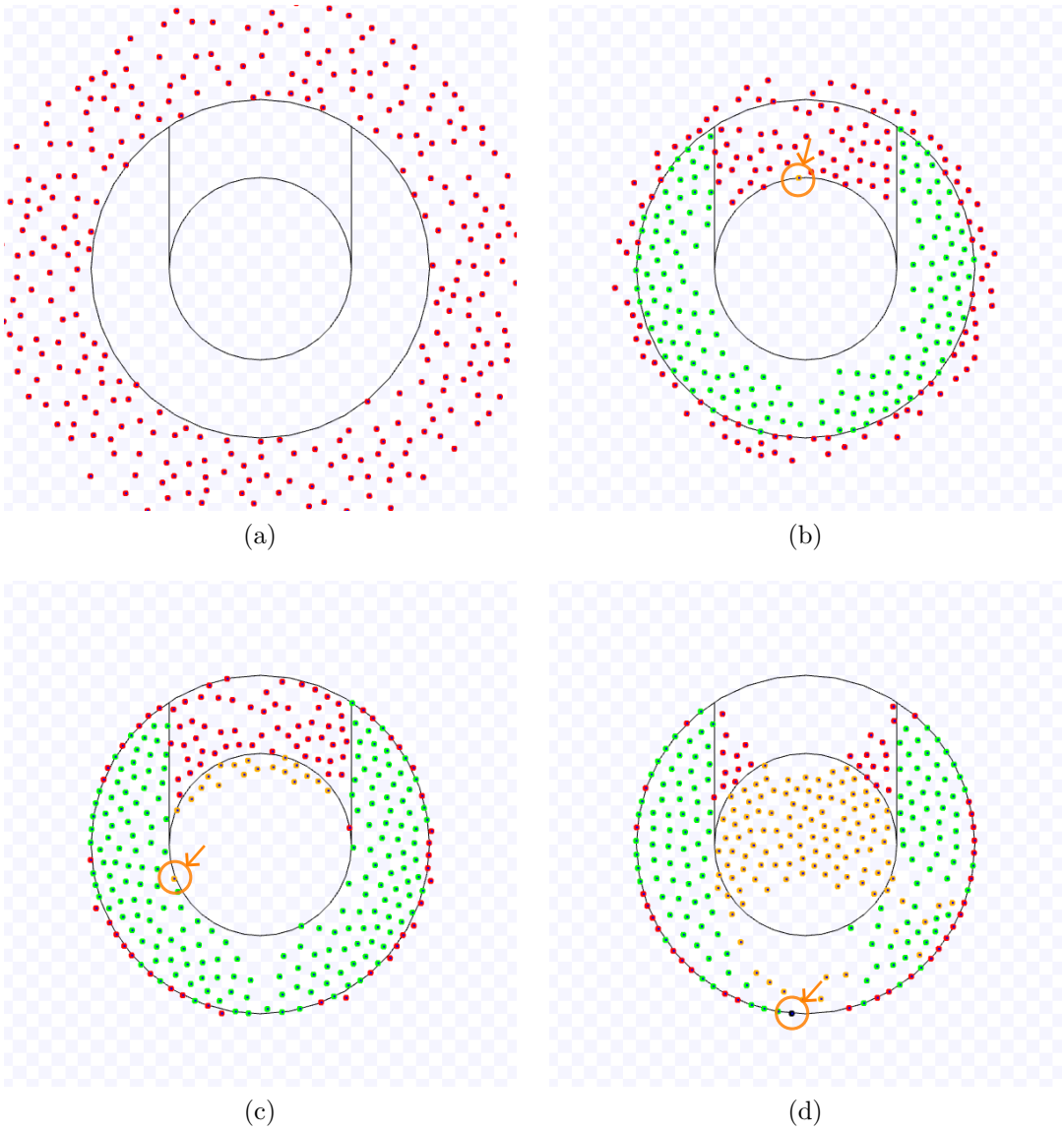
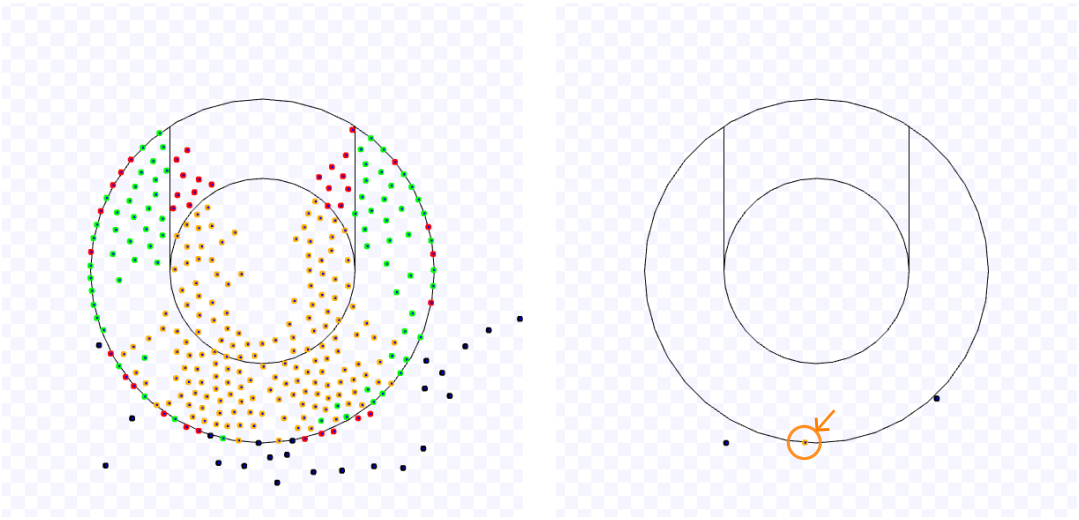


Figure 6.8: Example of an experiment of the SQF algorithm with default values except for $s = 7$ m and $N = 300$. (a) Start of the experiment. (b) The first robot to enter the target area by going to the corridor. (c) The first robot to leave the target area without passing through the corridor. (d) The first robot to leave the working area. Full simulation available on `https://youtu.be/pEQzg0JwPbQ`, accessed on 19 April 2023.



(e) (f)

Figure 6.9: Continuation of Figure 6.8. (e) Robots create a barrier obstructing others from leaving the working area. (f) The last robot to leave the working area.

entered the working area border to the corridor (e.g., the time from Figure 6.6b to 6.6d but excluding the waiting time (c)), (e) the bottom-most robot going from its starting position to the working area (e.g., Figure 6.6a–6.6b).

Part (a) is the time from the expected location in the target area reached by the last robot to the border of the working area by following a circular arc due to the exit force field. On average, the starting position of this arc is at the point A_1 in the circular target area and ends at A_2 in Figure 6.10a, which shows this arc in dashed line. This arc belongs to a circle with radius $\sqrt{D^2 - s^2}$ as illustrated by the right triangle formed by the points A_1 (at the target area circle with radius s by straightly following the corridor corner A_4 in Figure 6.10b), the target centre located at $\mathbf{G}$ and the point at $\mathbf{P}$[1]. Its length is proportional to the angle $\alpha_1 + \alpha_2$, with $\alpha_1 = \arcsin(s/D)$ in the right triangle and $\alpha_2 = \arccos(\sqrt{D^2 - s^2}/(2D))$ in the isosceles triangle $\mathbf{GP}A_2$. Thus, the expected time (a) is given by $\sqrt{D^2 - s^2}(\arcsin(s/D) + \arccos(\sqrt{D^2 - s^2}/(2D)))/\bar{v}$.

Part (b) is given by the time to move from the topmost edge (see point A_4 in Figure 6.10b) to the target area, that is, $(D - s)/\bar{v}$.

Part (c) depends on the target area size and the number of robots. If the number of robots is greater than the entry area capacity, there will be robots going to the corridor in the leaving area while others are leaving the working area, causing congestion (as in Figure 6.8d–6.9f). The entry area capacity is calculated from the part of the working

[1]In this work, bold is for points located at a previously defined position vector and non-bold for those not associated with such a vector.

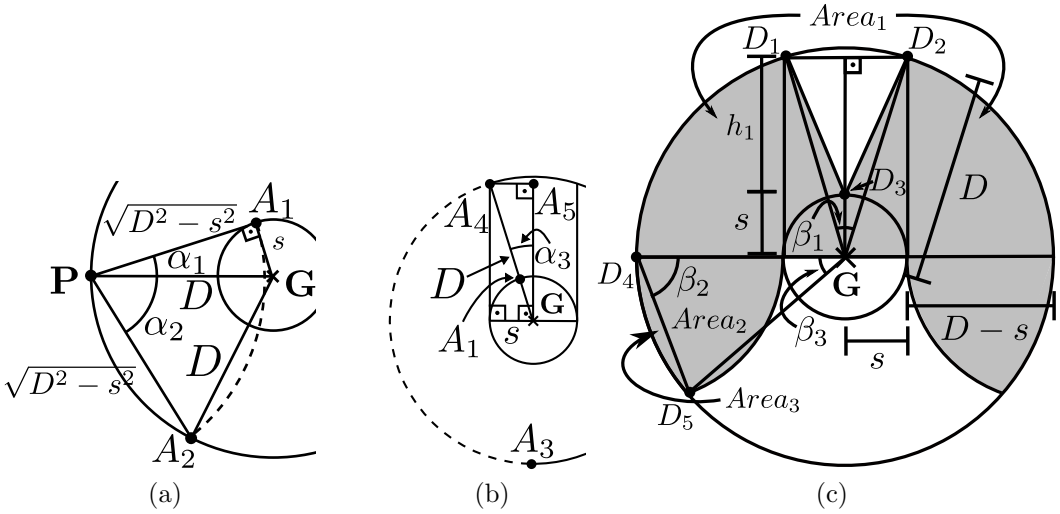


Figure 6.10: (a) Dashed arc length from the expected target arriving location at A_1 to the working area border at A_2. (b) Dashed arc length from the expected position of the bottom-most robot A_3 to the corridor corner A_4. (c) Area considered for entry capacity for the SQF algorithm.

area likely to be occupied by robots going to the target area and unused for leaving. Thus, it includes the part of the upper half working area that the robots are probable to occupy inside the corridor, including half of the target area, but excluding the area used only by the robots that first arrived in the corridor (for instance, in Figures 6.8d and 6.9e). The entry area also includes two lateral areas where robots leaving the target area are unlikely to go, given the SQF force field for leaving from the left- and rightmost points of the target area. The entry area is shaded in Figure 6.10c. For the calculation of the entry area, three areas are considered – namely, $Area_1$, $Area_2$ and $Area_3$ – defined in the following paragraphs. Due to symmetry, $Area_2$ and $Area_3$ are doubled for the final outcome.

$Area_1$ is the working area upper half minus the area of $\triangle D_1D_2D_3$ and the area of the circular segment of points D_1 and D_2. For the area of $\triangle D_1D_2D_3$, the height h_1 is given by the height of the isosceles triangle $\mathbf{G}D_1D_2$ – which is $\sqrt{D^2 - s^2}$ from its inner right triangle – minus the radius s. Thus, $h_1 = \sqrt{D^2 - s^2} - s$, and the area of $\triangle D_1D_2D_3$ is $s(\sqrt{D^2 - s^2} - s)$. For the circular segment area, β_1 is obtained from the isosceles triangle $\mathbf{G}D_1D_2$ (with equal sides measuring D and base side, $2s$). Thus, $\beta_1 = 2\arcsin(s/D)$, and the circular segment area is obtained by subtracting the area of $\triangle \mathbf{G}D_1D_2$ from the working area sector of β_1: $\frac{1}{2}D^2\beta_1 - \frac{1}{2}2s\sqrt{D^2 - s^2} = D^2\arcsin(s/D) - s\sqrt{D^2 - s^2}$. Hence, $Area_1 = \frac{\pi}{2}D^2 - s(\sqrt{D^2 - s^2} - s) - D^2\arcsin(s/D) + s\sqrt{D^2 - s^2} = D^2\left(\frac{\pi}{2} - \arcsin\left(\frac{s}{D}\right)\right) + s^2 =$

$D^2 \arccos\left(\frac{s}{D}\right) + s^2$.

$Area_2$ and $Area_3$ comprise the area on the left-hand side where robots leaving the target area are almost unlikely to pass, given the leaving force field from the leftmost point of the target area. $Area_2$ is a circular sector area of angle β_2, and $Area_3$ is the circular segment area of points D_4 and D_5 with angle β_3. β_2 and β_3 are obtained by the isosceles triangle $\mathbf{G}D_4D_5$ of equal sides measuring D and base side, $D - s$, obtained from the radius of the circular sector going from the leftmost point of the circular target area to the point D_5. Thus, $\beta_2 = \arccos\left(\frac{D-s}{2D}\right)$ and $\beta_3 = 2\arcsin\left(\frac{D-s}{2D}\right)$. Consequently, $Area_2 = \frac{1}{2}(D-s)^2\beta_2 = \frac{1}{2}(D-s)^2 \arccos\left(\frac{D-s}{2D}\right)$, and $Area_3$ is the circular sector area of angle β_3 minus the area of the isosceles triangle $\mathbf{G}D_4D_5$, that is, $Area_3 = D^2 \arcsin\left(\frac{D-s}{2D}\right) - \frac{D-s}{4}\sqrt{4D^2 - (D-s)^2}$. The entry capacity is approximated by considering each robot occupying a squared area of $\bar{d}^2$ m^2 inside the entry area: $\frac{1}{\bar{d}^2}(Area_1 + 2(Area_2 + Area_3)) = \frac{1}{2\bar{d}^2}\left(2D^2 \arccos\left(\frac{s}{D}\right) + 2s^2 + 2(D-s)^2 \arccos\left(\frac{D-s}{2D}\right) + 4D^2 \arcsin\left(\frac{D-s}{2D}\right) - (D-s)\sqrt{4D^2 - (D-s)^2}\right)$.

If the number of robots is less than or equal to the entry capacity, experiments show that the waiting time is linear in relation to the number of robots; otherwise, it is quadratic (due to the interference of the robots going to the corridor outside the entry area to the robots leaving the working area). Therefore, in the former case, part (c) is estimated by $C_{SQF1}N$, and, in the latter case, by $C_{SQF2}N^2$, for constants C_{SQF1} and C_{SQF2}. As before, these constants abstract characteristics intrinsic to the robot dynamics and the environment where the algorithm is applied.

Part (d) is calculated from the arc length from the estimated initial position of the last robot to the corridor topmost edge (point A_4 in Figure 6.10b). In Figure 6.10b, the point A_3 represents the maximum possible initial position. The estimated arc angle is the expected maximum from approximately $N/2$ robots on the left-hand side uniformly sampled over $(0, \pi - \alpha_3]$, i.e., $\frac{\frac{N}{2}(\pi-\alpha_3)}{\frac{N}{2}+1} = \frac{N(\pi-\alpha_3)}{N+2}$ (Casella and Berger, 2002, p. 229). From the right triangle formed by A_4, A_5 and $\mathbf{G}$ in Figure 6.10b, $\alpha_3 = \arcsin(s/D)$. Thus, the estimated time of part (d) is $\left(\frac{N(\pi-\arcsin(s/D))}{N+2}\right) D/\bar{v}$.

Part (e) is also obtained from the expected maximum but for N robots and the uniformly distributed distance from the starting point to the working area. It is given by $\frac{NE}{\bar{v}(N+1)}$. The desired result is obtained by summing parts (a)-(e). □

6.1.3 Touch and Run Vector Fields Algorithm

As presented before in Section 5.2, the TRVF algorithm creates K lanes around the target area by using potential fields. Each lane goes from the working area to next to the target area through an *entrance path* in a straight fashion, makes a circular curve intersecting only one point of the target area through a circular *curved path*, and then leaves the working area directly through an *exit path*. An entrance path is parallel to

the exit path on the opposing lane, and both are apart from d meters. Between them, a parallel ray passes from the target centre to the working area border. There are K such rays that divide the target area into sectors with angle $\alpha = 2\pi/K$. Each lane is contained in such sectors called the central angle region.

Consider a central angle region $z = \left\lfloor \frac{\eta}{\alpha} \right\rfloor + 1$, such that η is the angle of the vector $\mathbf{r}_i(t) - \mathbf{G}$ with the x-axis. Each lane is formed by four waypoints: $\mathbf{w_1}$ and $\mathbf{w_2}$ forms the entrance path, $\mathbf{w_2}$ and $\mathbf{w_3}$, the circular curved path, and $\mathbf{w_3}$ and $\mathbf{w_4}$, the exit path. From Section 5.2,

$$\mathbf{w_1} = \mathbf{G} + D(\cos(z\alpha), \sin(z\alpha)) + \frac{d}{2}(\sin(z\alpha), -\cos(z\alpha));$$

$$\mathbf{w_2} = \mathbf{G} + \sqrt{(r+s)^2 - (r+d/2)^2}(\cos(z\alpha), \sin(z\alpha)) + \frac{d}{2}(\sin(z\alpha), -\cos(z\alpha));$$

$$\mathbf{w_3} = \mathbf{G} + \sqrt{(r+s)^2 - (r+d/2)^2}(\cos((z-1)\alpha), \sin((z-1)\alpha)) + \frac{d}{2}(\sin((z-1)\alpha), -\cos((z-1)\alpha)); \text{ and}$$

$$\mathbf{w_4} = \mathbf{G} + D(\cos((z-1)\alpha), \sin((z-1)\alpha)) + \frac{d}{2}(\sin((z-1)\alpha), -\cos((z-1)\alpha))$$

for the radius r of the circular path from $\mathbf{w_2}$ to $\mathbf{w_3}$. Due to Lemma 3, $r = \frac{s\sin(\alpha/2) - d/2}{1 - \sin(\alpha/2)}$. This path has centre located at $\mathbf{c} = \mathbf{G} + (r+s)\left(\cos\left(\left(z - \frac{1}{2}\right)\alpha\right), \sin\left(\left(z - \frac{1}{2}\right)\alpha\right)\right)$ (Section 5.2).

The robot follows a straight line force field from the entrance and exit paths and an anti-clockwise orbit guided by the force field for the curved path. Let $\mathbf{w_i}$ and $\mathbf{w_f}$ be two arbitrary initial and final waypoints, and $\mathbf{C}$ and R any centre and radius to follow an orbit. The force field for the straight line following is expressed by (adapted from Nelson et al. (2006))

$$\xi_l = \begin{cases} \xi_f - \frac{\pi}{2}\rho, & \text{if } \epsilon > \frac{d}{5}, \\ \xi_f - \left(\frac{k_s \frac{\pi}{2} v}{k_\omega \left(\frac{d}{5}\right)^{k_s}}\right) \epsilon^{k_s - 1} \sin(\xi) - \frac{\pi}{2}\left(\frac{\epsilon}{\frac{d}{5}}\right)^{k_s}, & \text{otherwise,} \end{cases}$$

$$\mathbf{F_L}(\mathbf{r}_i(t), \mathbf{w_i}, \mathbf{w_f}) = \begin{cases} K_a(\cos(\xi_l), \sin(\xi_l)), & \text{if } \frac{(\mathbf{r}_i(t) - \mathbf{w_i}) \cdot \mathbf{w_{fi}}}{\|\mathbf{w_{fi}}\|^2} \geq 1, \\ 0, & \text{otherwise,} \end{cases} \tag{6.9}$$

for $\mathbf{w_{fi}} = \mathbf{w_f} - \mathbf{w_i}$, $\xi_f = \text{atan2}(w_{fi,y}, w_{fi,x})$, $\epsilon = \left\| \mathbf{r}_i(t) - \mathbf{w_i} - \frac{(\mathbf{r}_i(t) - \mathbf{w_i}) \cdot \mathbf{w_{fi}}}{\|\mathbf{w_{fi}}\|^2} \mathbf{w_{fi}} \right\|$, $\rho = sign(\mathbf{w_{fi}} \times (\mathbf{r}_i(t) - \mathbf{w_i}))$, a constant for setting the force magnitude K_a, the current

orientation of the robot ξ, the constant for proportional angular speed controller k_ω, the constant for exponentiation in the vector field calculation $k_s > 1$ and the maximum linear speed v.

The force field for the orbit following is given by (adapted from Nelson et al. (2006))

$$\xi_c = \begin{cases} \gamma - \frac{5\pi}{6} + \frac{v}{\|\mathbf{q}\|}\sin(\xi - \gamma), & \text{if } \|\mathbf{q}\| > 2R, \\ \gamma - \frac{\pi}{2} - \frac{\pi}{3}\left(\frac{\|\mathbf{q}\|-R}{R}\right)^{k_o} - \frac{v}{k_\omega\|\mathbf{q}\|}\sin(\xi - \gamma) & \\ -\frac{k_o v\pi}{3R^{k_o}k_\omega}(\|\mathbf{q}\| - R)^{k_o-1}\cos(\xi - \gamma), & \text{otherwise,} \end{cases}$$

$$\mathbf{F_O}(\mathbf{r}_i(t), \mathbf{C}, R, \mathbf{w_f}) = \begin{cases} K_a(\cos(\frac{\pi}{2} - \xi_c), \sin(\frac{\pi}{2} - \xi_c)), & \text{if } \mathbf{q} \times (\mathbf{w_f} - \mathbf{C}) \leq 0, \\ 0, & \text{otherwise,} \end{cases} \tag{6.10}$$

for $\mathbf{q} = (q_x, q_y) = \mathbf{r}_i(t) - \mathbf{C}$, $\gamma = \text{atan2}(q_x, q_y)$ and the constant for exponentiation in this vector field calculation $k_o > 1$.

In addition, the robots have six states to indicate in which part of the lane they are: *going to the target* (GT), *going to the entrance path* (GEP), *on the entrance path* (OEN), *entering through the curve path* (ECP), *leaving through the curve path* (LCP) and *on the exit path* (OEX). The initial state is *going to the target* (GT).

Consider the conditions named W, A, B, EN and TA, represented as well-formed formulas:

$$W = \|\mathbf{r}_i(t) - \mathbf{G}\| \leq D,$$
$$A = \exists t' \leq t : \|\mathbf{r}_i(t') - \mathbf{G}\| \leq s,$$
$$B = \exists t' \leq t : (\mathbf{r}_i(t') - \mathbf{G}) \times (\mathbf{w_1} - \mathbf{G}) \leq 0,$$
$$EN = \exists t' \leq t : \frac{(\mathbf{r}_i(t') - \mathbf{w_1}) \cdot (\mathbf{w_2} - \mathbf{w_1})}{\|\mathbf{w_2} - \mathbf{w_1}\|^2} \geq 1,$$
$$TA = \exists t' \geq t_A : (\mathbf{r}_i(t') - \mathbf{c}) \times (\mathbf{w_3} - \mathbf{c}) \leq 0.$$

W and A means the same as in SQF in the previous section. B, EN and TA respectively means in some moment the robot left the entering orbit towards the entrance path, left the entrance path and left the curved path orbit next to target area. t_A is the time that robot i arrived at target area, i.e., $t_A = \min_t \|\mathbf{r}_i(t) - \mathbf{G}\| \leq s$.

As in the previous section, the states can be expressed as mutually exclusive formulas using these conditions:

$$GT = \neg W$$
$$GEP = W \wedge \neg A \wedge \neg B \wedge \neg EN \wedge \neg TA,$$
$$OEN = W \wedge \neg A \wedge B \wedge \neg EN \wedge \neg TA,$$
$$ECP = W \wedge \neg A \wedge B \wedge EN \wedge \neg TA,$$
$$LCP = W \wedge A \wedge B \wedge EN \wedge \neg TA,$$
$$OEX = W \wedge A \wedge B \wedge EN \wedge TA.$$

Using them and Equations 6.9 and 6.10 the attractive forces of the TRVF algorithm are expressed as follows. Robots in GT are attracted to the current target by a force $\mathbf{F_{A1}}$ but are repelled by a $\mathbf{F_{TR}}$ from the working area of the previous target position $\mathbf{G'}$ (zero, if there is no previous target). The sum of these forces is normalised so that the resultant force has magnitude K_a:

$$\mathbf{F_{A1}}(\mathbf{r}_i(t)) = \frac{K_a(\mathbf{G} - \mathbf{r}_i(t))}{\|\mathbf{G} - \mathbf{r}_i(t)\|},$$

$$\mathbf{F_{TR}}(\mathbf{r}_i(t)) = \begin{cases} K_r \left(\frac{1}{\|\mathbf{G'} - \mathbf{r}_i(t)\|} - \frac{1}{D}\right) \frac{\mathbf{G'} - \mathbf{r}_i(t)}{\|\mathbf{G'} - \mathbf{r}_i(t)\|^3}, & \text{if } \|\mathbf{G'} - \mathbf{r}_i(t)\| - D \leq D, \\ 0, & \text{otherwise,} \end{cases}$$

$$\mathbf{F_A}(\mathbf{r}_i(t)) = K_a \frac{\mathbf{F_{A1}}(\mathbf{r}_i(t)) + \mathbf{F_{TR}}(\mathbf{r}_i(t))}{\|\mathbf{F_{A1}}(\mathbf{r}_i(t)) + \mathbf{F_{TR}}(\mathbf{r}_i(t))\|}, \text{ if } GT. \tag{6.11}$$

The robots in OEN and OEX only follow a straight line force field in $\mathbf{F_L}$ with different parameters:

$$\mathbf{F_A}(\mathbf{r}_i(t)) = \begin{cases} \mathbf{F_L}(\mathbf{r}_i(t), \mathbf{w_1}, \mathbf{w_2}), & \text{if } OEN, \\ \mathbf{F_L}(\mathbf{r}_i(t), \mathbf{w_3}, \mathbf{w_4}), & \text{if } OEX. \end{cases} \tag{6.12}$$

Robots in GEP are only guided by an orbital force field $\mathbf{F_O}$ with the following parameters:

$$\mathbf{F_A}(\mathbf{r}_i(t)) = \mathbf{F_O}(\mathbf{r}_i(t), \mathbf{G}, D, \mathbf{w_1}), \text{ if } GEP. \tag{6.13}$$

In ECP and LCP, they are submitted to a sum of an orbital force and a stronger attractive force, and the resultant modulus of this sum is constrained to K_a:

$$\mathbf{F_{A2}}(\mathbf{r}_i(t)) = 1.5 \frac{K_a(\mathbf{G} - \mathbf{r}_i(t))}{\|\mathbf{G} - \mathbf{r}_i(t)\|},$$

$$\mathbf{F_{A3}}(\mathbf{r}_i(t)) = 1.5 \frac{K_a(\mathbf{w_3} - \mathbf{r}_i(t))}{\|\mathbf{w_3} - \mathbf{r}_i(t)\|},$$

$$\mathbf{F_A}(\mathbf{r}_i(t)) = \begin{cases} K_a \frac{\mathbf{F_{A2}}(\mathbf{r}_i(t)) + \mathbf{F_O}(\mathbf{r}_i(t), \mathbf{c}, r, \mathbf{w_3})}{\|\mathbf{F_{A2}}(\mathbf{r}_i(t)) + \mathbf{F_O}(\mathbf{r}_i(t), \mathbf{c}, r, \mathbf{w_3})\|}, & \text{if } ECP, \\ K_a \frac{\mathbf{F_{A3}}(\mathbf{r}_i(t)) + \mathbf{F_O}(\mathbf{r}_i(t), \mathbf{c}, r, \mathbf{w_3})}{\|\mathbf{F_{A3}}(\mathbf{r}_i(t)) + \mathbf{F_O}(\mathbf{r}_i(t), \mathbf{c}, r, \mathbf{w_3})\|}, & \text{if } LCP. \end{cases} \tag{6.14}$$

Again, the controller in Equation 6.1 with the attractive force described by Equations 6.11–6.14 results in a complicated differential equation with mutual dependence. Consequently, the estimated expected task completion time function equation for the TRVF algorithm is also calculated by observing the experiments.

As an example, Figures 6.11 and 6.12 present an execution of the TRVF algorithm with its default values (Table 6.3). Robots in red, cyan, blue, magenta, yellow, orange and black represent the states GT, GEP, OEN, ECP, LCP, OEX and that they left the previous target area, respectively. Figure 6.11a illustrates the

K	d	v	k_ω	k_s	k_o
5	3 m	1 m/s	3	1.1	1.1

Table 6.3: Default values for the parameters used in the examples of Chapter 6 for the TRVF algorithm.

starting configuration. Robots go to the entrance path conducted by an orbital force around the working area. Then, they proceed to the target through the entrance path (Figure 6.11b). Figure 6.11c shows the first robot to reach the target in this experiment (marked with a circle and an arrow). Depending on the number of robots, some robots still have to wait to enter the entrance path (in cyan) while the first one to leave the working area exited (the black robot with a circle and an arrow in the bottom of Figure 6.11d). After that wait, the last robot to get into the working area proceeds through the entrance path in the lane, arrives in the target area (with a circle and an arrow in Figure 6.12e) and leave by the exit path, facing a few robots going to different directions (Figure 6.12f).

From such examples, an approximation of the expected task completion time for the TRVF is shown below.

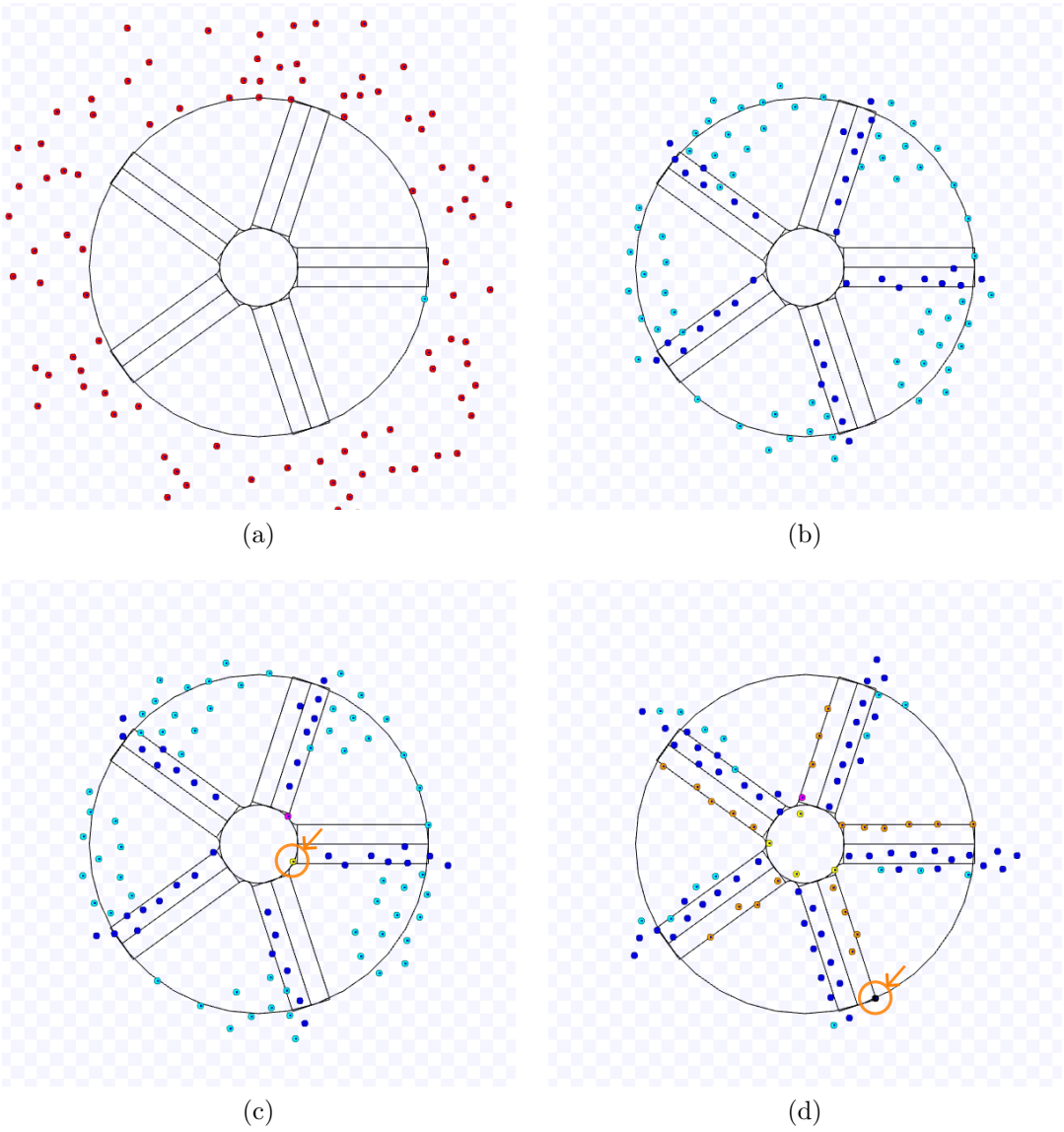
Estimation 3. *The estimated expected time for N robots starting at a random position with a distance from the target centre in $[D, D+E)$ to arrive at the common target area and leave the working area using the TRVF algorithm is*

$$\mathbb{E}[t_{TRVF}] \approx \frac{E}{\left(\frac{\arctan(\frac{d}{2D})}{2\pi}N+1\right)\bar{v}} + \frac{2d_s + r(\pi-\alpha)}{\bar{v}} + C_{TRVF}\frac{N}{K}, \tag{6.15}$$

for a constant C_{TRVF}, $\alpha = 2\pi/K$, $r = \frac{s\sin(\alpha/2)-d/2}{1-\sin(\alpha/2)}$ and $d_s = \sqrt{D^2-d^2/4} - \sqrt{s(2r+s)-d\left(r+\frac{d}{4}\right)}$.

Explanation. In order to estimate the expected time to complete the task with the TRVF algorithm, three parts are needed: the expected time for (a) the first robot reaching an entrance path without moving across the border of the working area circular sector from its starting point, (b) the first robot going to the target area and then leaving it until reaching the working area border by one of the K lanes (e.g., Figures 6.11b–6.11d), (c) the waiting and travel of the last robot through the lane.

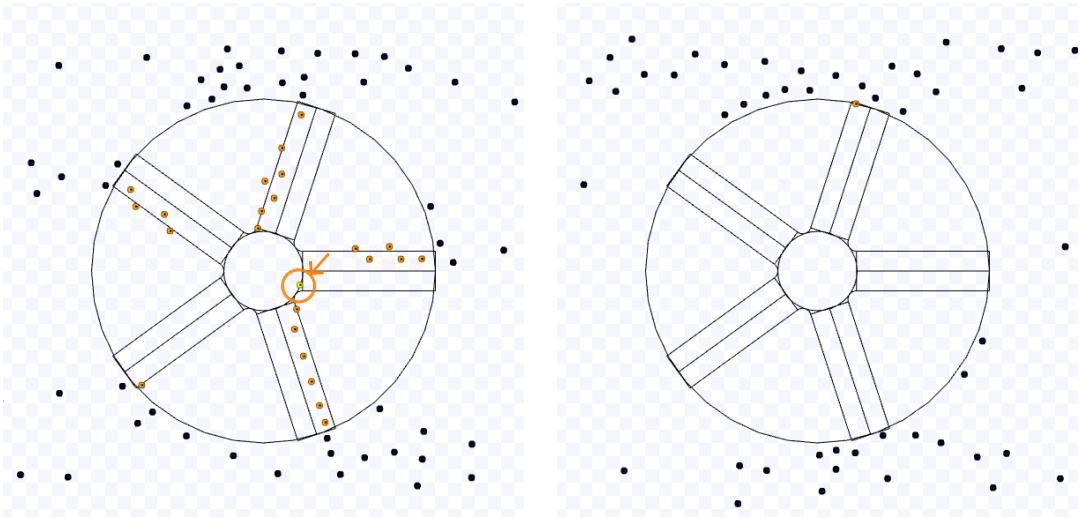
Part (a) is approximated by the expected minimum distance of a robot from its starting position going directly to an entrance path of a lane without moving next to the border of the working area circular sector. This approximation considers that it is more probable that the first robot to arrive at the entrance path was initially in front of the entrance path. Otherwise, the robot would have to move next to the



(a) (b)

(c) (d)

Figure 6.11: Example of an experiment of the TRVF algorithm with default values and $N = 100$. (a) Start of the experiment. (b) Robots go through the entrance path. (c) The first robot to reach the target area. (d) The first robot to leave the working area. Full simulation available on `https://youtu.be/bNIgvgpl_J4`, accessed on 14 October 2022.



(e) (f)

Figure 6.12: Continuation of Figure 6.11. (e) The last robot to reach the target area. (f) End of the experiment.

working area border until it arrives at the entrance path, which is more distant, as the cyan robots in Figures 6.11b–6.11d. From Figure 6.13a, the sector angle in front of the entrance path is given by $\alpha_4 = \arctan(d/(2D))$. Thus, the expected time is calculated from the expected minimum distance (Casella and Berger, 2002, p. 229) for the number of robots located in the sector angle α_4 when the experiment started $(\frac{\alpha_4}{\alpha}\frac{N}{K})$, i.e.,

$$\frac{E}{\left(\frac{\alpha_4}{\alpha}\frac{N}{K}+1\right)\bar{v}} = \frac{E}{\left(\frac{\alpha_4}{2\pi}N+1\right)\bar{v}} = \frac{E}{\left(\frac{\arctan(\frac{d}{2D})}{2\pi}N+1\right)\bar{v}}.$$

Figure 6.13b helps to understand how part (b) is calculated. This part is obtained from the time of the first robot moving at average speed through a distance of d_s meters via the entrance path, going through the circular path until the point A_6, leaving by this curved path and running again a distance of d_s meters via the exit path. The curved path length is obtained from the circular sector angle $\beta = \pi - \alpha$ (because the sum of the angles inside the quadrilateral containing the point A_6 must be 2π and the internal angle on the right-hand side is equal to α due to parallelism of the line in the lane).

The length d_s in Figure 6.13b is calculated from $\overline{C_1C}$ and $e_s = |\overline{A_1C_1}|$ in Figure 6.13c. The triangle ABC is the same depicted in the proof of Lemma 4.12, where the distance to the target centre for the robot to start turning was calculated using this

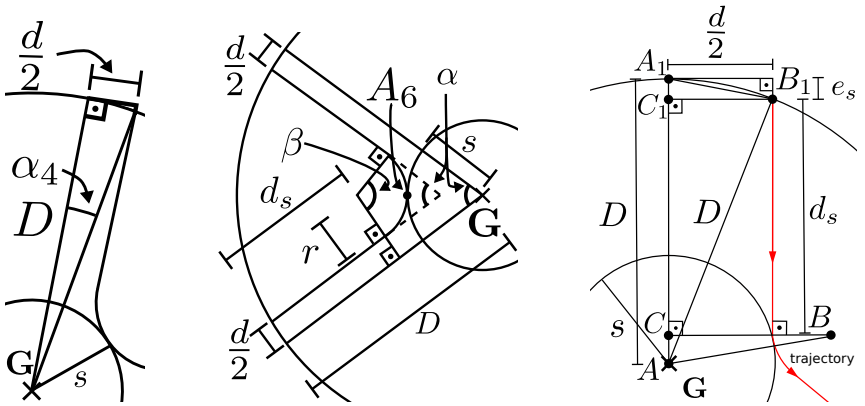


Figure 6.13: (a) Sector angle α_4 where the first robot on that lane arrives. (b) Estimated location A_6 where the first robot will arrive at the target area. (c) Right triangles $A_1B_1C_1$ and ABC for calculating the lenghts d_s and e_s.

triangle, so, from the proof of that lemma,

$$|\overline{AC}| = \sqrt{(r+s)^2 - (r+d/2)^2} = \sqrt{s(2r+s) - d\left(r + \frac{d}{4}\right)}.$$

From the right triangle AB_1C_1, $|\overline{AC_1}| = \sqrt{D^2 - d^2/4}$. As $D = |\overline{A_1A}| = |\overline{AC_1}| + |\overline{A_1C_1}| = \sqrt{D^2 - d^2/4} + e_s$, $e_s = D - \sqrt{D^2 - d^2/4}$. Also, $D = |\overline{A_1A}| = |\overline{A_1C_1}| + |\overline{C_1C}| + |\overline{AC}| = e_s + d_s + \sqrt{s(2r+s) - d\left(r + \frac{d}{4}\right)}$. Therefore,

$$d_s = D - e_s - \sqrt{s(2r+s) - d\left(r + \frac{d}{4}\right)} = \sqrt{D^2 - d^2/4} - \sqrt{s(2r+s) - d\left(r + \frac{d}{4}\right)}.$$

As mentioned before, part (b) is the time to go through the entrance lane, the curved path and the exit path, measuring d_s, $r\beta$ and d_s, respectively. Consequently, part (b) is given by $\frac{2d_s + r(\pi - \alpha)}{\bar{v}}$.

From experiments, part (c) is proportional to the number of robots in each lane, that is, $C_{TRVF}\frac{N}{K}$, for a constant C_{TRVF} that abstracts the characteristics of the experiment. The final result holds by adding parts (a)-(c). □

6.1.4 Mixed Teams

The MT is used by $M < N$ robots that do not know the algorithm executed by the other robots. As explained below, the estimation for MT shows that the estimations

of the presented algorithms can be used as a basis for further derivations. A robot at position $\mathbf{r}_i(t)$ in an MT executes a fixed alternative algorithm Alg which returns a vector $\mathbf{F}_{Alg}$. Following the definitions of Section 5.3, denote *ad hoc robots*, the group of M robots not knowing the algorithm used by the other group, and *aware robots*, the group of $N - M$ robots. Thus, their attractive force in an MT is simply $\mathbf{F}_{\mathbf{A}}(\mathbf{r}_i(t)) = \mathbf{F}_{Alg}(\mathbf{r}_i(t))$. As noticed in the previous algorithms, deriving an exact equation of the task completion time function from the controller equations is complicated because two groups of robots are executing possibly two different algorithms.

As shown in Section 5.4.1.3, using NC has the best results among the experimented alternative algorithms. Thus, $\mathbf{F}_{Alg}$ is the same as in (6.2). This alternative algorithm was used in the following illustrations.

Figures 6.14–6.17 show the execution of MT by 10% and 50% of ad hoc robots (in grey) to the total number of robots, respectively, when the aware robots are executing the SQF algorithm with its default values. In these figures, the total number of robots is $N = 100$. In Figure 6.14b, the usual behaviour of the 10% of the ad hoc robots in grey is to go directly to the target. In Figure 6.14c, some grey robots left the target region and are going to the next target on both sides (marked with circles and arrows). When the two last robots running NC leave the target area in Figure 6.14d (with circles and arrows), they try to go to the next target on the left-hand side, but other robots using the SQF are blocking their way. Due to that blockage, the two robots will continue through the SQF-induced leaving route until they can go to the left (Figure 6.15e). After that, only robots using SQF are in the experiment (Figure 6.15f). When few robots use NC, this often happens.

Figures 6.16 and 6.17 have more robots using NC. From the bottom of Figure 6.16b, more of them go directly to the target than in the previous example. Robots trying to reach the target may be pushed by robots using SQF on the way to leave the working area (Figure 6.16c). The cluttering formed by them may continue until the last robot using SQF reaches the target area (marked with a circle and an arrow in Figure 6.16d). This tendency to clutter occurs proportionally to the number of robots using NC as the alternative algorithm. The robots using NC which are oriented to the target but were on the SQF-induced leaving route are more likely to be the last ones to arrive at the target region (as the two grey robots marked with circles and arrows near the target in Figure 6.17e). At the end of this experiment, they are the last robots to leave the working area (one of them is marked with a circle and an arrow in Figure 6.17f).

Figures 6.18–6.21 illustrate the execution of MT with 10% and 50% of ad hoc robots to the total number of robots for an experiment with the TRVF with the default parameter values as the algorithm of the aware robots. In these figures, $N = 100$. As occurred for SQF, using 10% of the robots has almost the same result as using only TRVF for all robots. Figure 6.18b shows the robots using NC arriving

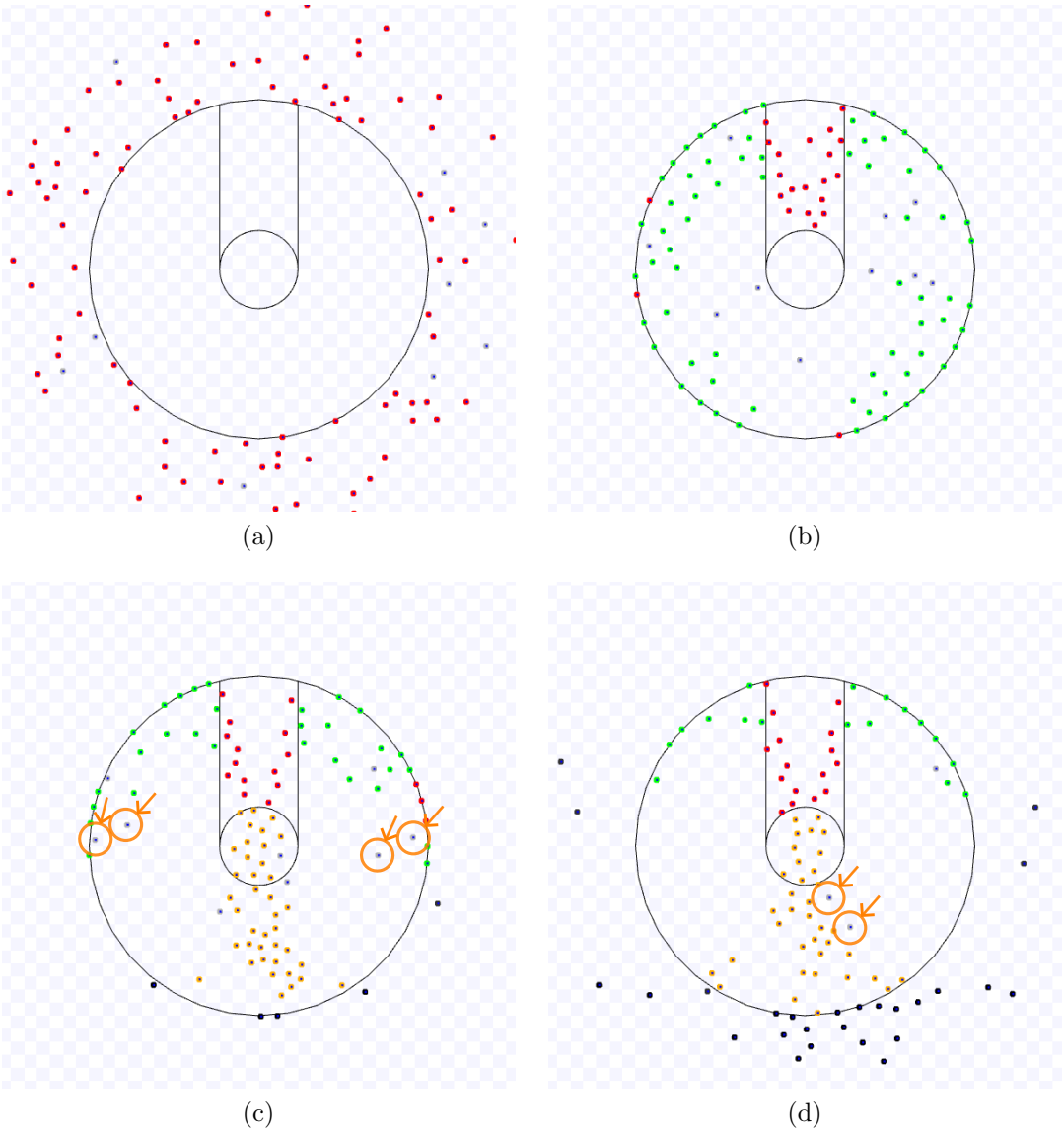
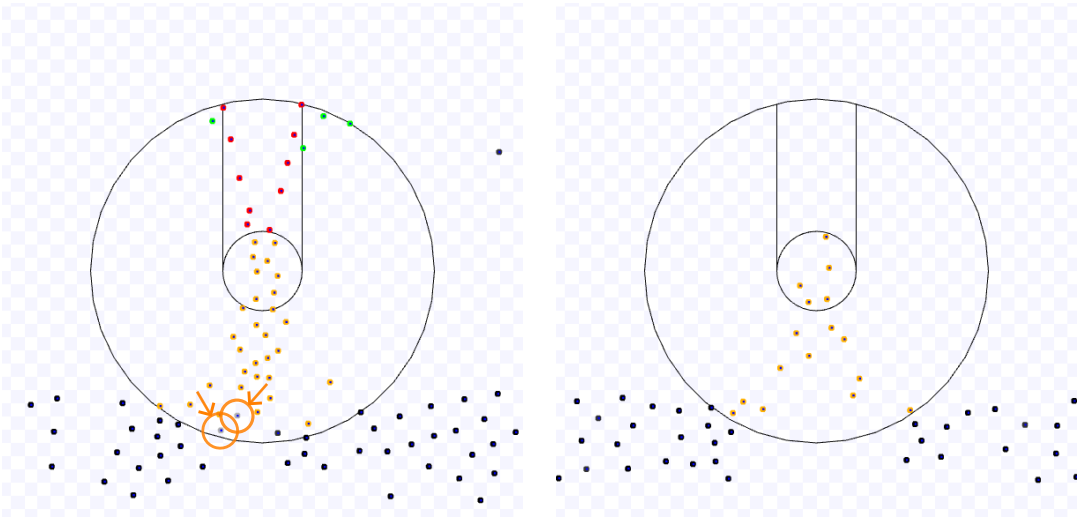


Figure 6.14: Example of an experiment of MT with 10% of ad hoc robots ($M = 10$ and $N = 100$) and the SQF algorithm executed by the others. (a) Start of the experiment. (b) Robots using NC go directly to the target. (c) Some grey robots go to the next target. (d) The last robots using NC leave the target, but SQF robots are on their way to the next target. Full simulation available on `https://youtu.be/uBetIxGOl20`, accessed on 24 April 2023.



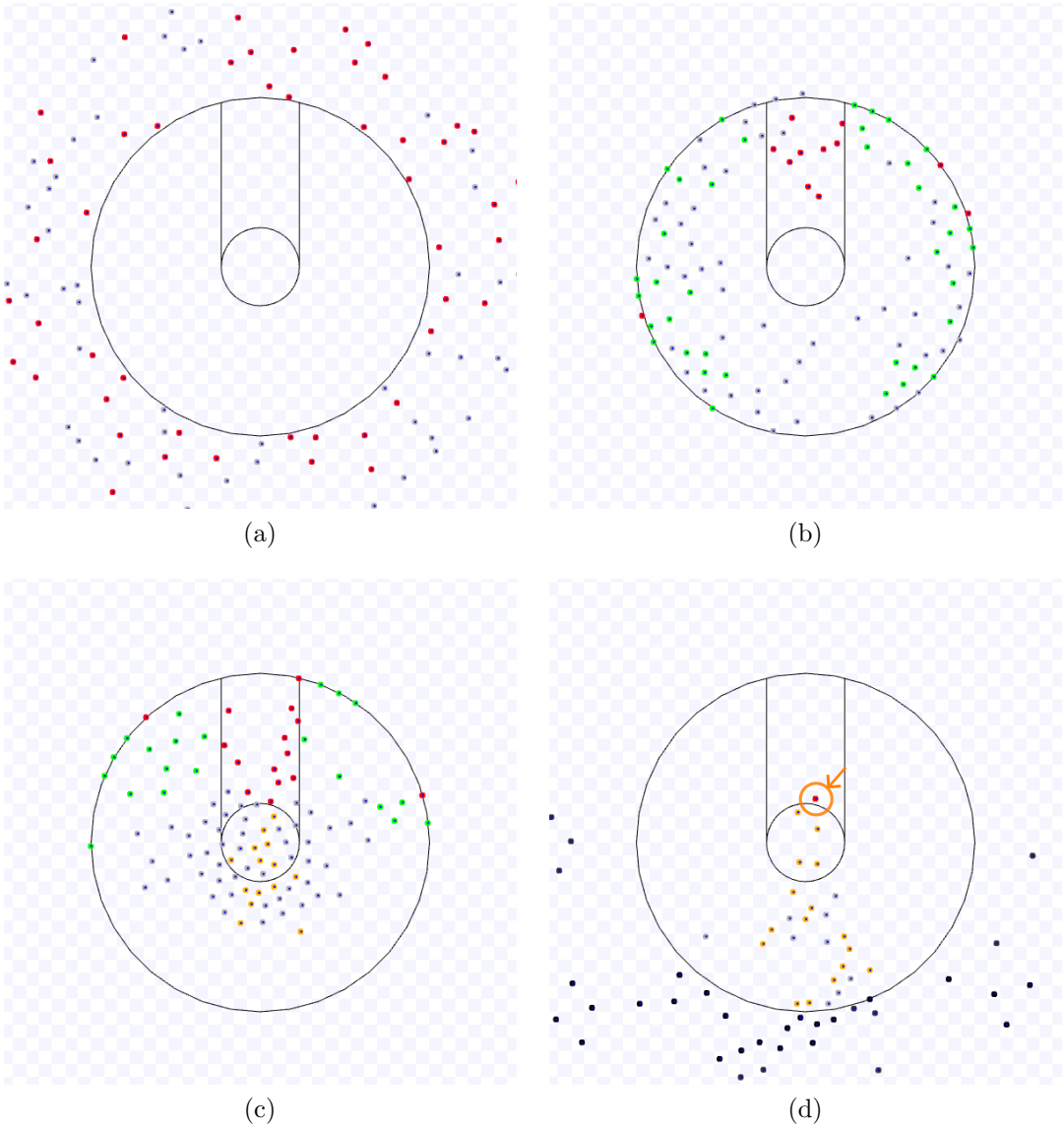
(e) (f)

Figure 6.15: Continuation of Figure 6.14. (e) The last grey robots leave the working area through the SQF-induced leaving route. (f) The remaining robots use the SQF algorithm near the end.

at the target area through the free space between the lanes. If a robot using NC is on the left-hand side, but its new target is on the right-hand side, it has to wait for robots using TRVF open space in the lane blocking its passage (as the grey robot marked with a circle and an arrow on the left side near the target area in Figure 6.18c). When a few robots use NC, they leave the working area faster than robots using TRVF, as they tend to follow the shortest distance to the target through the spaces between lanes (Figure 6.18d). In Figure 6.19e, all robots executing NC have finished the task. Only robots with TRVF are leaving the target area by exit paths until the end of the experiment in Figure 6.19f.

Figures 6.20 and 6.21 exemplify the usage of 50% of ad hoc robots, and the others execute TRVF with default values. As the proportion of robots using NC is more than in the previous example, they are more likely to cover the open space near the target (Figure 6.20b) and cause congestion inside the target area (Figure 6.20c). If a lane has more robots using NC, they hamper robots using TRVF to go through it, as the exit lane on the top left-hand side in Figure 6.20d. As for the 10% case, the last robot using NC leaves before the TRVF robots (marked with a circle and an arrow in Figure 6.21e). However, fewer TRVF robots than in 10% case are inside the working area until the end of the experiment (Figure 6.21f).

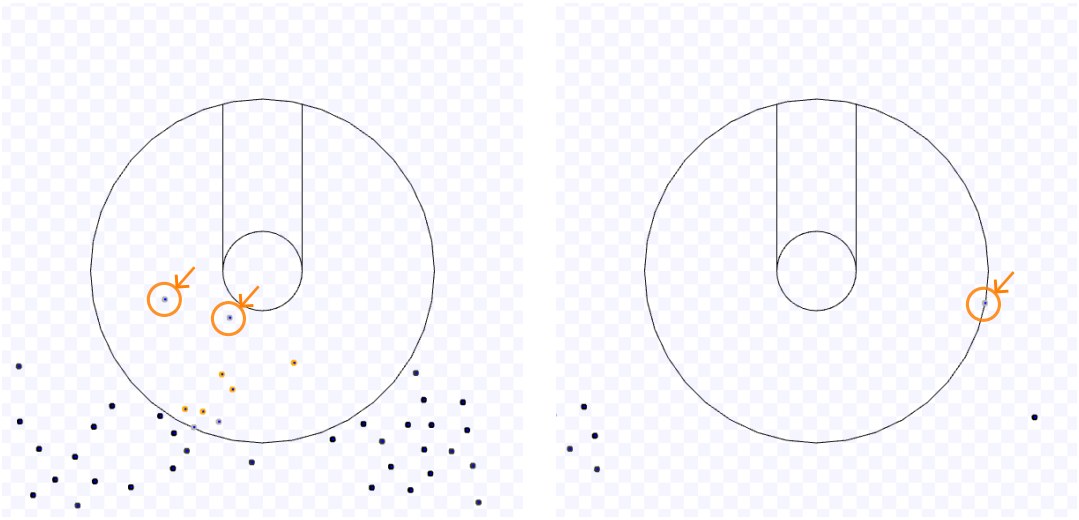
These illustrations hint that the proportion of ad hoc robots induces a cluttering similar to that occurred by using only NC. Thus, the task completion time tends to be



(a) (b)

(c) (d)

Figure 6.16: Example of an experiment of MT with 50% of ad hoc robots ($M = 50$ and $N = 100$) and the SQF algorithm executed by the others. (a) Start of the experiment. (b) Robots using NC go directly to the target. (c) Robots using NC go to the target after being pushed by SQF robots leaving the target. (d) The last robot using SQF reaches the target area. Full simulation available on `https://youtu.be/a-05slaNud8`, accessed on 24 April 2023.



(e) (f)

Figure 6.17: Continuation of Figure 6.16. (e) The robots on the SQF-induced leaving route are the last to arrive at the target region. (f) End of the experiment.

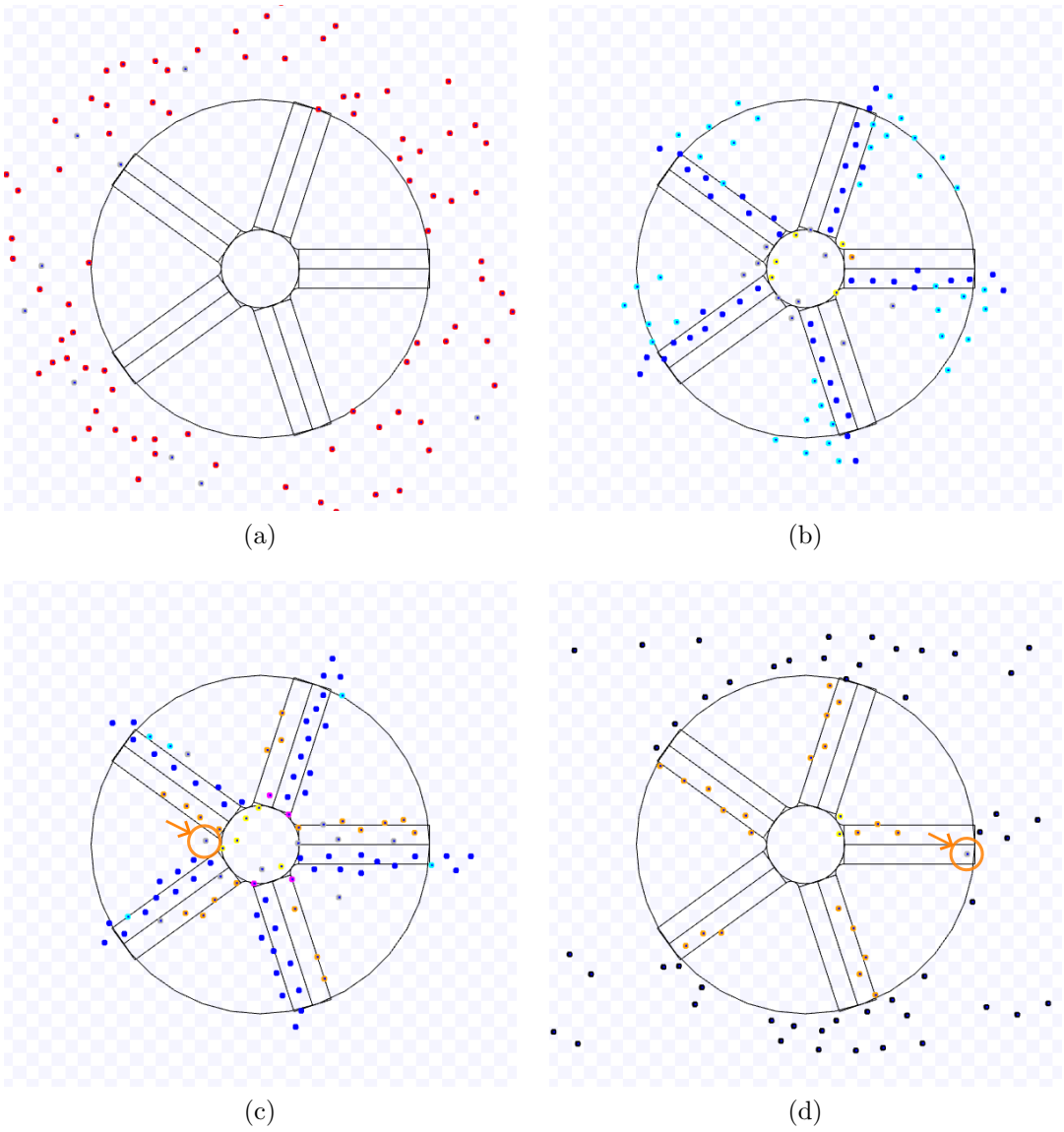
near NC time as the number of ad hoc robots M increases and the algorithm followed by the aware robots as M decreases. To better show this, Figures 6.22 and 6.23 show the result from the experiments with MT for scenarios with the robots executing SQF and TRVF with different values of M. The bars mean the 99% confidence interval of the average for 40 runs for each value in the horizontal axis. (The examples above are shown for convenience for the explanation below. They are obtained from the Section 5.4.1.3. Results containing the estimations for other values of percentage are shown in Section 6.2.) Based on these observations, the next estimation shows an approximation for an MT.

Estimation 4. *The estimated expected time for N robots starting at a random position with a distance from the target centre in $[D, D+E)$ to arrive at the common target area and leave the working area when $M < N$ robots are using MT with NC as the alternative algorithm depends on the ratio $p = M/N$ and is expressed by*

$$\mathbb{E}[t_{MT}](p) \approx C_{MTNC}(p)\mathbb{E}[t_{NC}] + C_{MTAw}(p)\mathbb{E}[t_{Aw}] \tag{6.16}$$

for constants C_{MTNC} and C_{MTAw} (dependent on p), Aw being the algorithm followed by the $N - M$ aware robots and its estimated expected time $\mathbb{E}[t_{Aw}]$.

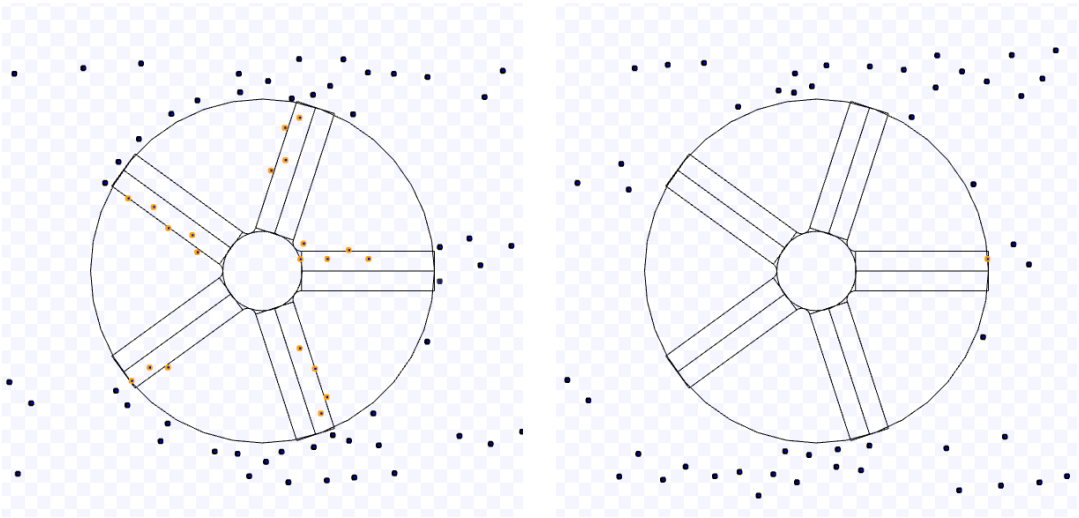
Explanation. Let C_{MTNC} be a given constant abstracting the environment and the dynamics of the ad hoc robots for the ratio p and C_{MTAw} a similar constant for



(a) (b)

(c) (d)

Figure 6.18: Example of an experiment of MT with 10% of ad hoc robots ($M = 10$ and $N = 100$) and the TRVF algorithm executed by the others. (a) Start of the experiment. (b) Robots using NC arrive at the target area through the free space. (c) A grey robot using NC waits for robots in a lane. (d) The last grey robot is leaving the working area. Full simulation available on `https://youtu.be/VlhXKM77q90`, accessed on 24 April 2023.



(e) (f)

Figure 6.19: Continuation of Figure 6.18. (e) Only robots using TRVF are inside the working area. (f) End of the experiment.

the aware robots. Thus, as observed in the experiments (Figures 6.22 and 6.23 and Section 6.2), the estimated expected time is the sum of the estimated expected time of NC and the control algorithm followed by the aware robots each multiplied by the constants C_{MTNC} and C_{MTAw}, respectively. □

6.2 Experiments and Results

Experiments were performed in the Stage simulator (Gerkey, Vaughan, and Howard, 2003) with the algorithms in the previous section using the same task scenario and the default values used in the examples of the last section. Only the number of robots N is varied from 20 to 300 with steps of 20 and the robotic movement type (holonomic and non-holonomic, to show the adaptability of our estimations). For each N, 40 executions were run. In the figures of this section, the bar intervals are the 99% confidence interval of the mean value of the variable considered. "Experiments" and "Estimation" stand for these intervals of the experimental data mean and the equation estimated from these experiments, respectively.

Moreover, the constants for the estimated time equations were obtained with the least squares method from the mean value of the experimental data. It is important to heed that the presented estimations produce curves that can be fitted to the simulation results by tuning the coefficients of the expressions. Thus, using wrong equations with

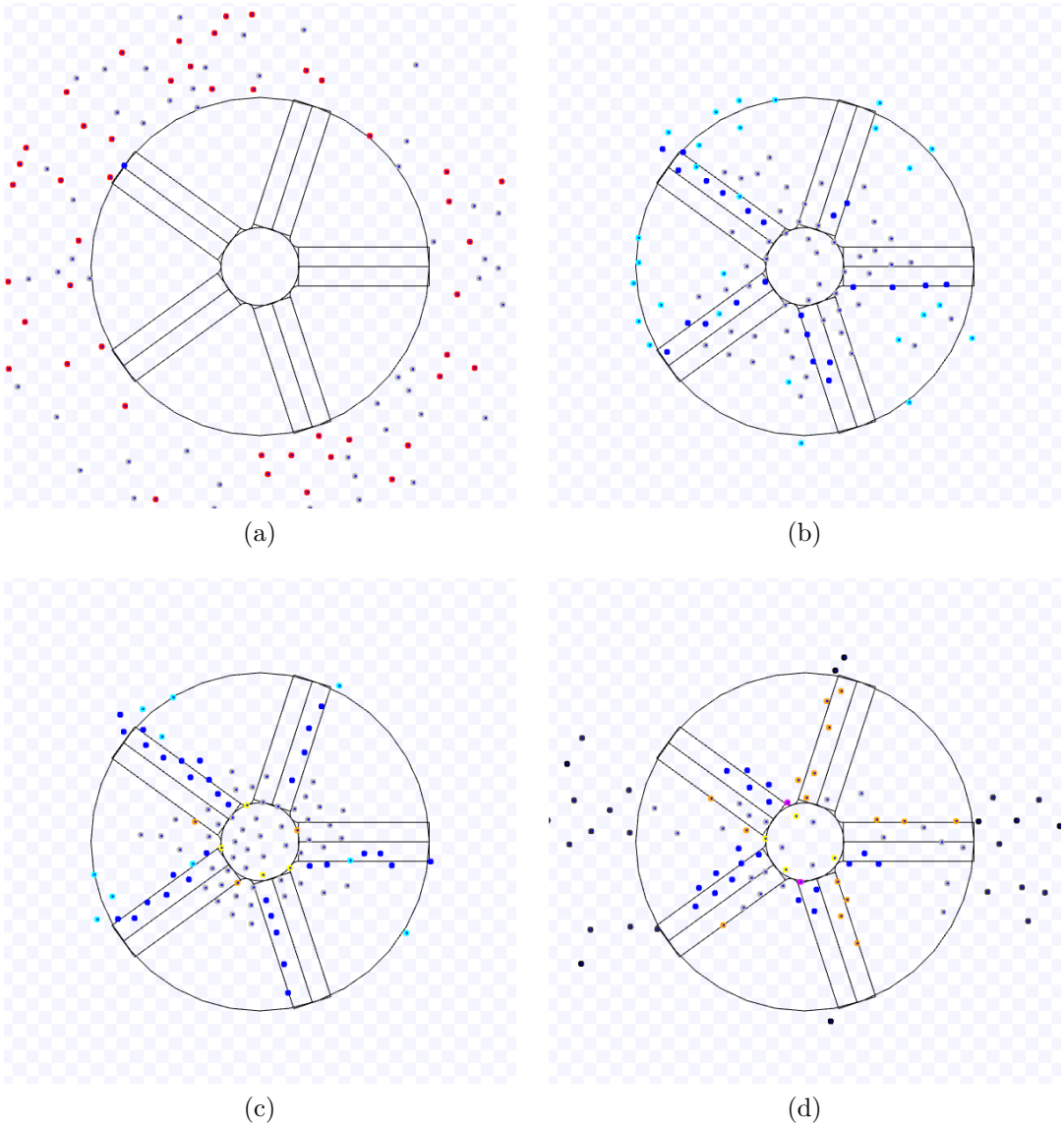
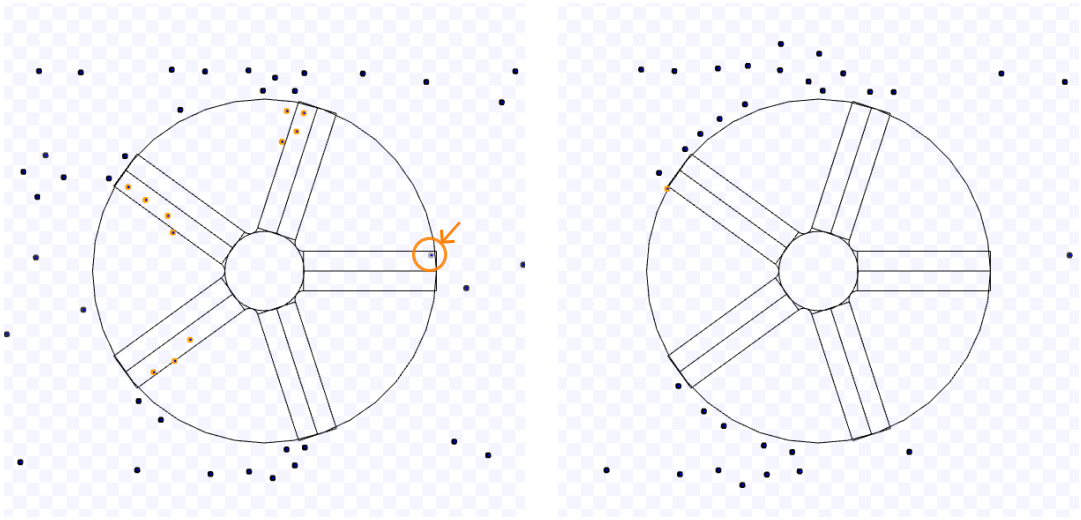


Figure 6.20: Example of an experiment of MT with 50% of ad hoc robots ($M = 50$ and $N = 100$) and the TRVF algorithm executed by the others. (a) Start of the experiment. (b) Robots using NC go directly to the target. (c) NC robots (in grey) cluster in the target area. (d) NC robots interfere with the access of robots using TRVF. Full simulation available on `https://youtu.be/1nxSEsre5mk`, accessed on 24 April 2023.



(e) (f)

Figure 6.21: Continuation of Figure 6.20. (e) The last robot using NC leaves the working area. (f) End of the experiment.

constants obtained from the least square method does not work appropriately. For example, the Estimation 1 has two cases depending on N. If one of the cases is ignored and the other is applied for every N, even by fitting its constant, the estimation will be further from the experiment data than the presented equation.

In addition, Appendix E contains more figures that support the insights in this chapter. (The source codes of the experimented algorithms and estimations are in Passos (2022a).)

Figure 6.24 and 6.25 present the comparison between the experimental data and the estimated expected time computed by Estimation 1 for NC and $s \in \{3, 5, 7\}$ m. Observe that the estimation is closer as the value of s is smaller.

Figures 6.26 and 6.27 show the comparison for the SQF algorithm with the estimated time in Estimation 2 for $s \in \{3, 5, 7\}$ m. The negative value of C_{SQF1} in the non-holonomic case for $s = 3$ m (Figure 6.26b) means that the waiting time was less than the predicted travel time of the last robot (i.e., the value from Estimation 2 without the term $C_{SQF1}N$). Notice that the shapes of the graphs are different for different type of robotic movement, although the equations are the same. This is because the average linear speed varies further with more robots in the non-holonomic case than with the holonomic one, as it will be shown later.

Figures 6.28 and 6.29 show the comparison for TRVF algorithm with the estimated time in Estimation 3 for $s \in \{3, 5, 7\}$ m. Observe that using this estimation for holonomic robots and $s = 7$ m is not as close as for non-holonomic ones (Figure 6.29).

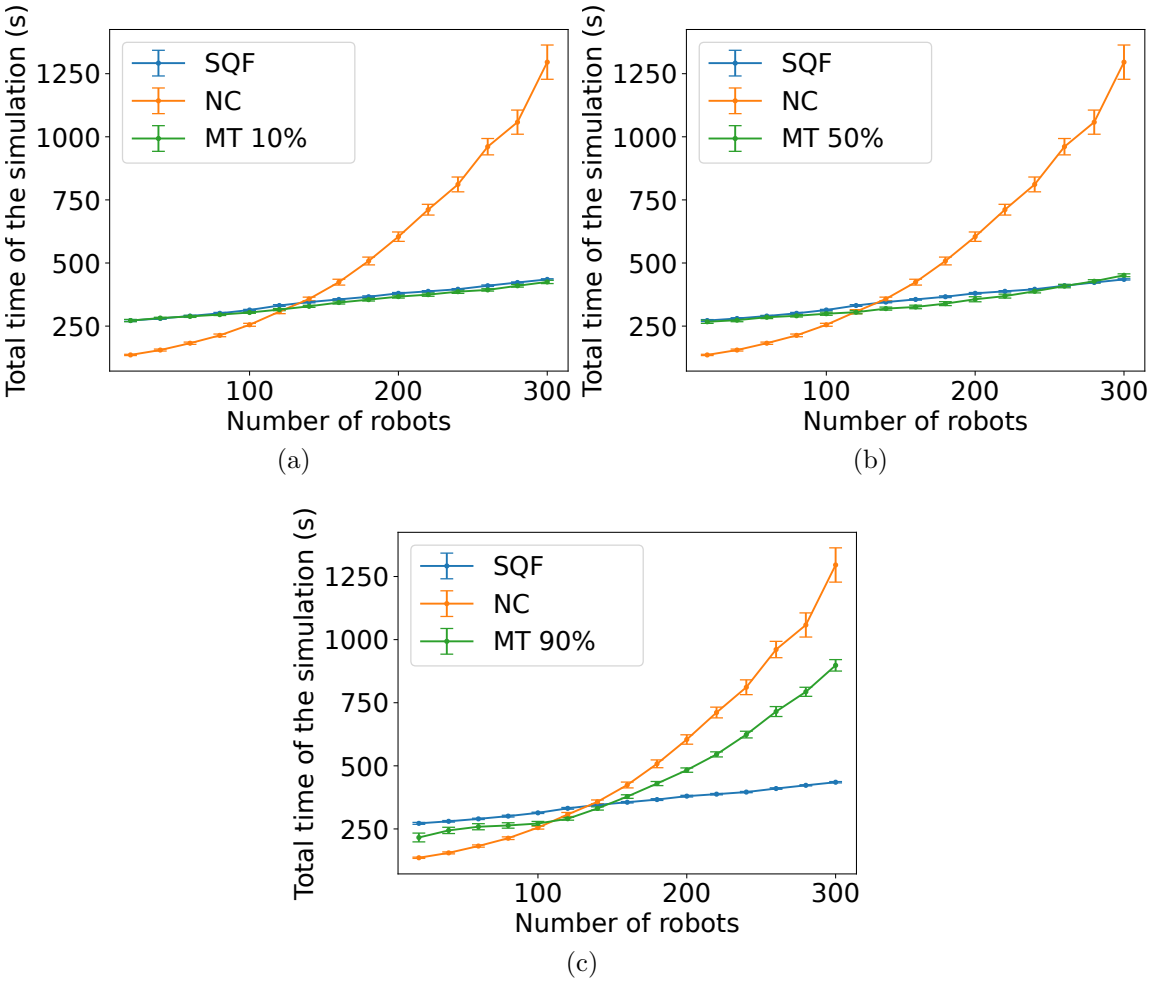


Figure 6.22: Task completion time versus the number of robots for NC, SQF and an MT with NC as the alternative algorithm and aware robots with SQF for holonomic robots and (a) $M = 0.1N$, (b) $M = 0.5N$ and (c) $M = 0.9N$.

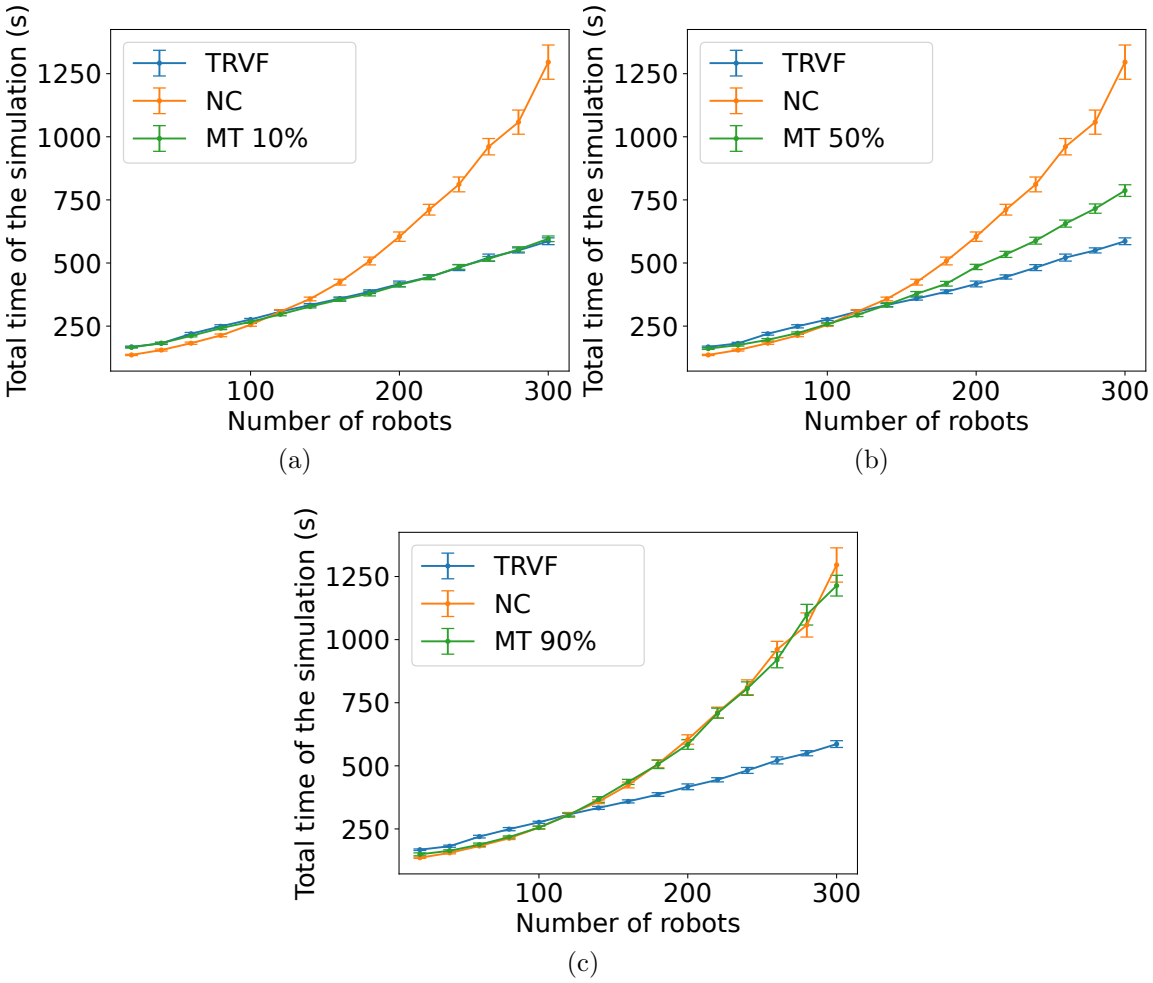


Figure 6.23: Task completion time versus the number of robots for NC, TRVF and MT with NC as the alternative algorithm and aware robots with TRVF for holonomic robots and (a) $M = 0.1N$, (b) $M = 0.5N$ and (c) $M = 0.9N$.

In the algorithms where the lower number of robots differs more from the estimation, the robots had little mutual influence as they were scattered. These estimations are not close to the experimental data for these values because they were deduced from the global behaviour affected by the influence of their surroundings assuming that the algorithms were applied with a large number of individuals.

Figure 6.30 gives the results of the root mean squared error normalised by the y-axis range (NRMSE) for the estimations in Figures 6.24–6.29. From it, I considered that these estimations are close to the experimental data because the NRMSE values were lower than 0.03. In Figure 6.30, the lowest error for holonomic robots was for SQF with $s = 5$ m while for non-holonomic case, it was NC with $s = 3$ m. NC has the highest error for $s = 7$ in both types of robotic movement because the graph seems linear instead of a curve.

In addition, this work assumes a fixed working area. That being so, notice that the influence of the number of robots has a limit because a higher number will clutter the environment no matter the algorithm. Thus, the parameters related to the environment area affect these functions in this sense.

Although it is not explicit from Estimations 1, 2 and 3 for the estimated expected time, $\bar{d}$ and $\bar{v}$ depend on N. Figure 6.31 presents the relation between the average speed and the number of robots for the algorithms for $s = 3$ m. Note that the range for the holonomic case is smaller (around 0.25 m/s) than for the non-holonomic robots (from 0.05 to 0.25 m/s). The holonomic robot can maintain the linear speed in all directions, while the non-holonomic one needs to go forward or backwards, depending on the situation, yielding positive and negative values for the linear speed. When more robots are next to each other, this forward and backward movement intensifies in the non-holonomic case. In the holonomic case, the slight rise in the speed as the number of robots increases is due to the velocity of a holonomic robot being calculated for every direction, as they can move freely in any direction. Thus, as more robots are added, they are deviating more and consequently, more speed is included in the average.

Figure 6.32 shows the relation between the mean distance between the robots and the number of robots. For both types of robots, the trends in the mean distance are similar: as the number of robots grows, they have to move nigher to each other in the algorithms.

Figure 6.33 presents an instance of the estimations for MT with the proportion $p = 10\%$ of ad hoc robots varying from 10 to 90%, and SQF is the control algorithm of the aware robots. For reference, the task completion time from the experiments of SQF and NC is shown. (Section E.1 contains the figures for p from 20% to 90% in increments of 10%.) Figure 6.34 shows the results of the NRMSE for the estimations for these values of p. From the NRMSE values, they match better for non-holonomic robots from 70 to 90%.

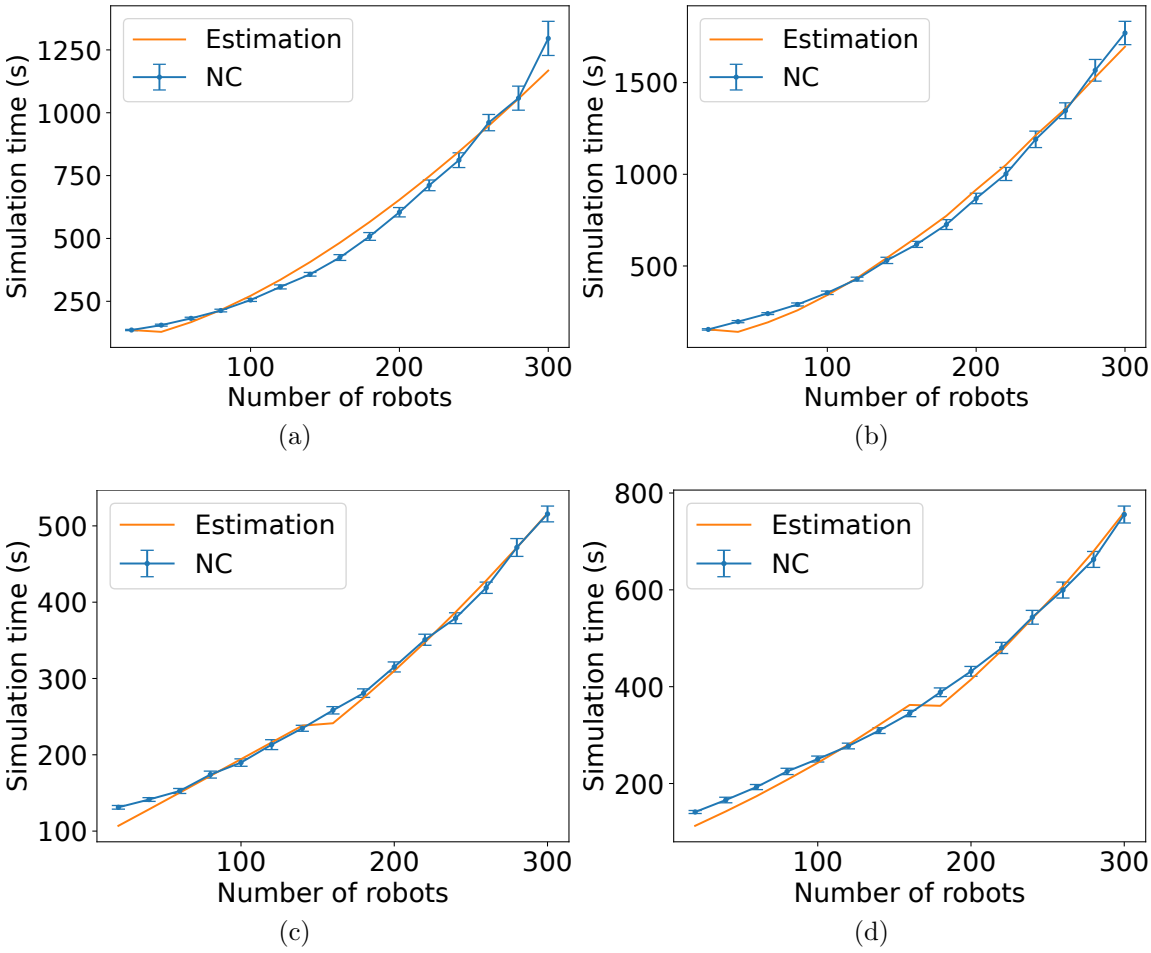


Figure 6.24: Comparison of the estimated expected time and the experimental data for NC, $s = 3$ m with (a) holonomic robots ($(C_{NC1}, C_{NC2}) = (2.1431, 0.6502)$) and (b) non-holonomic robots ($(C_{NC1}, C_{NC2}) = (2.7784, 0.5694)$) and $s = 5$ m with (a) holonomic robots ($(C_{NC1}, C_{NC2}) = (1.0994, 0.2614)$) and (b) non-holonomic robots ($(C_{NC1}, C_{NC2}) = (1.2867, 0.2539)$).

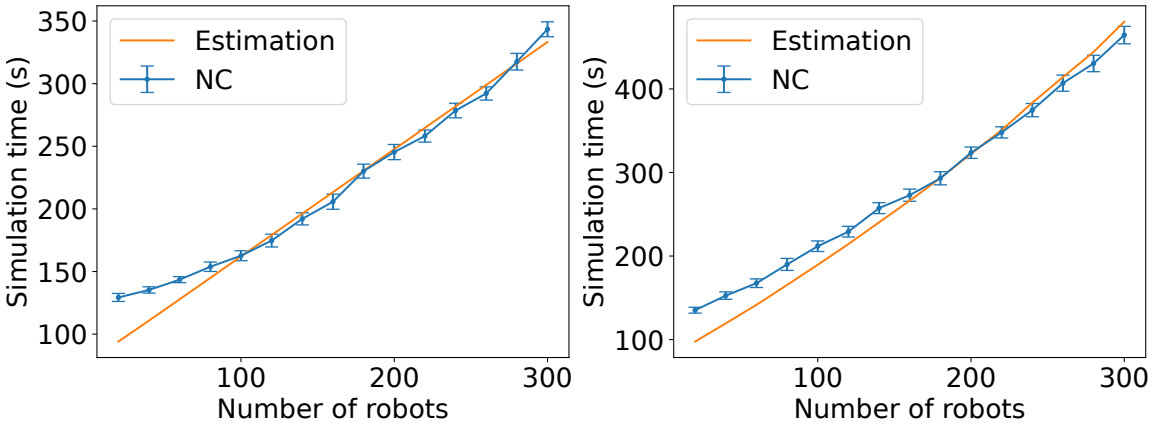


Figure 6.25: Comparison of the estimated expected time and the experimental data for NC, $s = 7$ m with (a) holonomic robots ($(C_{NC1}, C_{NC2}) = (0.857, 0)$) and (b) non-holonomic robots ($(C_{NC1}, C_{NC2}) = (0.9795, 0)$).

Similarly, Figure 6.35 shows an example of the estimations for MT with TRVF as the control algorithm of the aware robots with $p = 10\%$. As before, they have the task completion time from the experiments of TRVF and NC for reference. (The figures for p ranging from 20% to 90% in steps of 10% appear in Section E.1). Figure 6.36 shows the results of the NRMSE for these estimations. Due to the estimation equation for TRVF working better for a higher number of robots, the ad hoc estimation reflects this result. Also, the NRMSE is higher for holonomic case.

Figure 6.37 displays the change in constants C_{MTAw} and C_{MTNC} by the experimented percentage of ad hoc robots shown above for each tested control algorithm of the aware robots and robots' movement type. Note that when C_{MTNC} rises, C_{MTAw} decreases, because it is expected to these constants are proportional to the number of robots executing the algorithm associated with each of them. Although the values of the constants are different, the trend for the SQF algorithm is similar for both robots' movement types (Figures 6.37a and 6.37b). The same happens for the TRVF algorithm in Figures 6.37c and 6.37d. In addition, notice that the sum of $C_{MTAw}(p)$ and $C_{MTNC}(p)$ is not exactly one for a fixed p because these constants abstract properties of the environment and dynamics of the robots, which may have variations due to the unpredictable behaviour resulting from the combination of swarms with different algorithms in the same region.

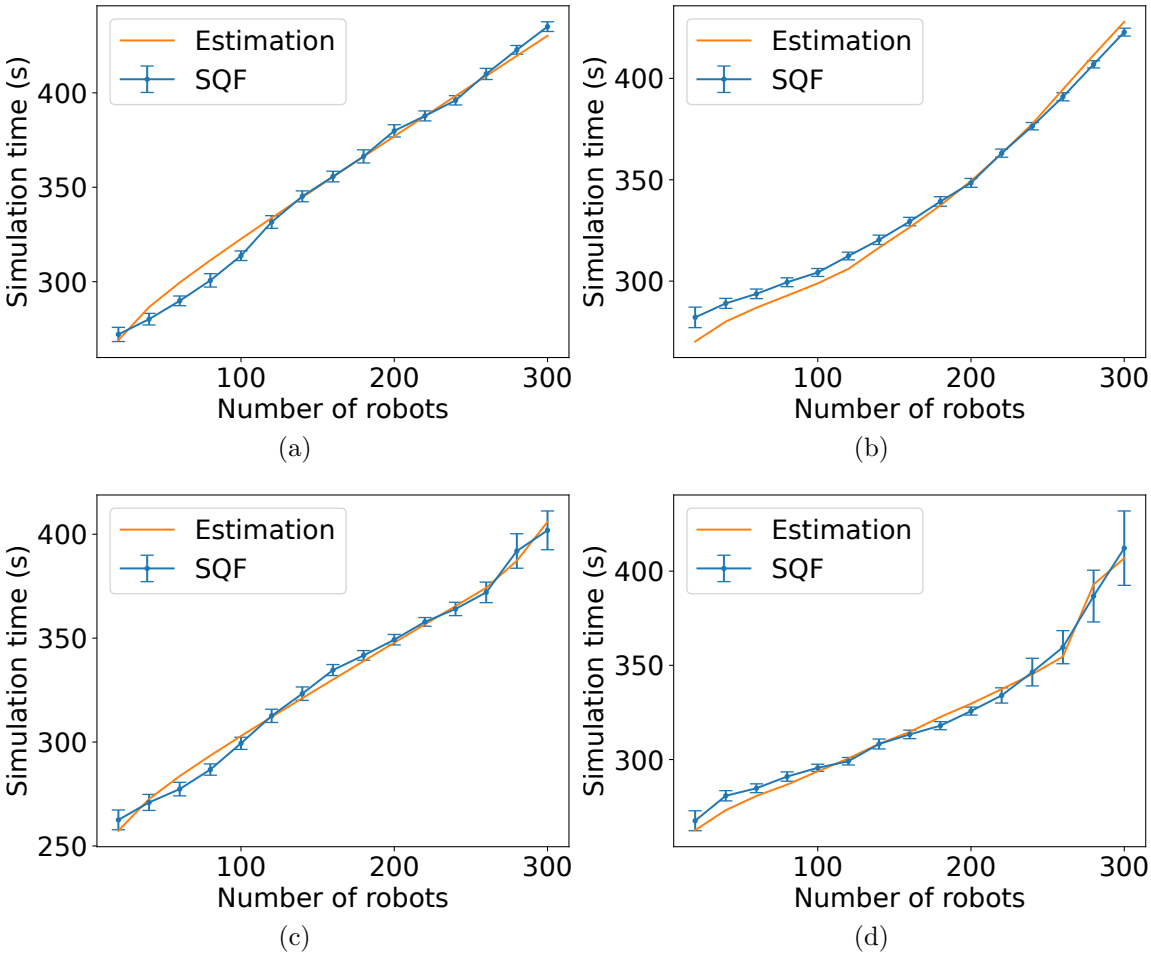


Figure 6.26: Comparison of the estimated expected time and the experimental data for the SQF algorithm, $s = 3$ m with (a) holonomic robots ($(C_{SQF1}, C_{SQF2}) = (0.5277, 0)$) and (b) non-holonomic robots ($(C_{SQF1}, C_{SQF2}) = (-0.1202, 0)$) and $s = 5$ m with (c) holonomic robots ($(C_{SQF1}, C_{SQF2}) = (0.4346, 0.0016)$) and (d) non-holonomic robots ($(C_{SQF1}, C_{SQF2}) = (0.0037, 0.0004)$).

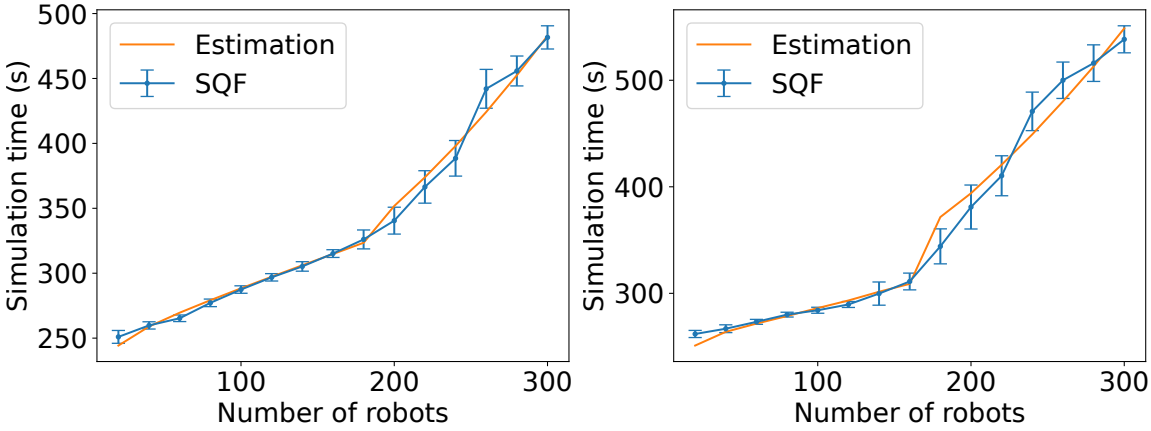


Figure 6.27: Comparison of the estimated expected time and the experimental data for the SQF algorithm, $s = 7$ m with (a) holonomic robots ($(C_{SQF1}, C_{SQF2}) = (0.4237, 0.0026)$) and (b) non-holonomic robots ($(C_{SQF1}, C_{SQF2}) = (0.0928, 0.0022)$).

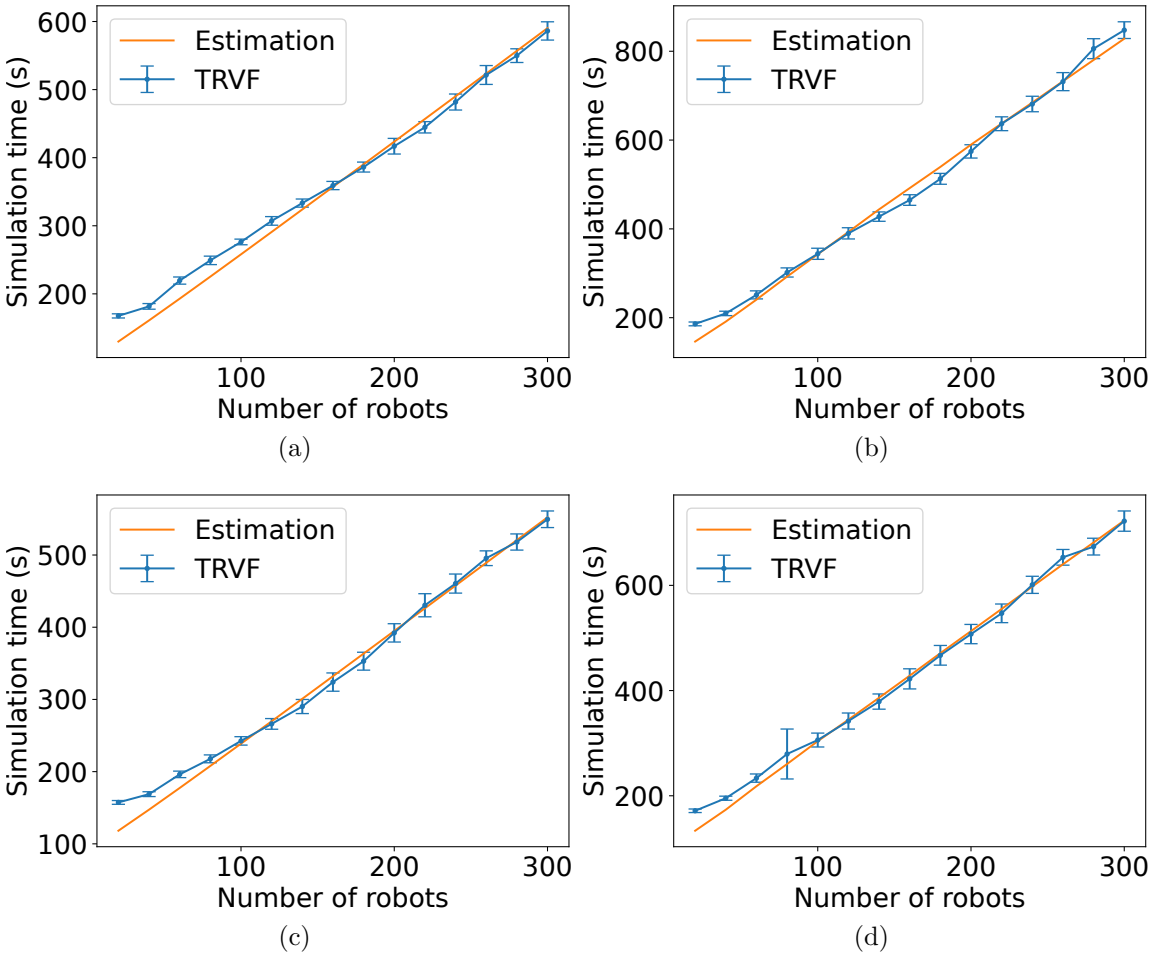
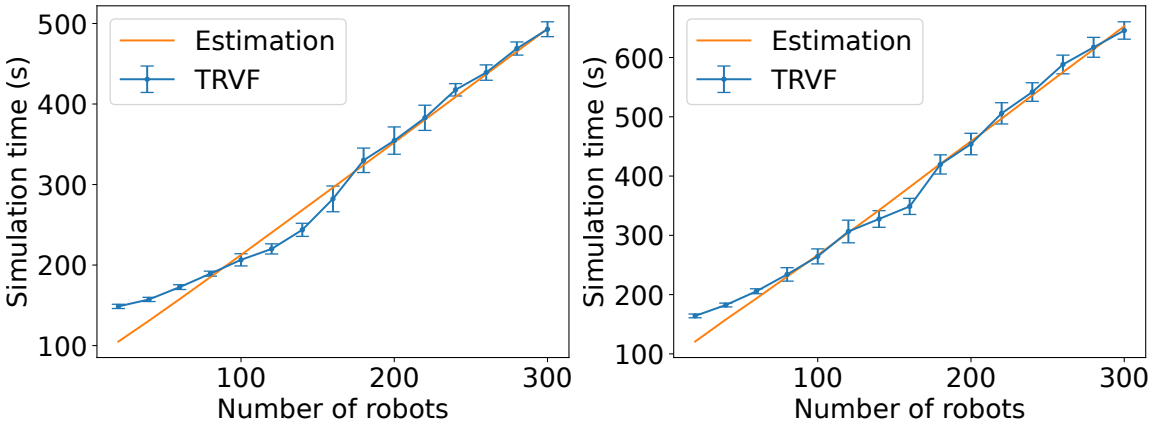
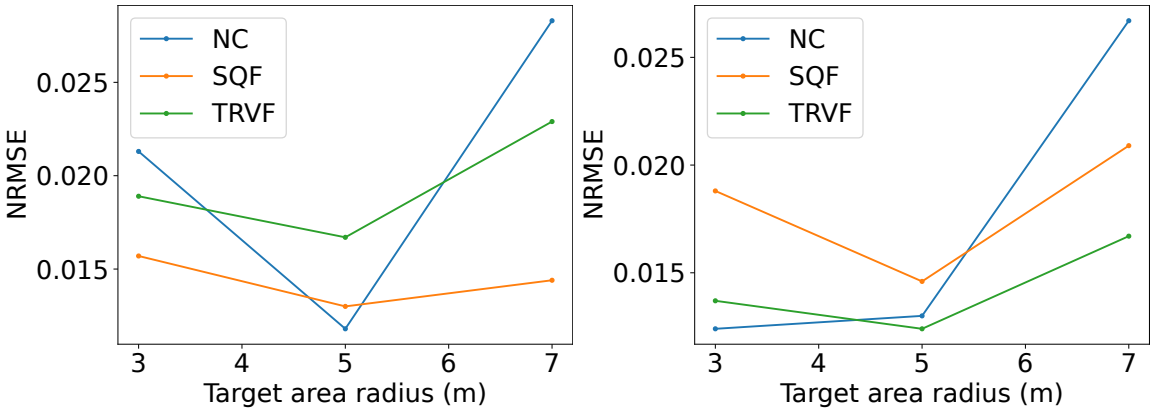


Figure 6.28: Comparison of the estimated expected time and the experimental data for the TRVF algorithm, $s = 3$ m with (a) holonomic robots ($C_{TRVF} = 8.3866$) and (b) non-holonomic robots ($C_{TRVF} = 10.0287$) and $s = 5$ m with (c) holonomic robots ($C_{TRVF} = 7.9166$) and (d) non-holonomic robots ($C_{TRVF} = 9.3022$).



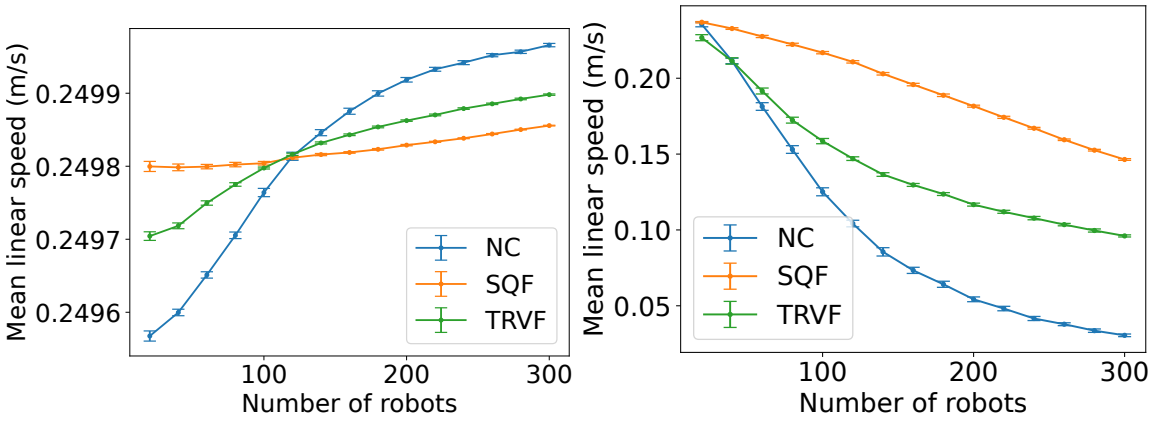
(a) (b)

Figure 6.29: Comparison of the estimated expected time and the experimental data for the TRVF algorithm, $s = 7$ m with (a) holonomic robots ($C_{TRVF} = 7.1029$) and (b) non-holonomic robots ($C_{TRVF} = 8.7103$).



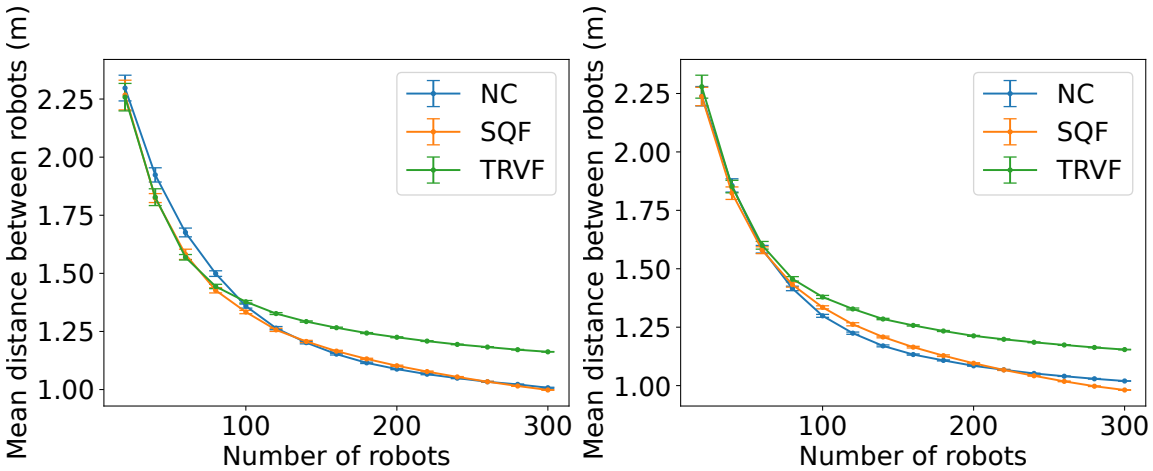
(a) (b)

Figure 6.30: Normalised root mean square error of the estimations in Figures 6.24–6.29 for (a) holonomic and (b) non-holonomic robots.



(a) (b)

Figure 6.31: Comparison of the linear speeds of the algorithms for $s = 3$ m with (a) holonomic robots and (b) non-holonomic robots.



(a) (b)

Figure 6.32: Comparison of the mean distance between the robots of the algorithms for $s = 3$ m with (a) holonomic robots and (b) non-holonomic robots.

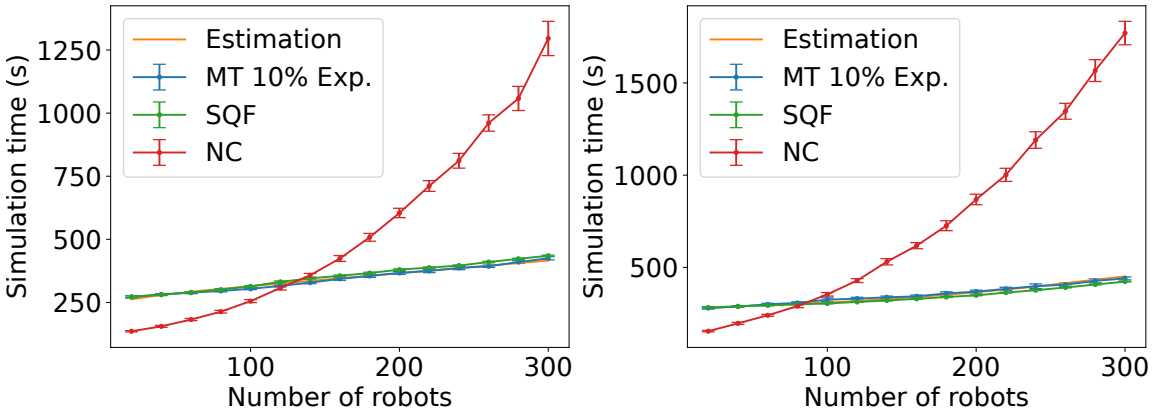


Figure 6.33: Comparison of the estimated expected time and the experimental data for MT with NC as the alternative algorithm, the swarm control algorithm is the SQF algorithm and $p = 10\%$ for (a) holonomic robots ($C_{MTNC} = -0.0042$ and $C_{MTAw} = 0.9779$) and (b) non-holonomic robots ($C_{MTNC} = 0.0069$ and $C_{MTAw} = 1.027$).

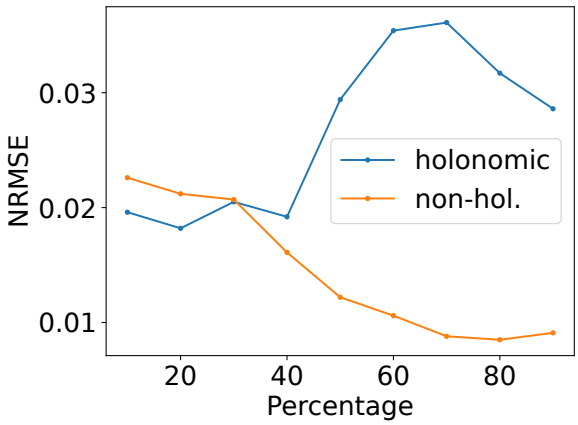


Figure 6.34: Normalised root mean square error of the estimations in Figures 6.33–E.4 for MT with SQF and NC.

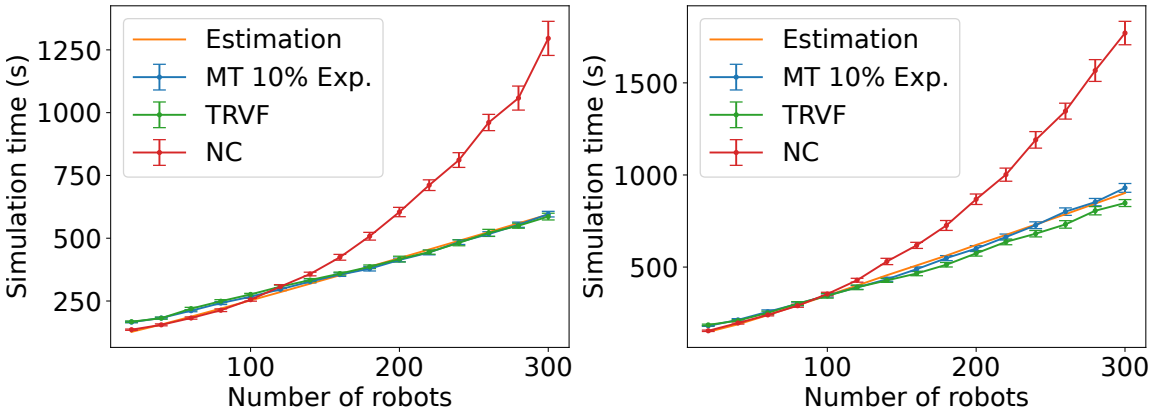


Figure 6.35: Comparison of the estimated expected time and the experimental data for MT with NC as the alternative algorithm, the swarm control algorithm is the TRVF algorithm and $p = 10\%$ for (a) holonomic robots ($C_{MTNC} = 0.0309$ and $C_{MTAw} = 0.9459$) and (b) non-holonomic robots ($C_{MTNC} = 0.0766$ and $C_{MTAw} = 0.9323$).

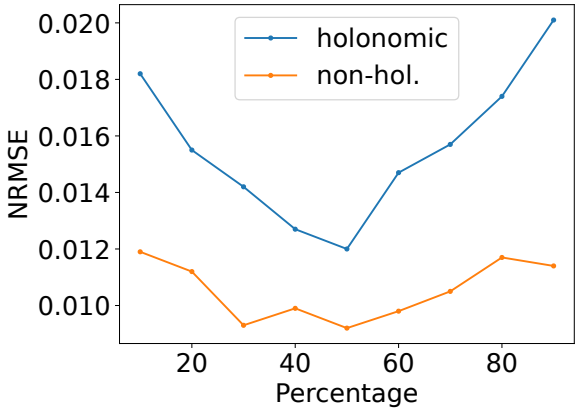


Figure 6.36: Normalised root mean square error of the estimations in Figures 6.35–E.8 for MT with TRVF and NC.

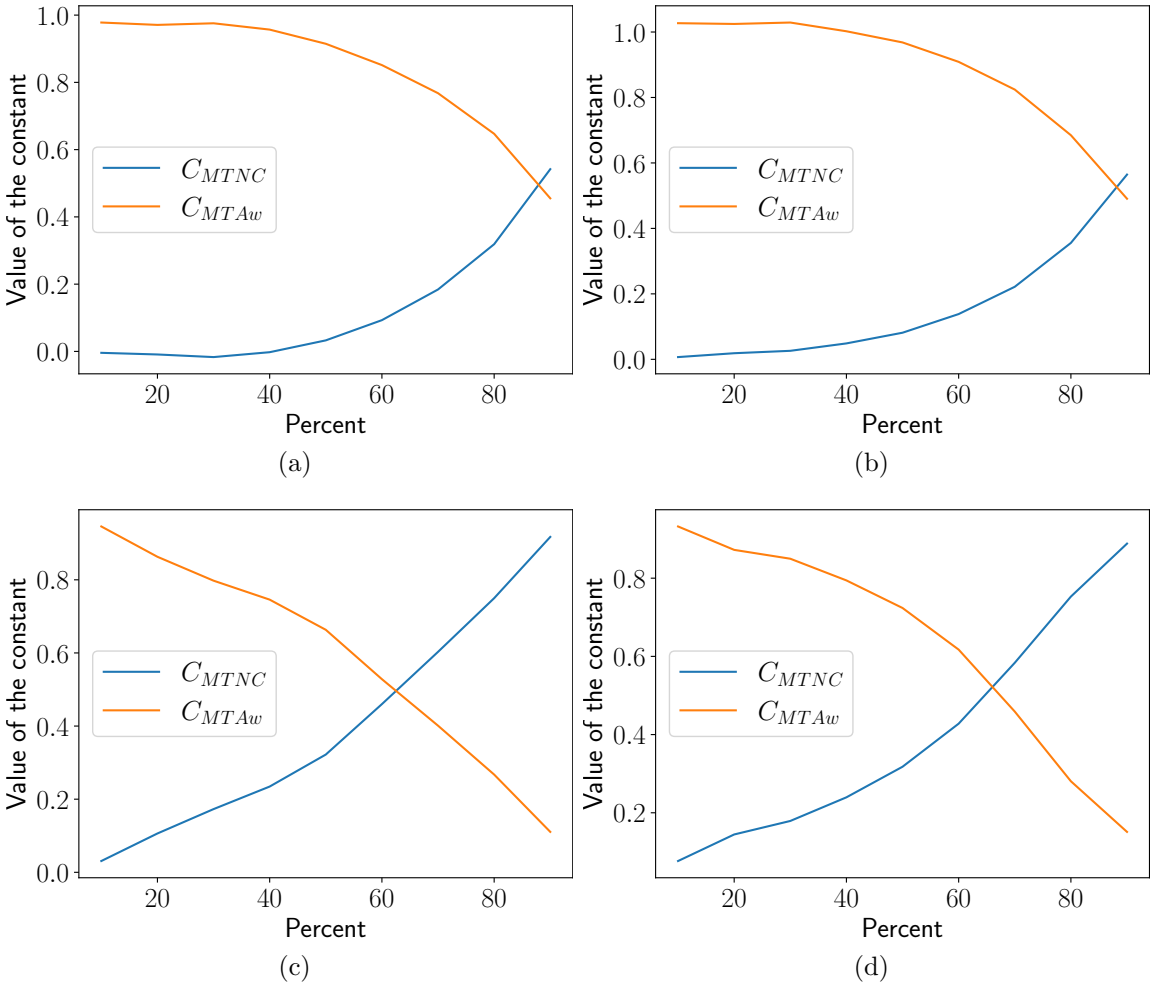


Figure 6.37: Constants used for MT estimations (C_{MTAw} and C_{MTNC}) versus percent of ad hoc robots for SQF as the control algorithm of aware robots and (a) holonomic and (b) non-holonomic robots; and TRVF and (c) holonomic and (d) non-holonomic robots.

Chapter 7

Conclusion and future work

In a swarm of robots, when all the robots have to go to the same region, congestion interferes with their time to move to a shared target region and from it. The common target problem in robotic swarms occurs in that situation. Thus, congestion control algorithms must be executed to minimise that interference. Hence, this thesis presents an extensive theoretical study for a better understanding of that problem and new algorithms. Additionally, for a situation where two groups of robots do not know each other and have to go to a common target area, this thesis experiments and discusses new solutions that do not need learning algorithms to overcome that situation.

A novel metric was proposed for measuring the effectiveness of algorithms to minimise congestion in a swarm of robots trying to reach the same goal: the common target area throughput. In addition, the asymptotic throughput for the common target area was defined as the throughput when the time tends to infinity.

Assuming robots moving at constant maximum speed and the distance between each other being as close as possible to a fixed value, it was shown how to calculate the maximum throughput for different theoretical strategies to arrive at the common circular target region: (i) making parallel queues to reach the target region, (ii) using a corridor with robots in hexagonal packing to enter in the region, and (iii) following curved trajectories to touch the region boundary. As the growing number of robots needs to be considered in robotic swarms, the asymptotic throughput abstracts the measure of how many robots can reach the target as time passes and, consequently, as the number of robots raises. When a closed form for the asymptotic throughput is given, it can be used to compare algorithms, as it will always be finite.

To achieve the analytical results of the proposed theoretical strategies, constant linear speed and distance between the robots are assumed. However, in the real world, these values are dynamic. Thus, inspired by these strategies, two novel algorithms were developed: SQF and TRVF. It was demonstrated by simulations in the Stage platform that SQF outperforms TRVF and the state-of-the-art algorithms PCC,

PCC-EE and EE by time for reaching the target for a large number of robots from 280 for holonomic and 120 for non-holonomic robots. From these same numbers of robots, swarms running the SQF algorithm also use less time for total simulation time (i.e., reaching plus leaving the target) than the other algorithms. SQF also outperforms these algorithms concerning the average target leaving time from 180 non-holonomic robots. However, for holonomic robots, TRVF is better up to 240 individuals and PCC-EE from 280 robots. Additionally, this work showed that the previous approaches might completely fail for small target areas, that is, with an area fitting less than five times the area of a robot, and they are outperformed in throughput by SQF when they succeed. Moreover, TRVF helps us to understand the potential impacts of the variation in the linear speed and distance between the robots when translating an idea from theoretical strategies to concrete algorithms.

Even though SQF has more throughput than TRVF for a number of robots greater than 140 for holonomic robots and 60 for non-holonomic ones, the comparison of their corresponding inspiration strategies (hexagonal packing and touch and run, respectively) had reversed outcomes concerning asymptotic throughput. When constant maximum linear speed and fixed minimum distance between robots are assumed, analytical calculations of the throughput for a given time and the asymptotic throughput could be provided for the different theoretical strategies. Based solely on these calculations, it could be compared which strategy is better. However, for robots using artificial potential fields, it is not straightforward to get explicit throughput equations due to the changeability of those quantities previously assumed constant. Then, in the lack of closed asymptotic equations, simulations were performed to get experimental throughput and compare algorithms for varying linear speed and inter-robot distance. As shown by the experimental data, their variation and the effect of the other robots in the trajectory affect the throughput, but the analytically calculated throughput values are still within the upper bounds of the ones obtained from the simulation.

Accordingly, when the closed theoretical equation cannot be obtained, the comparison by experimental throughput per number of robots has the same result as the simulation time per number of robots. However, if the asymptotic throughput of any algorithm can be theoretically calculated, it may be utilised to compare them to decide which one is better. When a robotic swarm has all robots going to the same target, the function relating the number of robots and the time of arrival on the target region tends to be infinite as the number of robots grows, while the function relating the number of robots and throughput tends to a finite number. When an algorithm exhibits a lower target region arrival time for a number of robots, the throughput is higher, so the comparison by this latter metric can replace the former as it reflects the same ordering but is reversed. Although for the proposed algorithms closed asymptotic equations tackling dynamic speed and inter-robot distance were not

given, I believe that, if they were, the comparison could be done only by the analytical asymptotic throughput, as accomplished when the linear speed and distance between the robots are constant. Thus, for common target area congestion in robotic swarms, the throughput is well suited for comparing algorithms due to its abstraction of the rate of the common target area access as the number of robots grows, whether the closed throughput equation is given or not.

Furthermore, algorithms were presented for ad hoc robots for the common target problem in a robotic swarm. Unlike the usual approach of an ad hoc multi-agent system in which the agent needs to learn about the environment or other agents, no learning is required. Experiments showed that not following a robot and using NC as the alternative algorithm for AHF have better results in most tests. When only one ad hoc robot executes it, the total simulation time will increase by a small amount. However, as more robots run AHF, the swarm behaves similarly to the alternative algorithm. That is a consequence of its estimated expected task completion time being a linear combination of the estimated expected time of NC, used as the alternative algorithm in the experiments, and the control algorithm proportional to the ratio of robots. If the percentage of ad hoc robots using MT is lower than 30% and 60% for non-holonomic and holonomic robots, respectively, the results on the average total simulation time are better for NC as the alternative algorithm, although it is not suitable for a swarm with only one team.

In addition, this thesis presented estimations of the expected task completion time in relation to the number of robots in algorithms for the common target problem using potential fields. These relations were calculated by analysing the experiments with NC, SQF, TRVF and AHF. Besides the number of robots, the task completion time estimations are also expressed in terms of the average speed of the robots and algorithm parameters such as the working area radius, the target area radius, the number of lanes in TRVF, its central angle, the distance travelled through its lanes and the ratio of the number of ad hoc robots to the total number of individuals. In addition, the experimental relation of the average speed and distance between the robots as a function of the number of individuals in the swarm for SQF and TRVF was illustrated.

7.1 Future Work

The key contribution of the presented theoretical work is a fundamental theoretical study of congestion in Swarm Robotics, which already served as an inspiration to create new algorithms. However, future work could extend this study further by considering varying linear speeds and distances between the robots. Research about which statistical distributions are suited for these variables applied to the throughput functions developed here may be a starting point for analysing that study. In this

way, random variables and their properties also could be appropriate for this analysis.

Furthermore, this thesis worked with the throughput of the target area in equivalence with the target area reaching time as an inspiration to new algorithms. As the leaving time may interfere with some algorithms, such as NC and MT, a study of a measure involving the throughput of the robots leaving a target area and the working area may be interesting to broaden the understanding of the common target problem.

Moreover, dynamical systems and chaos theory may be applied for further studies on the hexagonal packing strategy, both for constant and variable hexagonal packing angles. In the constant case, these topics may help to find the exact limit for a given hexagonal packing angle instead of an interval, as did here. Also, these studies may answer whether finding such an exact limit is indeed feasible. When dealing with varying speed and inter-robot distance, the case of a variable hexagonal packing angle will appear. Therefore, in that case, these studies may discover the most probable limit or a new interval of the lower and upper bound of the limit. Although effort has to be invested in that research, these analyses are interesting and may deliver new facts useful for a better understanding of the common target problem and the development of new algorithms or adaptations of the existing ones.

Other subsequent work is to test newer obstacle avoidance methods with the proposed algorithms. As presented in the related work, despite the ORCA algorithm not working in the common target problem, new methods such as the Model Predictive Control (MPC) – also known as Receding Horizon Control (RHC) – may yield distinct outcomes for that problem. Also, they may have different behaviours when put in place of the potential fields in the presented algorithms, which may or may not be more efficient in some aspects.

In addition, another future work is to experiment with the execution of mixed approaches. An example of a mixed approach is an algorithm that employs multiple corridors or lanes around the target area, as in the TRVF algorithm but uses potential fields similar to those in the SQF algorithm.

As the usage of aerial robots is increasing nowadays, another future step is to adapt to three-dimensional robots and experiment with their type of movements. As this thesis considered robots and the target regions in a plane, the equations must be extended to three dimensions. Also, three-dimensional robots have different types of movement, and they have to be experimented with the algorithms for feasibility. For instance, aircraft-like robots cannot stop in mid-air, and some aerial robots are controlled by Dubins paths.

Furthermore, using learning to discover better algorithms may be valuable. Statistical learning, deep learning and evolutionary algorithms are examples of methods to be applied to discover better algorithms. Another subsequent work is the comparison of reinforcement learning for ad hoc robots with MT. As said before,

the advantage of MT is that the robot does not spend time learning. However, it is worthwhile to analyse how the effort of learning about other agents by methods such as reinforcement learning can be advantageous to the common target problem with ad hoc robots.

Additionally, the work about estimations is a fundamental first step to study the common target problem as a local-to-global analysis of the task completion time from the specification of the individual robot to the swarm behaviour. An extensive and challenging study is needed to explain the local-to-global nature of these subsumed details. This invokes a theory to untangle this relationship, and I aim to start a discussion about that with this work by presenting a baseline approximation to compare with future work. There is room for improvement on these estimations. Approaches using adaptations of the methods discussed in the Section 3.2 are welcome to be compared. Thus, in future work, methods from statistical mechanics, in special for many-body problems, will be applied to this problem and compared with those developed here.

Consequently, a question may arise: is it possible to find a potential field that minimises the relation of the task completion time versus the number of robots? Notice that if this is possible, a time relation for a given field has to be given to be minimised. From future improvements, I expect to be possible to find a better potential field minimising the task completion time per number of robots, presuming that finding it is decidable.

Finally, this work has direct application in diverse areas as subsequent works. A list of suggestions is below:

- emergency services – fast access to locations for rescue operations by a robotic swarm;
- surveillance or remote monitoring – mobile robots going from their stations to the target region to be monitored or surveilled;
- logistics – the transportation of goods by robots from various places to a common target area such as a parked ship or a warehouse;
- warehouse management – robots in a swarm inside a warehouse may have to stack products in a shared storage hack, and then the proposed algorithms are utilised to access the region of a common hack;
- construction – robots collecting materials in a shared deposit or disposing of debris in a common rubbish site;
- searching in regions harmful to humans – fast collection of precious or dangerous objects in areas such as deep sea, contaminated areas or small places;

- microscopically locating cancer cells – quick access to a tissue to be destroyed or given medicine by autonomous microscopic robotic swarms;
- disaster response – for a swarm entering a destroyed place in search of people or dangerous objects, such as non-activated bombs, and a swarm of autonomous firefight robots going to a region to put out fire quickly and leaving it to pick more extinguishing agents;
- plantation management – for robots hunting insects in a region, for accessing the infestation focus or the disposal site to throw the insects after captured; and
- common or hazardous waste collection – fast entry in the waste location in remote areas such as the ocean and for going to the disposal site to store the waste.

Appendix A

Pseudocodes of the State-of-the-art Algorithms

Algorithm 8 describes PCC assuming a robot executes it at every update in its position. It outputs a force vector pointing to where the robot must head depending on its current coordinates and state. This force vector has a constant magnitude K_{PCC}. The robot still has to compute the repulsive force by another procedure – not described in Algorithm 8. At line 6, **w** holds the robot position when it switches its state (lines 11 and 17). At line 22, the variable *iteration* – incremented at line 46 – must be initialised at zero before calling the PCC algorithm. Messages are sent in lines 40 and 43. The *warning* type at line 40 is sent by a *normal* robot to other robots in the danger region to warn them that they may switch to *waiting* state. The *stop* type at line 43 is for a *waiting* or *locked* robot to tell other robots outside the danger region that they may need to alter their state to *locked*.

Algorithm 10 describes the EE algorithm, executed after position updates. As before, it returns a force with constant magnitude K_{EE}. However, as this algorithm reduces the repulsive force, a function for computing it must be given and include the multiplicative constant and the robot and obstacle positions as parameters. In line 12, the repulsive force is reduced by half only for the robots that push the robot away from the entry region. However, in line 15, the repulsive force from nearby robots is reduced to half.

Algorithm 11 describes the PCC-EE algorithm. Similar to the previous algorithms, it returns a force with constant magnitude $K_{PCC\text{-}EE}$. In line 7, the PCC algorithm is called with the appropriate parameters. As said before, the PCC does not compute any repulsive force, so it returns only the attractive force according to the robot state and position. However, the EE algorithm alters the repulsive force depending on the position and whether the robot reached the target area from line 8 to 17.

Input : K_{PCC}: force magnitude;
$\mathbf{G}_1, \ldots, \mathbf{G}_n$: a list of $n \geq 2$ circular target region centres;
$s_1, \ldots, s_n$: a list of $n \geq 2$ target region radii;
j: the current target index;
r_σ, r_γ: radius of the free region and outer radius of the danger region, respectively;
δ, δ_{comm}: radius of the α-area and of communication, respectively;
T_{comm}, T_η: number of iterations before sending a message and for testing if a *waiting* robot changes state, respectively;
α_w, α_l: angle of the α-area to switch to *waiting* and *locked* states, respectively;
ρ_I: Probability of impatience.
Output: $\mathbf{F}$: force vector;

Get robot position $\mathbf{p}$, and let $\mathbf{G} = \mathbf{G}_j$ and $s = s_j$;
if *state = normal or state = impatient* **then**
  $\mathbf{F} \leftarrow K_{PCC} \frac{\mathbf{G}-\mathbf{p}}{\|\mathbf{G}-\mathbf{p}\|}$;
end
if *state = waiting or state = locked* **then**
  $\mathbf{F} \leftarrow K_{PCC} \frac{\mathbf{w}-\mathbf{p}}{\|\mathbf{w}-\mathbf{p}\|}$;
end
if *state = normal* **then**
  if *the robot is inside danger region and has neighbour in α_w-area* **then**
    state $\leftarrow$ waiting;
    $\mathbf{w} \leftarrow \mathbf{p}$;
  end
  if *the robot is outside danger region and has neighbour i in α_l-area* **then**
    $state_i \leftarrow$ robot i state from its message;
    if *$state_i$ = waiting or $state_i$ = locked* **then**
      state $\leftarrow$ locked;
      $\mathbf{w} \leftarrow \mathbf{p}$;
    end
  end
end

Algorithm 8: PCC algorithm (continues on Algorithm 9).

```
21 if state = waiting then
22 |  if iteration mod T_η = 0 and rand() < ρ_I then
23 |  |  state ← impatient;
24 |  end
25 end
26 if state = locked then
27 |  if the robot is inside danger region then
28 |  |  state ← waiting;
29 |  else if there is no neighbour i in α_l-area where state_i = waiting or locked
   |   then
30 |  |  state ← normal;
31 end
32 if state = impatient then
33 |  if ‖p − G‖ ≤ s then
34 |  |  state ← normal;
35 |  |  j ← j + 1;
36 |  end
37 end
38 if the robot detects another robot at distance δ_comm and iteration
     mod T_comm = 0 then
39 |  if state = normal and r_γ > ‖p − G‖ > r_σ then
40 |  |  Send a warning message with G and state;
41 |  end
42 |  if state = waiting or state = locked then
43 |  |  Send a stop message with G and state;
44 |  end
45 end
46 iteration ← iteration + 1;
47 return F;
```

Algorithm 9: PCC algorithm (continuation).

Input : K_{EE}: force magnitude;
$\mathbf{G}_1, \ldots, \mathbf{G}_n$: a list of $n \geq 2$ circular target region centres;
$s_1, \ldots, s_n$: a list of $n \geq 2$ target region radii;
j: the current target index;
r_γ: radius of the entry region;
ω: angle of the entry region;
$\mathbf{F_R}(K_r, \mathbf{p}, \mathbf{q})$: a repulsive force with multiplicative constant K_r, robot position $\mathbf{p}$ and obstacle position $\mathbf{q}$.

Output: a force vector;

Get robot position $\mathbf{p}$, and let $\mathbf{G} = \mathbf{G}_j$ and $s = s_j$;
Let N_r be the set of robots locally sensed and $\mathbf{q}_i$ the neighbour robot i position;
if $\|\mathbf{p} - \mathbf{G}\| < D$ **then**
  if *the robot is not in entry region by (2.1) and* $\|\mathbf{p} - \mathbf{G}\| > r_\gamma$ **then**
    Calculate $\mathbf{w}$ by (2.2);
    $\mathbf{F_A} \leftarrow K_{EE} \frac{\mathbf{w}-\mathbf{p}}{\|\mathbf{w}-\mathbf{p}\|}$;
    $\mathbf{F'_R} \leftarrow \sum_{i \in N_r} \mathbf{F_R}(K_r, \mathbf{p}, \mathbf{q}_i)$;
  else
    $\mathbf{F_A} \leftarrow K_{EE} \frac{\mathbf{G}-\mathbf{p}}{\|\mathbf{G}-\mathbf{p}\|}$;
    if *the robot is in entry region by (2.1) and* $\|\mathbf{p} - \mathbf{G}\| > r_\gamma$ *and did not reach the target area* **then**
      Let $N_r^1 \subset N_r$ be the set of robots that pushes the robot away from the entry region and $N_r^2 = N_r - N_r^1$;
      $\mathbf{F'_R} \leftarrow \sum_{i \in N_r^1} \mathbf{F_R}(\frac{K_r}{2}, \mathbf{p}, \mathbf{q}_i) + \sum_{i \in N_r^2} \mathbf{F_R}(K_r, \mathbf{p}, \mathbf{q}_i)$;
    else
      if *the robot reached the target area* **then**
        $\mathbf{F'_R} \leftarrow \sum_{i \in N_r} \mathbf{F_R}(\frac{K_r}{2}, \mathbf{p}, \mathbf{q}_i)$;
      else
        $\mathbf{F'_R} \leftarrow \sum_{i \in N_r} \mathbf{F_R}(K_r, \mathbf{p}, \mathbf{q}_i)$;
      end
    end
  end
else
  $\mathbf{F_A} \leftarrow K_{EE} \frac{\mathbf{G}-\mathbf{p}}{\|\mathbf{G}-\mathbf{p}\|}$;
  $\mathbf{F'_R} \leftarrow \sum_{i \in N_r} \mathbf{F_R}(K_r, \mathbf{p}, \mathbf{q}_i)$;
end
return $K_{EE} \frac{\mathbf{F_A}+\mathbf{F'_R}}{\|\mathbf{F_A}+\mathbf{F'_R}\|}$;

Algorithm 10: EE algorithm.

Input : $K_{PCC\text{-}EE}$: force magnitude;
$\mathbf{G}_1, \ldots, \mathbf{G}_n$: a list of $n \geq 2$ circular target region centres;
$s_1, \ldots, s_n$: a list of $n \geq 2$ target region radii;
j: the current target index;
r_σ, r_γ: radius of the free region and outer radius of the danger region, respectively;
δ, δ_{comm}: radius of the α-area and of communication, respectively;
T_{comm}, T_η: number of iterations before sending a message and for testing if a *waiting* robot changes state, respectively;
α_w, α_l: angle of the α-area to switch to *waiting* and *locked* states, respectively;
ρ_I: Probability of impatience.
ω: angle of the entry region;
$\mathbf{F_R}(K_r, \mathbf{p}, \mathbf{q})$: a repulsive force with multiplicative constant K_r, robot position $\mathbf{p}$ and obstacle position $\mathbf{q}$.
Output: a force vector;

Let N_r be the set of robots locally sensed and $\mathbf{q}_i$ the neighbour robot i position;
if $\|\mathbf{p} - \mathbf{G}\| < D$ *and the robot is not in entry region by (2.1) and* $\|\mathbf{p} - \mathbf{G}\| > r_\gamma$ **then**
  Calculate $\mathbf{w}$ by (2.2);
  $\mathbf{F_A} \leftarrow K_{PCC\text{-}EE} \frac{\mathbf{w}-\mathbf{p}}{\|\mathbf{w}-\mathbf{p}\|}$;
  $\mathbf{F'_R} \leftarrow \sum_{i \in N_r} \mathbf{F_R}(K_r, \mathbf{p}, \mathbf{q}_i)$;
else
  $\mathbf{F_A} \leftarrow$ PCC($K_{PCC\text{-}EE}$,$\mathbf{G}_1, \ldots, \mathbf{G}_n$,$s_1, \ldots, s_n$,$j$,$r_\sigma, r_\gamma$, δ, δ_{comm},T_{comm}, T_η, α_w, α_l,ρ_I);
  if *the robot is in entry region by (2.1) and* $\|\mathbf{p} - \mathbf{G}\| > r_\gamma$ *and did not reach the target area* **then**
    Let $N_r^1 \subset N_r$ be the set of robots that pushes the robot away from the entry region and $N_r^2 = N_r - N_r^1$;
    $\mathbf{F'_R} \leftarrow \sum_{i \in N_r^1} \mathbf{F_R}(\frac{K_r}{2}, \mathbf{p}, \mathbf{q}_i) + \sum_{i \in N_r^2} \mathbf{F_R}(K_r, \mathbf{p}, \mathbf{q}_i)$;
  else
    if *the robot reached the target area* **then**
      $\mathbf{F'_R} \leftarrow \sum_{i \in N_r} \mathbf{F_R}(\frac{K_r}{2}, \mathbf{p}, \mathbf{q}_i)$;
    else
      $\mathbf{F'_R} \leftarrow \sum_{i \in N_r} \mathbf{F_R}(K_r, \mathbf{p}, \mathbf{q}_i)$;
    end
  end
end

Algorithm 11: PCC-EE algorithm (continues on Algorithm 12).

return $K_{PCC\text{-}EE}\frac{\mathbf{F_A}+\mathbf{F'_R}}{\|\mathbf{F_A}+\mathbf{F'_R}\|}$;

Algorithm 12: PCC-EE algorithm (continuation).

Appendix B

Proofs of Chapter 4

B.1 Proof of Lemma 1

Let $l_\delta(t) = \sqrt{(x_1(t) - x_2(t))^2 + (y_1(t) - y_2(t))^2}$ be the distance between the two robots. The robots must maintain their minimum distance d at all time:

$$\forall t \in \mathbb{R}, l_\delta(t) \geq d. \tag{B.1}$$

To avoid a collision, $\delta \neq \pi$, which corresponds to the case where robots face each other exactly. As a result, $\cos(\delta) \neq -1$. For ease of calculation, define $X = \tau v$, that is, the distance between Robot 1 and Robot 2 when Robot 1 reaches the target. Also define $P_\delta(t) = l_\delta(t)^2 - d^2$, so the constraint in (B.1) for the distance between them is expressed by

$$\forall t \in \mathbb{R}, l_\delta(t) \geq d \Leftrightarrow \forall t \in \mathbb{R}, P_\delta(t) \geq 0.$$

Additionally,

$$\begin{aligned}
P_\delta(t) &= (vt\cos(\delta_1) - v(t-\tau)\cos(\delta_2))^2 + (vt\sin(\delta_1) - v(t-\tau)\sin(\delta_2))^2 - d^2 \\
&= (vt\cos(\delta_1) - (vt - X)\cos(\delta_2))^2 + (vt\sin(\delta_1) - (vt - X)\sin(\delta_2))^2 - d^2 \\
&= (vt)^2\cos(\delta_1)^2 - 2vt\cos(\delta_1)(vt - X)\cos(\delta_2) + (vt - X)^2\cos(\delta_2)^2 + \\
&\quad (vt)^2\sin(\delta_1)^2 - 2vt\sin(\delta_1)(vt - X)\sin(\delta_2) + (vt - X)^2\sin(\delta_2)^2 - d^2 \\
&= (vt)^2 - 2vt(vt - X)(\cos(\delta_1)\cos(\delta_2) + \sin(\delta_1)\sin(\delta_2)) + (vt - X)^2 - d^2 \\
&= (vt)^2 - 2vt(vt - X)\cos(\delta_2 - \delta_1) + (vt - X)^2 - d^2 \\
&= (vt)^2 - 2vt(vt - X)\cos(\delta) + (vt - X)^2 - d^2 \\
&= (vt)^2 - 2vt(vt - X) + 2vt(vt - X) - 2vt(vt - X)\cos(\delta) + (vt - X)^2 - d^2 \\
&= (vt)^2 - 2vt(vt - X) + (vt - X)^2 + 2vt(vt - X) - 2vt(vt - X)\cos(\delta) - d^2 \\
&= (vt - (vt - X))^2 + 2vt(vt - X) - 2vt(vt - X)\cos(\delta) - d^2
\end{aligned}$$

$$
\begin{aligned}
&= X^2 + 2vt(vt - X) - 2vt(vt - X)\cos(\delta) - d^2 \\
&= 2vt(vt - X) - 2vt(vt - X)\cos(\delta) + X^2 - d^2 \\
&= 2(vt)^2 - 2Xvt - 2(vt)^2\cos(\delta) + 2Xvt\cos(\delta) + X^2 - d^2 \\
&= 2(vt)^2 - 2(vt)^2\cos(\delta) - 2Xvt + 2Xvt\cos(\delta) + X^2 - d^2 \\
&= 2(1 - \cos(\delta))(vt)^2 - 2X(1 - \cos(\delta))vt + X^2 - d^2 \\
&= 2(1 - \cos(\delta))v^2t^2 - 2X(1 - \cos(\delta))vt + X^2 - d^2,
\end{aligned}
$$

where $\cos(\delta) = \cos(\delta_2 - \delta_1) = \cos(\delta_2)\cos(\delta_1) + \sin(\delta_2)\sin(\delta_1)$ is used.

Two cases are identified:

1. Case 1: $\cos(\delta) \neq 1$. Then $P_\delta(t)$ is a second-degree polynomial in t. It is of the form at^2+bt+c with $a = 2(1-\cos(\delta))v^2$, $b = -2X(1-\cos(\delta))v$ and $c = X^2-d^2$. $P_\delta(t)$ has a with positive sign for all t, because $(1-\cos(\delta)) > 0$ when $\cos(\delta) \neq 1$. Thus, as $a > 0$, by second-degree polynomial inequalities properties, $P_\delta(t) \geq 0$ for all t if and only if its discriminant $\Delta = b^2 - 4ac$ is negative, that is,

$$\forall t \in \mathbb{R}, P_\delta(t) \geq 0 \Leftrightarrow b^2 - 4ac \leq 0.$$

Thus:

$$
\begin{aligned}
&2^2X^2(1 - \cos(\delta))^2v^2 - 4 \cdot 2(1 - \cos(\delta))v^2(X^2 - d^2) \leq 0 \Leftrightarrow \\
&(4(1 - \cos(\delta))v^2)(X^2(1 - \cos(\delta)) - 2(X^2 - d^2)) \leq 0 \Rightarrow \\
&X^2(1 - \cos(\delta)) - 2(X^2 - d^2) \leq 0 \Leftrightarrow 2d^2 - X^2(1 + \cos(\delta)) \leq 0 \Leftrightarrow \\
&\frac{2d^2}{1 + \cos(\delta)} \leq X^2 \Rightarrow
\end{aligned}
$$

$$X \geq d\sqrt{\frac{2}{1 + \cos(\delta)}} \tag{B.2}$$

2. Case 2: $\cos(\delta) = 1$. Then $P_\delta(t) = X^2 - d^2$. In this case, $P_\delta(t) \geq 0$ for all t when $X^2 - d^2 \geq 0 \Rightarrow X \geq d$. This is the same as using $\cos(\delta) = 1$ in (B.2).

Hence, (B.2) gives, for the robots to respect the minimum distance d for every time t, a relation between the minimum distance, the angle between the lanes and the distance between Robot 1 and Robot 2 when Robot 1 reaches the target. The final result is obtained by noticing that (B.2) is equivalent to

$$\tau \geq \frac{d}{v}\sqrt{\frac{2}{1 + \cos(\delta)}}.$$

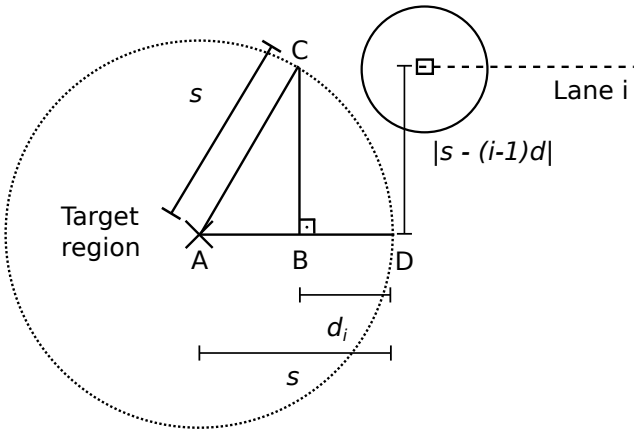


Figure B.1: The distance from the target region to a robot at the beginning of the Lane i is equal to d_i (represented by $\overline{BD}$).

B.2 Proof of Proposition 4

When robots move in straight lines in a single lane, the optimal throughput is $\frac{v}{d}$ (Proposition 1). Since $s \geq \frac{d}{2}$, multiple straight line lanes can be parallel to each other (Figure 4.4).

As the robots are going to a circular target region, the robots next to the centre reach the region in a shorter time than the others. The first robot of each lane must run an additional distance d_i from the beginning of its lane, which is related to its y-coordinate. Figure B.1 illustrates this distance for a robot in Lane i. The right triangle ABC has hypotenuse $\overline{AC}$ measuring s, so the horizontal cathetus $\overline{AB}$ measures $\sqrt{s^2-(s-(i-1)d)^2}$. Consequently, the robot in the Lane i needs to walk an additional distance represented by $\overline{BD}$, which has $d_i = |\overline{BD}| = s - |\overline{AB}| = s - \sqrt{s^2-(s-(i-1)d)^2}$ units of length.

This distance is minimised when $|\overline{BD}| = 0$, that would happen if $i = \frac{s}{d} + 1$. However, i must be integer, so $|\overline{BD}|$ is minimised by an integer J that minimises d_J. If $\frac{s}{d} \notin \mathbb{Z}$, the two nearest integers are $\lfloor \frac{s}{d} \rfloor$ and $\lceil \frac{s}{d} \rceil$. Thus, if $J = \lfloor \frac{s}{d} \rfloor + 1$ then, equivalently,

$$d_{\lfloor \frac{s}{d} \rfloor + 1} \leq d_{\lceil \frac{s}{d} \rceil + 1} \Leftrightarrow s - \sqrt{s^2 - \left(s - \left\lfloor \frac{s}{d} \right\rfloor d\right)^2} \leq s - \sqrt{s^2 - \left(s - \left\lceil \frac{s}{d} \right\rceil d\right)^2} \Leftrightarrow$$

$$\sqrt{s^2 - \left(s - \left\lceil \frac{s}{d} \right\rceil d\right)^2} \leq \sqrt{s^2 - \left(s - \left\lfloor \frac{s}{d} \right\rfloor d\right)^2} \Leftrightarrow s^2 - \left(s - \left\lceil \frac{s}{d} \right\rceil d\right)^2 \leq s^2 - \left(s - \left\lfloor \frac{s}{d} \right\rfloor d\right)^2 \Leftrightarrow$$

$$\left(s - \left\lfloor \frac{s}{d} \right\rfloor d\right)^2 \leq \left(s - \left\lceil \frac{s}{d} \right\rceil d\right)^2 \Leftrightarrow \left|s - \left\lfloor \frac{s}{d} \right\rfloor d\right| \leq \left|s - \left\lceil \frac{s}{d} \right\rceil d\right|$$

Thus,

$$J = \begin{cases} \left\lfloor \frac{s}{d} \right\rfloor + 1, & \text{if } \left| s - \left\lfloor \frac{s}{d} \right\rfloor d \right| \leq \left| s - \left\lceil \frac{s}{d} \right\rceil d \right|, \\ \left\lceil \frac{s}{d} \right\rceil + 1, & \text{otherwise.} \end{cases}$$

Let $N(t)$ be the number of robots that arrive at the target region until a given time t after the first robot has reached it. Thus,

$$N(t) = \sum_{i=1}^{\left\lfloor \frac{2s}{d} \right\rfloor + 1} N_i(t),$$

where $N_i(t)$ is the number of robots at Lane i that arrived at the target region by time t. As every robot has the same linear speed and started at the same x-coordinate, when the first robot at Lane J reaches the target region, all robots have run d_J units of length. Hence, at each Lane i, instead of running an additional d_i to reach the target region, they need to run $d_i - d_J$. Consequently,

$$N_i(t) = \begin{cases} \left\lfloor \frac{vt - (d_i - d_J)}{d} + 1 \right\rfloor, & \text{if } t \geq \frac{d_i - d_J}{v}, \\ 0, & \text{otherwise,} \end{cases}$$

and, by Definition 2,

$$f_p(t) = \frac{N(t) - 1}{t} = \frac{1}{t} \left(\sum_{i=1}^{\left\lfloor \frac{2s}{d} \right\rfloor + 1} N_i(t) \right) - \frac{1}{t}.$$

Also,

$$\begin{aligned} f_p = \lim_{t \to \infty} f_p(t) &= \lim_{t \to \infty} \left(\frac{1}{t} \left(\sum_{i=1}^{\left\lfloor \frac{2s}{d} \right\rfloor + 1} N_i(t) \right) - \frac{1}{t} \right) \\ &= \lim_{t \to \infty} \frac{1}{t} \sum_{i=1}^{\left\lfloor \frac{2s}{d} \right\rfloor + 1} N_i(t) = \lim_{t \to \infty} \sum_{i=1}^{\left\lfloor \frac{2s}{d} \right\rfloor + 1} \frac{N_i(t)}{t} \\ &= \lim_{t \to \infty} \sum_{i=1}^{\left\lfloor \frac{2s}{d} \right\rfloor + 1} \frac{1}{t} \left\lfloor \frac{vt - d_i + d_J}{d} + 1 \right\rfloor \qquad \left[\text{as } t \to \infty \Rightarrow t \geq \frac{d_i - d_J}{v} \right] \\ &= \lim_{t \to \infty} \sum_{i=1}^{\left\lfloor \frac{2s}{d} \right\rfloor + 1} \frac{1}{t} \left(\frac{vt - d_i + d_J}{d} + 1 - frac\left(\frac{vt - d_i + d_J}{d} + 1 \right) \right) \end{aligned}$$

$$= \lim_{t\to\infty} \sum_{i=1}^{\lfloor \frac{2s}{d} \rfloor + 1} \left(\frac{v}{d} - \frac{d_i - d_J}{dt} + \frac{1}{t} \right) - \lim_{t\to\infty} \frac{1}{t} \sum_{i=1}^{\lfloor \frac{2s}{d} \rfloor + 1} frac\left(\frac{vt - d_i + d_J}{d} + 1 \right)$$
$$= \left\lfloor \frac{2s}{d} + 1 \right\rfloor \frac{v}{d},$$

as $frac$ and d_i are bounded for every i due to $0 \leq d_i \leq s$ and $0 \leq frac(x) < 1$ for any x.

B.3 Proof of Proposition 5

Without loss of generality, consider the target at the origin of a coordinate system and that the robots are moving parallel to the x-axis. By Definition 2, the throughput considers the number of robots that cross the target during a unit of time and after the first robot has reached it. The number of robots, N_t, is evaluated during a time t. As a result, computing the maximum throughput is reduced to finding the maximum number of robots (their centre of mass) that can fit in a rectangle of width $w(t) = vt$ and height $h = 2s$.

Figure B.2 illustrates how N_t is calculated. Robots are represented by black dots in hexagonal formation and distant by d. In (I), only the first robot reached the target. In (II), all robots not in the dashed area arrived in the target region before the last robot. The robots on the right dashed area should not be counted because their arrival time is greater than the arrival time of the last robot. Hence, they arrive after the considered time frame t. That is, all robots on the dashed area in (I) should be counted as part of the number of robots that reached the target region in the time between the first and the last robot, while the robots on the dashed area in (II) should not. As these dashed areas have the same value, this proof considers the number of robots inside a rectangle $vt \times 2s$. Then, the dashed area used for counting in (II) replaces the unconsidered robots in the dashed area in (I). As $t \to \infty$ is of concern, any possible difference between the number of robots on the dashed areas of either side due to the configuration of the hexagonal packing is negligible.

Due to the constraint that robots must be at a distance d from each other, consider discs of radius $d/2$ as reserved areas for each robot, and any two reserved areas must not intersect. Therefore, the problem is equivalent to finding the optimal arrangement of circles of radius $d/2$ in a rectangle of width $W(t) = w(t) + 2\frac{d}{2} = vt + d$ and height $H = h + 2\frac{d}{2} = 2s + d$. This formulation is a variant of the *circle packing* problem, which is already well studied. (See `http://packomania.com/` and `http://hydra.nat.uni-magdeburg.de/packing/crc_var/crc.html`, acessed on 16 November 2021, for an informal introduction.) The term $2\frac{d}{2}$ was added because the circle packing problem deals with full circles, not their centres.

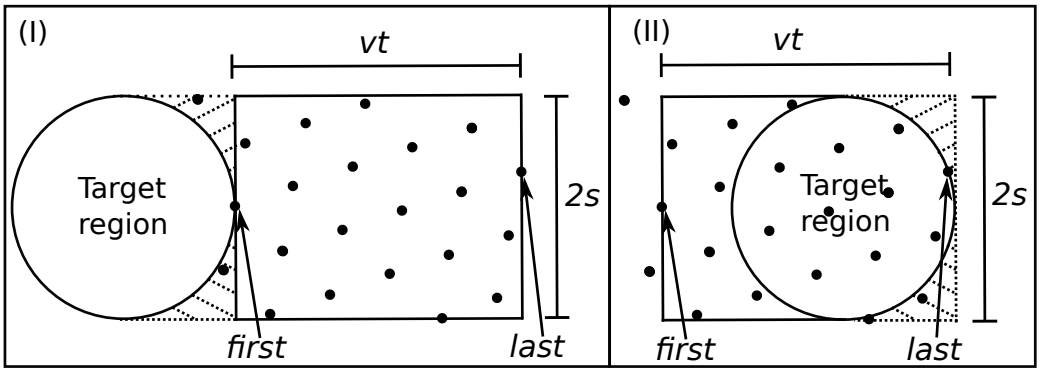


Figure B.2: The rectangular area regarding the calculation of N_t over time.

The optimal surface occupied by the circles divided by the rectangle area was proven to be $\pi\sqrt{3}/6$ in the case of hexagonal packing over an infinite area (Chang and Wang, 2010). Thus, the total area occupied by the circles representing the reserved areas of the robots is given by $\left(\pi\sqrt{3}/6\right) HW(t)$. Hence, the maximum number of robots N_t that can fit inside the $HW(t)$ area is bounded by $N_{opt}(t) \geq N_t$, for

$$N_{opt}(t) = \left\lfloor \frac{\left(\pi\sqrt{3}/6\right) HW(t)}{\pi d^2/4} \right\rfloor = \left\lfloor \frac{2HW(t)}{\sqrt{3}d^2} \right\rfloor.$$

By Definition 2, the maximum throughput is

$$f_h^{max}(t) = \frac{N_{opt}(t) - 1}{t} = \frac{\left\lfloor \frac{2HW(t)}{\sqrt{3}d^2} \right\rfloor - 1}{t}.$$

As for any x, $\lfloor x \rfloor = x - frac(x)$ and $0 \leq frac(x) < 1$, the upper bound of the asymptotic throughput is

$$\begin{aligned}
f_h^{max} &= \lim_{t\to\infty} f_h^{max}(t) = \lim_{t\to\infty} \frac{\left\lfloor \frac{2HW(t)}{\sqrt{3}d^2} \right\rfloor - 1}{t} = \lim_{t\to\infty} \frac{2HW(t)}{\sqrt{3}d^2 t} = \lim_{t\to\infty} \frac{2(h+d)(w(t)+d)}{\sqrt{3}d^2 t} \\
&= \lim_{t\to\infty} \frac{2(2s+d)(vt+d)}{\sqrt{3}d^2 t} = \lim_{t\to\infty} \frac{2}{\sqrt{3}} \frac{2s+d}{d^2} \frac{vt+d}{t} \\
&= \lim_{t\to\infty} \frac{2}{\sqrt{3}} \left(\frac{2s}{d^2} + \frac{d}{d^2}\right) \left(\frac{vt}{t} + \frac{d}{t}\right) = \lim_{t\to\infty} \frac{2}{\sqrt{3}} \left(\frac{2s}{d^2} + \frac{1}{d}\right) \left(v + \frac{d}{t}\right) \\
&= \frac{2}{\sqrt{3}} \left(\frac{2s}{d^2} + \frac{1}{d}\right) v = \frac{2}{\sqrt{3}} \left(\frac{2s}{d} + 1\right) \frac{v}{d}.
\end{aligned}$$

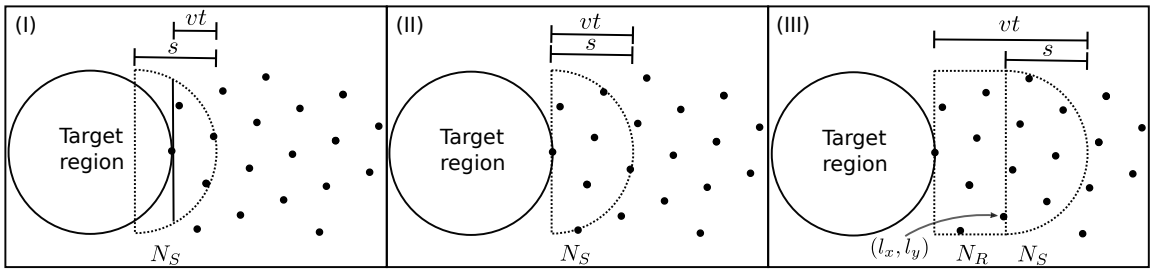


Figure B.3: Arrival of the robots on the target region over time.

B.4 Proof of Proposition 6

This proof concerns about the throughput of the target region for a given time and hexagonal packing angle θ, $f_h(t,\theta) = \frac{N(t,\theta)-1}{t}$, where $N(t,\theta)$ denotes the number of robots which arrived at the target region. Figure B.3 illustrates the arrival of the robots on the target region.

In Figure B.3 (I), when the robots – here represented by black dots – in hexagonal packing begin to arrive at the target region, only the robots inside a part of the semicircle are counted. In Figure B.3 (II), consider the first robot to reach the target region being at (x_0, y_0) at time 0. As t grows, this continues until $vt = s$. In Figure B.3 (III), when $vt > s$, the robots are counted on two regions: a rectangular, N_R, and a semicircular, N_S. When $vt > s$, the semicircular region counting starts after the last robot on the rectangular region located at (l_x, l_y).

As this region has a circular shape, not all robots at the distance vt arrive at target region by the time t. Thence, the number of robots in hexagonal packing are divided into the number of robots located inside a rectangle, N_R, and of robots inside a semicircle N_S (Figure B.3 (III)). That is, $N(t,\theta) = N_S(t,\theta) + N_R(t,\theta)$ and $N_R = 0$ whenever $vt \leq s$.

This proof is divided in lemmas for helping the construction of the equation to compute $N_R(t,\theta)$ and $N_S(t,\theta)$ as well for calculating $\lim_{t\to\infty} f_h(t,\theta)$. Before presenting them, it is discussed a coordinate space transformation which will be used to count the robots for N_R and N_S. This transformation was inspired by (Red Blob Games, 2021).

Figure B.4 shows the coordinate spaces used in this proof: the usual Euclidean space (x, y) in relation to the target region and the rectangle region formed by robots in hexagonal packing going to it; the coordinate space (x_g, y_g), formed by the usual space after a translation to the first robot to reach the target region at (x_0, y_0), followed by a rotation by $-\psi$; the coordinate space (x_h, y_h), a hexagonal grid coordinate space made after this transformation and a linear transformation H. Robots are represented by the black dots and they are on hexagonal formation. Each neighbour of a robot is

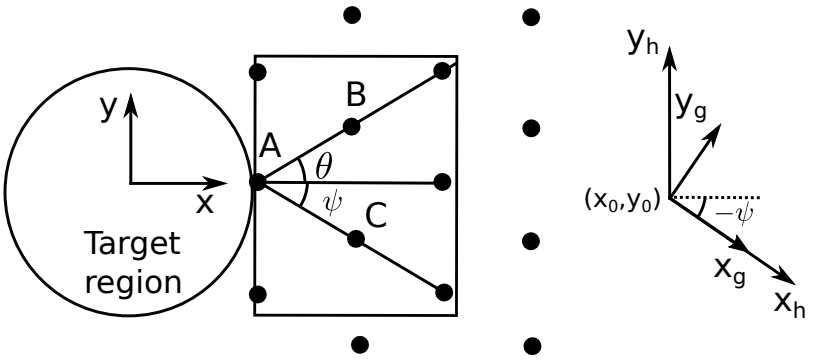


Figure B.4: The reference frames used in this proof.

distant by d, so $\triangle ABC$ is equilateral. Thus, $\theta + \psi = \pi/3$.

Let $\psi = \pi/3 - \theta$ (because the angle of the equilateral triangle formed by neighbours is $\pi/3$, as explained in Figure B.4). Accordingly, $\psi \in [0, \pi/3)$, too. The usual Euclidean coordinate space which represents the location of all robots is denoted here by (x, y) coordinates. The next coordinate space is denoted by (x_g, y_g), and it is the result of a translation of the usual Euclidean coordinate space by the position of the first robot to reach the target region at (x_0, y_0), then a rotation of $-\psi$, that is,

$$\begin{bmatrix} x_g \\ y_g \end{bmatrix} = \begin{bmatrix} \cos(-\psi) & -\sin(-\psi) \\ \sin(-\psi) & \cos(-\psi) \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix}.$$

The last coordinate space is denoted by (x_h, y_h), and it is intended to represent a hexagonal grid such that the position of each robot is an integer pair.

Figure B.5 shows an example of the location of robots with respect to that hexagonal grid. Robots are located in the line representing the y_h-axis and in the lines named l, m, n, o, p and q, which are parallel to the y_h-axis. In this example, $\psi = 0.227$ and the distance between all robots is $d = 0.5$. The distance between those parallel-to-y_h lines is $\sqrt{3}d/2$. The robots inside the rectangle $EFGH$ are counted and are indicated by red points, while blue points are robots outside the rectangle. Although the x_h-axis coincides with the x_g-axis, x_h is scaled by d.

Let $(x_h, y_h) \in \mathbb{Z}^2$ be the hexagonal coordinates of a robot in this hexagonal grid space. In this figure, there is an integer grid in grey – the horizontal lines correspond to fixed integer y_h values and the inclined ones, x_h values. For example, in Figure B.5 robots R_{10}, R_{11} and R_{20} respectively are at $(0, 1)$, $(1, 1)$ and $(1, 0)$ at (x_h, y_h) coordinate system, which is equivalent to $\left(-1/4, \sqrt{3}/4\right)$, $\left(1/4, \sqrt{3}/4\right)$ and $(1/2, 0)$ on the usual two dimensional coordinate system with origin at (x_0, y_0).

A linear transformation H from a point (x_h, y_h) to (x_g, y_g) basis is obtained by knowing the result of this transformation for the standard vectors $(1, 0)$ and $(0, 1)$. Observing Figure B.5 and having that the angle between the x-axis and y_h-axis is

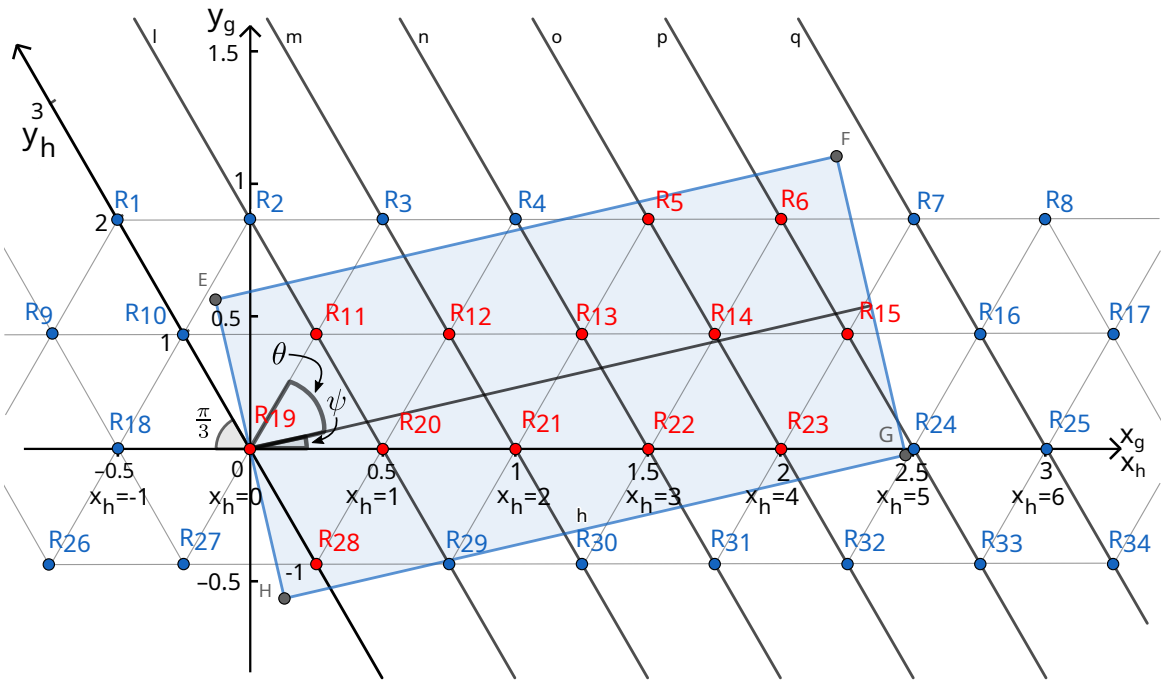


Figure B.5: Example of robots in hexagonal packing formation and the corresponding rectangular corridor which will reach the target region. The axes of the coordinate spaces (x_g, y_g) and (x_h, y_h) overlap. Red robots are inside the rectangle $EFGH$, blue robots are outside it, and the first robot to reach the target area is located at $(x_g, y_g) = (0, 0)$ – equivalently $(x_h, y_h) = (0, 0)$.

by definition $2\pi/3$, one gets the following mappings $(x_h, y_h) \mapsto (x_g, y_g)$: $(1,0) \mapsto (d,0)$ and $(0,1) \mapsto (d\cos(2\pi/3), d\sin(2\pi/3)) = (-\frac{d}{2}, \frac{\sqrt{3}d}{2})$ (in Figure B.5 these two mappings are represented by robots R_{20} and R_{10}, respectively, with $d = 0.5$). Then,

$$\begin{bmatrix} x_g \\ y_g \end{bmatrix} = \begin{bmatrix} H\left(\begin{bmatrix}1\\0\end{bmatrix}\right) & H\left(\begin{bmatrix}0\\1\end{bmatrix}\right) \end{bmatrix} \begin{bmatrix} x_h \\ y_h \end{bmatrix} = \begin{bmatrix} d & -\frac{d}{2} \\ 0 & \frac{\sqrt{3}d}{2} \end{bmatrix} \begin{bmatrix} x_h \\ y_h \end{bmatrix}. \tag{B.3}$$

Counting the robots inside the rectangle is the same as counting the number of integer hexagonal coordinate points lying inside it. Figure B.6 shows the rectangular part with some robots in hexagonal packing, where the robots are the red dots, and the hexagonal packing is guided by the grey lines inside the rectangle based on the value of the angle ψ. The rectangle has width $vt-s$ and height $2s$. The reference frame of the hexagonal grid is rotated about the target region (Figure B.4). The problem involves the rectangle $EFGH$ in a hexagonal grid (grey lines inside the rectangle) of robots (the red dots). The x_h-axis is horizontal and coincides with the x-axis. The y_h-axis forms a $2\pi/3$ angle with it. $\overline{EH}$ and $\overline{AB}$ have length $2s$ and $vt - s$, respectively. In this example, $\psi = 19\pi/180$ and $s = 1$. The angles marked with a line are equal to ψ, because the angle formed by $\overrightarrow{y_2 A}$ and the x-axis is right, as well as $\widehat{EAB}$. Accordingly, $\widehat{y_2AB} = \pi/2 - \psi$ implies that $\widehat{EAy_2} = \psi$.

From Figure B.6,

$$2s = (y_2 - y_1)\cos(\psi),\ y_2 = \frac{s}{\cos(\psi)} \text{ and } y_1 = -\frac{s}{\cos(\psi)}. \tag{B.4}$$

Consider a robot with coordinates (x_g, y_g). The four sides of the rectangle $EFGH$, $\overline{HG}$, $\overline{EF}$, $\overline{EH}$ and $\overline{FG}$, have the following equations of line: $y_g = y_1 + \tan(\psi)x_g$, $y_g = y_2 + x_g\tan(\psi)$, $y_g = \tan\left(\psi + \frac{\pi}{2}\right)x_g$ and $y_g = \tan(\psi + \frac{\pi}{2})\left(x_g - \frac{vt-s}{\cos(\psi)}\right)$, respectively. The term $\frac{vt-s}{\cos(\psi)}$ in the last equation arises because of the length of $\overline{AC}$, which is the hypotenuse of $\triangle ABC$ whose side $\overline{AB}$ measures vt. Knowing that $\tan\left(\psi + \frac{\pi}{2}\right) = -\cot(\psi)$, the equations below are all true for a robot at (x_g, y_g) to be inside or on the boundary of the previously defined rectangle,

$$\begin{aligned} &y_g \geq y_1 + x_g\tan(\psi),\ y_g \leq y_2 + x_g\tan(\psi),\ -x_g \leq \tan(\psi)y_g, \text{ and} \\ &-\left(x_g - \frac{vt-s}{\cos\psi}\right) \geq \tan(\psi)y_g. \end{aligned} \tag{B.5}$$

Now take the minimum and maximum y_h value for each parallel-to-y_h line depending on the x_h value. Using (B.3) for converting (B.5) to x_h and y_h coordinate system, i.e., hexagonal coordinates, one obtains for $\overline{HG}$ and $\overline{EF}$

$$\left(\frac{\sqrt{3}}{2} + \frac{1}{2}\tan(\psi)\right)y_h - \tan(\psi)x_h \geq \frac{y_1}{d} \text{ and}$$

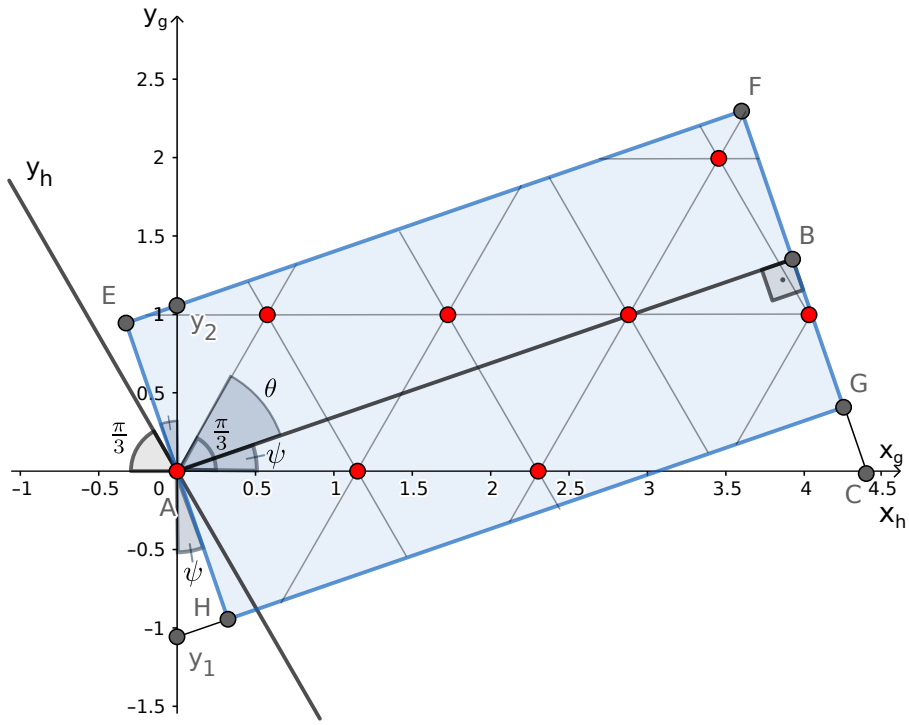


Figure B.6: Example of counting robots in hexagonal packing with rotation in the reference frame.

$$\left(\frac{\sqrt{3}}{2}+\frac{1}{2}\tan(\psi)\right)y_h-\tan(\psi)x_h\leq\frac{y_2}{d}.$$

Hence,

$$\frac{y_1}{d}\leq\left(\frac{\sqrt{3}}{2}+\frac{1}{2}\tan\psi\right)y_h-\tan(\psi)x_h\leq\frac{y_2}{d}\Leftrightarrow$$
$$\frac{\frac{2y_1}{d}+2\tan(\psi)x_h}{\sqrt{3}+\tan(\psi)}\leq y_h\leq\frac{\frac{2y_2}{d}+2\tan(\psi)x_h}{\sqrt{3}+\tan(\psi)}. \tag{B.6}$$

Analogously, but considering $\overline{EH}$ and $\overline{FG}$,

$$-x_h\leq\left(\tan(\psi)\frac{\sqrt{3}}{2}-\frac{1}{2}\right)y_h \text{ and } \left(\tan(\psi)\frac{\sqrt{3}}{2}-\frac{1}{2}\right)y_h\leq\frac{vt-s}{d\cos(\psi)}-x_h. \tag{B.7}$$

Based on the sign of $\left(\tan(\psi)\frac{\sqrt{3}}{2}-\frac{1}{2}\right)$ and excluding the null case (when $\psi=\pi/6$), there are two different inequalities over y_h. Assuming $\psi\in[0,\pi/3)$, $\left(\tan(\psi)\frac{\sqrt{3}}{2}-\frac{1}{2}\right)>0\Leftrightarrow\tan(\psi)\frac{\sqrt{3}}{2}>\frac{1}{2}\Leftrightarrow\tan(\psi)>\frac{1}{\sqrt{3}}\Leftrightarrow\psi>\pi/6$. Thus, from (B.7),

$$\begin{aligned}&\frac{-x_h}{\frac{\sqrt{3}\tan(\psi)-1}{2}}\leq y_h\leq\frac{\frac{vt-s}{d\cos(\psi)}-x_h}{\frac{\sqrt{3}\tan(\psi)-1}{2}}, \text{ if } \psi>\pi/6,\\ &\frac{\frac{vt-s}{d\cos(\psi)}-x_h}{\frac{\sqrt{3}\tan(\psi)-1}{2}}\leq y_h\leq\frac{-x_h}{\frac{\sqrt{3}\tan(\psi)-1}{2}}, \text{ if } \psi<\pi/6.\end{aligned} \tag{B.8}$$

$$\begin{aligned}&\frac{-2x_h}{\sqrt{3}\tan(\psi)-1}\leq y_h\leq\frac{\frac{2(vt-s)}{d\cos(\psi)}-2x_h}{\sqrt{3}\tan(\psi)-1}, \text{ if } \psi>\pi/6,\\ &\frac{\frac{2(vt-s)}{d\cos(\psi)}-2x_h}{\sqrt{3}\tan(\psi)-1}\leq y_h\leq\frac{-2x_h}{\sqrt{3}\tan(\psi)-1}, \text{ if } \psi<\pi/6.\end{aligned} \tag{B.9}$$

(B.6) and (B.9) restrict the value of y_h depending on the value of x_h by the relation

$$
\begin{aligned}
&\max\left(\frac{\frac{2y_1}{d}+2\tan(\psi)x_h}{\sqrt{3}+\tan(\psi)},\frac{-2x_h}{\sqrt{3}\tan(\psi)-1}\right)\leq y_h\\
&\leq\min\left(\frac{\frac{2y_2}{d}+2\tan(\psi)x_h}{\sqrt{3}+\tan(\psi)},\frac{\frac{2(vt-s)}{d\cos(\psi)}-2x_h}{\sqrt{3}\tan(\psi)-1}\right),\text{ if }\psi>\pi/6,\\
&\max\left(\frac{\frac{2y_1}{d}+2\tan(\psi)x_h}{\sqrt{3}+\tan(\psi)},\frac{\frac{2(vt-s)}{d\cos(\psi)}-2x_h}{\sqrt{3}\tan(\psi)-1}\right)\leq y_h\\
&\leq\min\left(\frac{\frac{2y_2}{d}+2\tan(\psi)x_h}{\sqrt{3}+\tan(\psi)},\frac{-2x_h}{\sqrt{3}\tan(\psi)-1}\right),\text{ if }\psi<\pi/6.
\end{aligned}
\tag{B.10}
$$

Using hexagonal coordinates the position of each robot is represented by a pair of integers. Then, assuming x_h and y_h integers, (B.10) becomes $\lceil Y_1^R(x_h)\rceil\leq y_h\leq\lfloor Y_2^R(x_h)\rfloor$, for

$$
\begin{aligned}
Y_1^R(x_h)&=\begin{cases}\max\left(\frac{\frac{2y_1}{d}+2\tan(\psi)x_h}{\sqrt{3}+\tan(\psi)},\frac{-2x_h}{\sqrt{3}\tan(\psi)-1}\right), & \text{if }\psi>\pi/6,\\ \max\left(\frac{\frac{2y_1}{d}+2\tan(\psi)x_h}{\sqrt{3}+\tan(\psi)},\frac{\frac{2(vt-s)}{d\cos(\psi)}-2x_h}{\sqrt{3}\tan(\psi)-1}\right), & \text{if }\psi<\pi/6,\\ \frac{\sqrt{3}y_1+dx_h}{2d}, & \text{if }\psi=\pi/6,\end{cases}\\
&=\begin{cases}\max\left(\frac{\frac{2y_1}{d}+2\tan(\psi)x_h}{\sqrt{3}+\tan(\psi)},\frac{-2x_h}{\sqrt{3}\tan(\psi)-1}\right), & \text{if }\psi>\pi/6,\\ \max\left(\frac{\frac{2y_1}{d}+2\tan(\psi)x_h}{\sqrt{3}+\tan(\psi)},\frac{\frac{2(vt-s)}{d\cos(\psi)}-2x_h}{\sqrt{3}\tan(\psi)-1}\right), & \text{if }\psi<\pi/6,\\ \frac{\frac{-\sqrt{3}s}{\cos(\pi/6)}+dx_h}{2d}, & \text{if }\psi=\pi/6,\end{cases}\\
&=\begin{cases}\max\left(\frac{\frac{2y_1\cos(\psi)}{d}+2\sin(\psi)x_h}{\sqrt{3}\cos(\psi)+\sin(\psi)},\frac{-2x_h\cos(\psi)}{\sqrt{3}\sin(\psi)-\cos(\psi)}\right), & \text{if }\psi>\pi/6,\\ \max\left(\frac{\frac{2y_1\cos(\psi)}{d}+2\sin(\psi)x_h}{\sqrt{3}\cos(\psi)+\sin(\psi)},\frac{\frac{2(vt-s)}{d}-2x_h\cos(\psi)}{\sqrt{3}\sin(\psi)-\cos(\psi)}\right), & \text{if }\psi<\pi/6,\\ \frac{\frac{-2\sqrt{3}s}{\sqrt{3}}+dx_h}{2d}, & \text{if }\psi=\pi/6,\end{cases}
\end{aligned}
$$

$$
= \begin{cases} \max\left(\dfrac{\frac{-2s}{d} + 2\sin(\psi)x_h}{\sqrt{3}\cos(\psi) + \sin(\psi)}, \dfrac{-2x_h\cos(\psi)}{\sqrt{3}\sin(\psi) - \cos(\psi)}\right), & \text{if } \psi > \pi/6, \\ \max\left(\dfrac{\frac{-2s}{d} + 2\sin(\psi)x_h}{\sqrt{3}\cos(\psi) + \sin(\psi)}, \dfrac{\frac{2(vt-s)}{d} - 2x_h\cos(\psi)}{\sqrt{3}\sin(\psi) - \cos(\psi)}\right), & \text{if } \psi < \pi/6, \\ \dfrac{-2s + dx_h}{2d}, & \text{if } \psi = \pi/6, \end{cases}
$$

$$
= \begin{cases} \max\left(\dfrac{\sin(\psi)x_h - \frac{s}{d}}{\cos\left(\frac{\pi}{6} - \psi\right)}, \dfrac{-\cos(\psi)x_h}{\sin\left(\psi - \frac{\pi}{6}\right)}\right), & \text{if } \psi > \pi/6, \\ \max\left(\dfrac{\sin(\psi)x_h - \frac{s}{d}}{\cos\left(\frac{\pi}{6} - \psi\right)}, \dfrac{\frac{vt-s}{d} - \cos(\psi)x_h}{\sin\left(\psi - \frac{\pi}{6}\right)}\right), & \text{if } \psi < \pi/6, \\ \dfrac{x_h}{2} - \dfrac{s}{d}, & \text{if } \psi = \pi/6, \end{cases}
$$

$$
= \begin{cases} \max\left(\dfrac{d\sin(\psi)x_h - s}{d\cos\left(\frac{\pi}{6} - \psi\right)}, \dfrac{-\cos(\psi)x_h}{\sin\left(\psi - \frac{\pi}{6}\right)}\right), & \text{if } \psi > \pi/6, \\ \max\left(\dfrac{d\sin(\psi)x_h - s}{d\cos\left(\frac{\pi}{6} - \psi\right)}, \dfrac{vt - s - d\cos(\psi)x_h}{d\sin\left(\psi - \frac{\pi}{6}\right)}\right), & \text{if } \psi < \pi/6, \\ \dfrac{x_h}{2} - \dfrac{s}{d}, & \text{if } \psi = \pi/6, \end{cases} \tag{B.11}
$$

$$
Y_2^R(x_h) = \begin{cases} \min\left(\dfrac{\frac{2y_2}{d} + 2\tan(\psi)x_h}{\sqrt{3} + \tan(\psi)}, \dfrac{\frac{2(vt-s)}{d\cos(\psi)} - 2x_h}{\sqrt{3}\tan(\psi) - 1}\right), & \text{if } \psi > \pi/6, \\ \min\left(\dfrac{\frac{2y_2}{d} + 2\tan(\psi)x_h}{\sqrt{3} + \tan(\psi)}, \dfrac{-2x_h}{\sqrt{3}\tan(\psi) - 1}\right), & \text{if } \psi < \pi/6, \\ \dfrac{\sqrt{3}y_2 + dx_h}{2d}, & \text{if } \psi = \pi/6, \end{cases} \tag{B.12}
$$

$$
= \begin{cases} \min\left(\dfrac{\sin(\psi)x_h + \frac{s}{d}}{\cos\left(\frac{\pi}{6} - \psi\right)}, \dfrac{\frac{vt-s}{d} - \cos(\psi)x_h}{\sin\left(\psi - \frac{\pi}{6}\right)}\right), & \text{if } \psi > \pi/6, \\ \min\left(\dfrac{\sin(\psi)x_h + \frac{s}{d}}{\cos\left(\frac{\pi}{6} - \psi\right)}, \dfrac{-\cos(\psi)x_h}{\sin\left(\psi - \frac{\pi}{6}\right)}\right), & \text{if } \psi < \pi/6, \\ \dfrac{x_h}{2} + \dfrac{s}{d}, & \text{if } \psi = \pi/6. \end{cases}
$$

The simplifications above used (B.4), $\cos\left(\frac{\pi}{6} - \psi\right) = \frac{\sqrt{3}}{2}\cos(\psi) + \frac{1}{2}\sin(\psi)$ and $\sin\left(\psi - \frac{\pi}{6}\right) = \frac{\sqrt{3}}{2}\sin(\psi) - \frac{1}{2}\cos(\psi)$.

Now the possible integer values for the x_h-axis which are inside the rectangle $EFGH$ are obtained, that is, the number of lines parallel with the y_h-axis that intersect the rectangle for x_h integer values is counted. Let n_l be the number of

such parallel lines. Consider $n_l = n_l^- + n_l^+$, such that n_l^- is the number of lines parallel to the y_h-axis whose intersection with the x_h-axis is a point $(i, 0)$ for $i < 0$ and $i \in \mathbb{Z}$, and n_l^+ is similar but for non-negative integer i. For example, Figure B.5 has $n_l^- = 0$ and $n_l^+ = 6$ (it is marked below the values of the points over the x-axis the equivalent over x_h-axis, in order to aid enumerating them). Note that the point $(i, 0)$ may be outside of the rectangle, but it will still be counted if there are integer (i, y_h) coordinates inside the rectangle. The next lemma shows how to compute n_l^+ and n_l^- to aid in this proof development.

Lemma 4. *On the (x_h, y_h) coordinate system, the integer values for x_h robot coordinates inside the rectangle $EFGH$ are in the set $\{-n_l^-, \ldots, n_l^+ - 1\}$ with*

$$n_l^+ = \left\lfloor \frac{2(vt - s)\cos(\psi - \pi/6) + 2s\sin(|\psi - \pi/6|)}{\sqrt{3}d} + 1 \right\rfloor, \tag{B.13}$$

and

$$n_l^- = \left\lfloor \frac{2s\sin\left(|\psi - \pi/6|\right)}{\sqrt{3}d} \right\rfloor. \tag{B.14}$$

Proof. The value of n_l^+ is obtained by counting how many parallel-to-y_h lines, when projected over the x-axis, are distant from each other by d on this axis and are inside the rectangle. These lines must intersect the diagonal $\overline{HF}$ of the rectangle, but commencing from the intersection between the y_h-axis and the diagonal, i.e., from B_0 in the Figure B.7. In this figure, B_i is the intersection of a parallel-to-y_h line on a x_h integer coordinate and the diagonal $\overline{HF}$. The triangles AD_iC_i for any $i \in \{1, 2\}$ and ADC are similar. $\overline{AD_i}$ and $\overline{AC_i}$ have distance $i \cdot d$ and $i \cdot e$, respectively. In this example, $d = 2$ and there are three points lying over $\overline{HF}$.

Let $\phi = \arctan\left(\frac{2s}{vt-s}\right)$ be the angle of the diagonal in relation to the rectangle base. There are two cases depending on the value of ψ.

- Case $\psi \leq \frac{\pi}{6}$: from Figure B.7, every line parallel to y_h is distant by d on the projection onto the x-axis. The triangles AD_iC_i for any $i \in \{1, \ldots, n_l^+ - 1\}$ and ADC are similar, $|\overline{AD_1}| = d$ and $|\overline{AC_1}| = e$, whose value is unknown for the moment. $\triangle ADC$ has angles $\widehat{CAD} = \psi + \phi$, $\widehat{ADC} = \pi/3$ and $\widehat{ACD} = \pi - \widehat{CAD} - \widehat{ADC} = 2\pi/3 - \psi - \phi$. As for every i, $\triangle AD_iC_i \sim \triangle ADC$,

$$\frac{|\overline{AC}|}{|\overline{AC_1}|} = \frac{|\overline{AD}|}{|\overline{AD_1}|} \Leftrightarrow \frac{|\overline{AC}|}{e} = \frac{|\overline{AD}|}{d}. \tag{B.15}$$

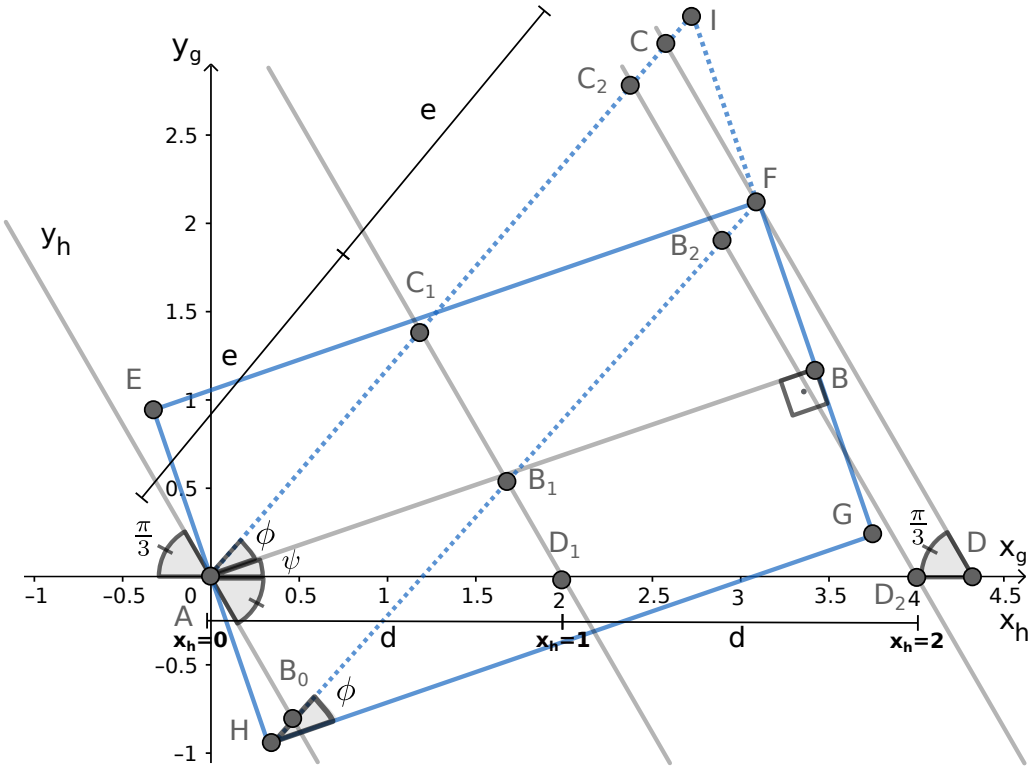


Figure B.7: Counting how many points named B_i lie in the diagonal $\overline{HF}$.

As $AHFI$ is a parallelogram, $|\overline{HF}| = |\overline{AI}|$ and $|\overline{FI}| = |\overline{AH}| = s$, then $|\overline{BI}| = 2s$. Thus, $|\overline{AI}| = \sqrt{(2s)^2 + (vt-s)^2}$, because $\triangle ABI$ is right-angled. Also, by the law of sines, $\frac{|\overline{AD}|}{\sin(\widehat{ACD})} = \frac{|\overline{AC}|}{\sin(\widehat{ADC})} \Leftrightarrow$

$$\begin{aligned} |\overline{AD}| &= |\overline{AC}|\frac{\sin(\widehat{ACD})}{\sin(\widehat{ADC})} = \left(|\overline{AI}| - |\overline{CI}|\right)\frac{\sin(\widehat{ACD})}{\sin(\widehat{ADC})} \\ &= \left(|\overline{AI}| - |\overline{CI}|\right)\frac{\sin(2\pi/3 - \psi - \phi)}{\sin(\pi/3)}. \end{aligned} \tag{B.16}$$

AB_0FC is a parallelogram as well, so $|\overline{AC}| = |\overline{B_0F}| = |\overline{HF}| - |\overline{HB_0}|$ and $|\overline{CI}| = |\overline{HB_0}|$.

The $\triangle AB_0H$ has angles $\widehat{HAB_0} = \widehat{HAB} - \widehat{B_0AB} = \widehat{HAB} - (\widehat{B_0AD} + \widehat{DAB}) = \pi/2 - (\pi/3 + \psi) = \pi/6 - \psi$, $\widehat{AHB_0} = \widehat{AHG} - \widehat{FHG} = \pi/2 - \phi$ and $\widehat{HB_0A} = \pi - \widehat{HAB_0} - \widehat{AHB_0} = \pi/3 + \psi + \phi$. By the law of sines, $|\overline{HB_0}| = \frac{\sin(\widehat{HAB_0})|\overline{AH}|}{\sin(\widehat{HB_0A})} = \frac{\sin(\pi/6-\psi)s}{\sin(\pi/3+\psi+\phi)}$. Hence,

$$\begin{aligned} |\overline{AD}| &= \left(|\overline{AI}| - |\overline{CI}|\right)\frac{\sin(2\pi/3 - \psi - \phi)}{\sin(\pi/3)} && \text{[from (B.16)]} \\ &= \left(\sqrt{(2s)^2 + (vt-s)^2} - \frac{s\sin(\pi/6-\psi)}{\sin(\pi/3+\psi+\phi)}\right)\frac{\sin(2\pi/3 - \psi - \phi)}{\sin(\pi/3)} \\ &= \sqrt{(2s)^2 + (vt-s)^2}\frac{\sin(2\pi/3 - \psi - \phi)}{\sin(\pi/3)} - \frac{s\sin(\pi/6-\psi)}{\sin(\pi/3)} \\ &= 2\sqrt{(2s)^2 + (vt-s)^2}\frac{\sin(2\pi/3 - \psi - \phi)}{\sqrt{3}} - \frac{2s\sin(\pi/6-\psi)}{\sqrt{3}} \\ &= \frac{2\sqrt{(2s)^2 + (vt-s)^2}\sin\left(2\pi/3 - \psi - \phi\right) - 2s\sin(\pi/6-\psi)}{\sqrt{3}} \\ &= \frac{2\sqrt{(2s)^2 + (vt-s)^2}\left(\sin(\frac{2\pi}{3} - \psi)\cos(\phi) - \cos\left(\frac{2\pi}{3} - \psi\right)\sin(\phi)\right)}{\sqrt{3}} \\ &\quad - \frac{2s\sin(\frac{\pi}{6} - \psi)}{\sqrt{3}} \end{aligned}$$

$$
\begin{aligned}
&= \frac{2\sqrt{(2s)^2+(vt-s)^2}\left(\frac{\sin(2\pi/3-\psi)(vt-s)}{\sqrt{(2s)^2+(vt-s)^2}} - \frac{2s\cos(2\pi/3-\psi)}{\sqrt{(2s)^2+(vt-s)^2}}\right)}{\sqrt{3}} \\
&\quad - \frac{2s\sin(\pi/6-\psi)}{\sqrt{3}} \\
&= \frac{2\left(\sin(2\pi/3-\psi)(vt-s) - 2s\cos(2\pi/3-\psi)\right) - 2s\sin(\pi/6-\psi)}{\sqrt{3}} \\
&= \frac{2\left(\sin(2\pi/3-\psi)(vt-s) + 2s\sin(\pi/6-\psi)\right) - 2s\sin(\pi/6-\psi)}{\sqrt{3}} \\
&= \frac{2\sin(2\pi/3-\psi)(vt-s) + 4s\sin(\pi/6-\psi) - 2s\sin(\pi/6-\psi)}{\sqrt{3}} \\
&= \frac{2\sin(2\pi/3-\psi)(vt-s) + 2s\sin(\pi/6-\psi)}{\sqrt{3}} \\
&= \frac{2\cos(\pi/6-\psi)(vt-s) + 2s\sin(\pi/6-\psi)}{\sqrt{3}} \qquad \text{(B.17)}
\end{aligned}
$$

Above it was used $\sin(2\pi/3-\psi) = \cos(\pi/6-\psi)$, $\cos(2\pi/3-\psi) = -\sin(\pi/6-\psi)$, $\sin(2\pi/3-\psi-\phi) = \sin(\pi/3+\psi+\phi)$, $\sin(2\pi/3-\psi-\phi) = \sin(2\pi/3-\psi)\cos(\phi) - \cos(2\pi/3-\psi)\sin(\phi)$, $\sin(\arctan(y/x)) = \frac{y}{\sqrt{x^2+y^2}}$, and $\cos(\arctan(y/x)) = \frac{x}{\sqrt{x^2+y^2}}$.

Therefore, the number of lines parallel to the y_h-axis intersecting $\overline{B_0F}$ for integer x_h values is

$$
\begin{aligned}
n_l^+ &= \left\lfloor \frac{|\overline{B_0F}|}{e} + 1 \right\rfloor = \left\lfloor \frac{|\overline{HF}| - |\overline{HB_0}|}{e} + 1 \right\rfloor = \left\lfloor \frac{|\overline{AC}|}{e} + 1 \right\rfloor \\
&= \left\lfloor \frac{|\overline{AD}|}{d} + 1 \right\rfloor && \text{[from (B.15)]} \\
&= \left\lfloor \frac{2\cos(\pi/6-\psi)(vt-s) + 2s\sin(\pi/6-\psi)}{\sqrt{3}d} + 1 \right\rfloor && \text{[from (B.17)]}
\end{aligned}
$$

- Case $\psi > \frac{\pi}{6}$: Figure B.8 shows this case. The triangles AD_iC_i for any $i \in \{1,2\}$ and ADC are similar. $\overline{AD_i}$ and $\overline{AC_i}$ have distance $i \cdot d$ and $i \cdot e$, respectively. In this example, $d = 2$ and there are three points lying over $\overline{EG}$. Observe that when $\psi > \frac{\pi}{6}$, $\overline{EA}$ is on the left side of the y_h-axis. Also, note that it is being considered now the diagonal $\overline{EG}$, because the y_h-axis does not intersect the diagonal $\overline{HF}$ for these values of ψ. Then, one has to consider $\overline{B_0G}$ to count n_l^+. Additionally,

$|B_0G| = |AC|$, due to the AB_0GC parallelogram properties. As in the previous case, for $i \in \{1, \ldots, n_l^+ - 1\}$, $\triangle AD_iC_i \sim \triangle ADC$, $\widehat{CAD} = \widehat{BAD} - \widehat{BAC} = \psi - \phi$, $\widehat{ADC} = \pi/3$, $\widehat{ACD} = \pi - \widehat{CAD} - \widehat{ADC} = 2\pi/3 - \psi + \phi$, and $\frac{|\overline{B_0G}|}{e} = \frac{|\overline{AC}|}{e} = \frac{|\overline{AD}|}{d}$, by the similarity of these triangles as showed in the previous case. Also, $\widehat{EAB_0} = \widehat{DAE} - \widehat{DAB_0} = \psi + \pi/2 - 2\pi/3 = \psi - \pi/6$, $\widehat{B_0EA} = \widehat{FEA} - \widehat{FEB_0} = \pi/2 - \phi$, $\widehat{EB_0A} = \pi - \widehat{B_0EA} - \widehat{EAB_0} = \pi - (\pi/2 - \phi) - (\psi - \pi/6) = 2\pi/3 + \phi - \psi$. Thus, by the law of sines, $\frac{|\overline{B_0E}|}{\sin(\widehat{EAB_0})} = \frac{|\overline{EA}|}{\sin(\widehat{EB_0A})} \Leftrightarrow |\overline{B_0E}| = \frac{s\sin(\widehat{EAB_0})}{\sin(\widehat{EB_0A})} = \frac{s\sin(\psi - \pi/6)}{\sin(2\pi/3 + \phi - \psi)}$. $EAIG$ and B_0ACG are parallelograms sharing the points G and A, so $|\overline{B_0E}| = |\overline{CI}|$. By following similar steps as before,

$$
\begin{aligned}
n_l^+ &= \left\lfloor \frac{|\overline{B_0G}|}{e} + 1 \right\rfloor = \left\lfloor \frac{|\overline{AC}|}{e} + 1 \right\rfloor = \left\lfloor \frac{|\overline{AD}|}{d} + 1 \right\rfloor \\
&= \left\lfloor \frac{\left(|\overline{AI}| - |\overline{CI}|\right) \frac{\sin(2\pi/3 - \psi + \phi)}{\sin(\pi/3)}}{d} + 1 \right\rfloor \\
&= \left\lfloor \frac{\left(\sqrt{(2s)^2 + (vt - s)^2} - \frac{s\sin(\psi - \pi/6)}{\sin(2\pi/3 - \psi + \phi)}\right) \frac{\sin(2\pi/3 - \psi + \phi)}{\sin(\pi/3)}}{d} + 1 \right\rfloor \\
&= \left\lfloor \frac{\sqrt{(2s)^2 + (vt - s)^2} \frac{\sin(2\pi/3 - \psi + \phi)}{\sin(\pi/3)} - \frac{s\sin(\psi - \pi/6)}{\sin(\pi/3)}}{d} + 1 \right\rfloor \\
&= \left\lfloor \frac{2\sqrt{(2s)^2 + (vt - s)^2}\sin(2\pi/3 - \psi + \phi) - 2s\sin(\psi - \pi/6)}{\sqrt{3}d} + 1 \right\rfloor \\
&= \left\lfloor \frac{2\sqrt{(2s)^2 + (vt - s)^2}(\sin(2\pi/3 - \psi)\cos(\phi) + \cos(2\pi/3 - \psi)\sin(\phi))}{\sqrt{3}d} \right. \\
&\quad \left. - \frac{2s\sin(\psi - \pi/6)}{\sqrt{3}d} + 1 \right\rfloor \\
&= \left\lfloor \frac{2\sin(2\pi/3 - \psi)(vt - s) + 4s\cos(2\pi/3 - \psi) - 2s\sin(\psi - \pi/6)}{\sqrt{3}d} + 1 \right\rfloor \\
&= \left\lfloor \frac{2\sin(2\pi/3 - \psi)(vt - s) + 4s\sin(\psi - \pi/6) - 2s\sin(\psi - \pi/6)}{\sqrt{3}d} + 1 \right\rfloor \\
&= \left\lfloor \frac{2\sin(2\pi/3 - \psi)(vt - s) + 2s\sin(\psi - \pi/6)}{\sqrt{3}d} + 1 \right\rfloor \\
&= \left\lfloor \frac{2\cos(\pi/6 - \psi)(vt - s) + 2s\sin(\psi - \pi/6)}{\sqrt{3}d} + 1 \right\rfloor .
\end{aligned}
$$

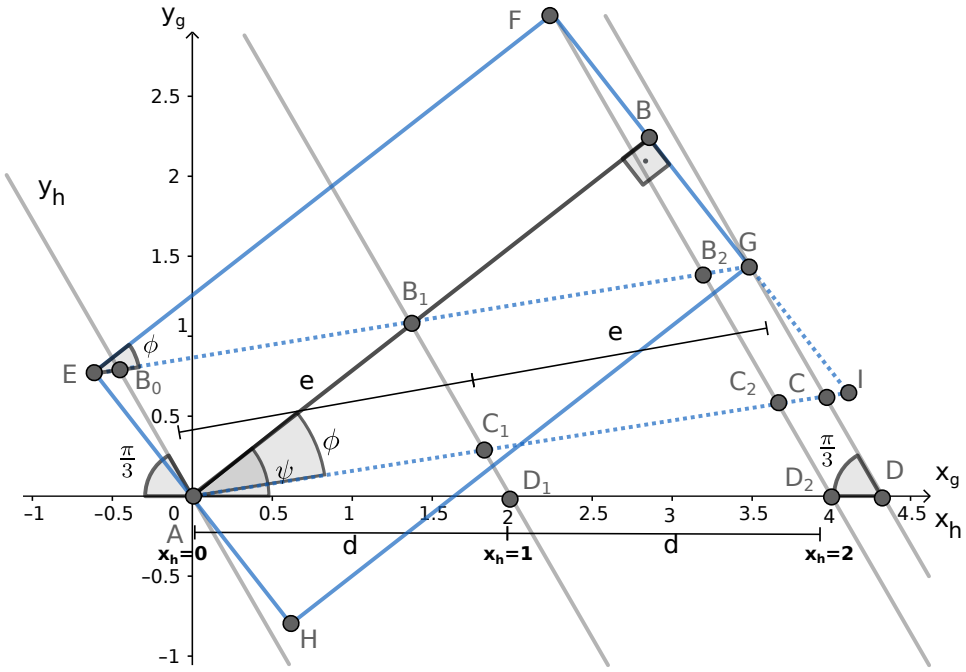


Figure B.8: Counting how many points named B_i lie in the diagonal $\overline{EG}$.

This time $\sin(2\pi/3 - \psi) = \cos(\psi - \pi/6)$ and $\cos(2\pi/3 - \psi) = \sin(\psi - \pi/6)$ was used.

For the final result on (B.13), one can simplify it using the fact that when $\psi \leq \pi/6$, $\sin(|\psi - \pi/6|) = \sin(\pi/6 - \psi)$, otherwise, $\sin(|\psi - \pi/6|) = \sin(\psi - \pi/6)$.

For n_l^-, it is also calculated how many lines parallel to the y_h-axis projected over the x-axis are distant from each other by d on this axis and are inside the rectangle. However, consider only those on the left side of the point A, i.e., commencing from the one whose intersection with the x-axis is at $(-d, 0)$, equivalently, $(-1, 0)$ on the (x_h, y_h) coordinate system. Also, there are two cases here.

- Case $\psi \leq \pi/6$: Figure B.9 shows the $\triangle HIA$ on the left side of the rectangle $EFGH$. The pink line on the left side is an example of one satisfying Lemma 6, while the one on the right side, Lemma 7. The triangles ACE, HIA, BMG and BNF are congruent, because their respective angles are equal – due to parallelism – and $|\overline{EA}| = |\overline{AH}| = |\overline{GB}| = |\overline{FB}| = s$. In this example, except for $\overleftrightarrow{JH}, \overleftrightarrow{EC}, \overleftrightarrow{MG}, \overleftrightarrow{BL}$ and $\overleftrightarrow{FD}$, the lines parallel-to-y_h are distant by d on the projection over the x-axis and can have robots on them. As the robots are over the parallel-to-y_h lines distant by d on the projection over the x-axis, the goal is to know how many parallel lines intersect $\overline{HI}$ (equivalently, how many such lines

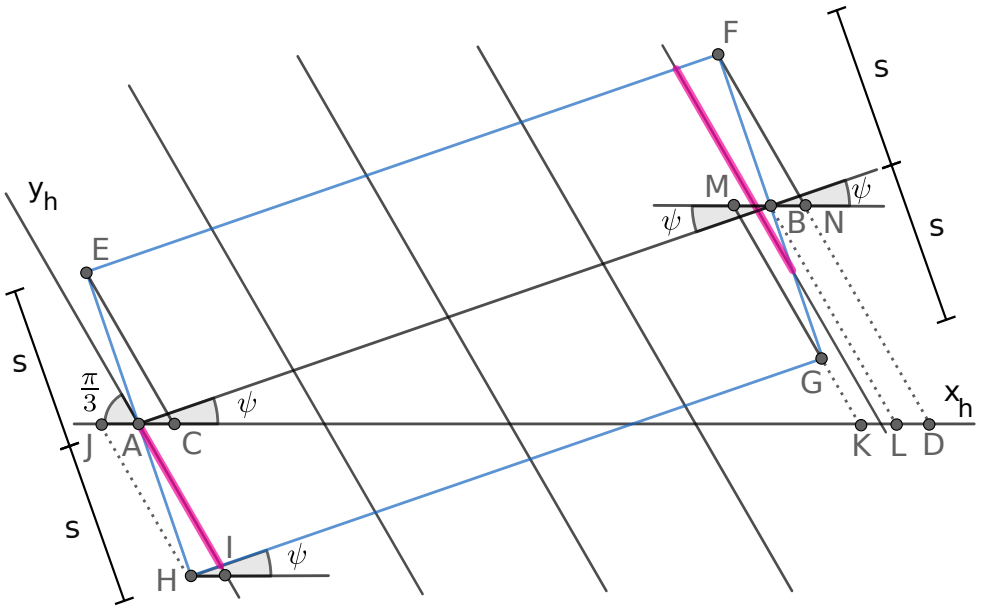


Figure B.9: Triangles ACE, HIA, BMG, BNF and the rectangle $EFGH$ for $\psi \leq \pi/6$.

intersect $\overline{JA}$ due to parallelism), excluding $\overleftrightarrow{AI}$ (because it was already counted on n_l^+). Thus,

$$n_l^- = \left\lfloor \frac{|HI|}{d} \right\rfloor.$$

It is known that $|\overline{AH}| = s, \widehat{H} = \pi/2 + \psi, \widehat{I} = \pi/3$ and $\widehat{A} = \pi - \widehat{I} - \widehat{H} = \pi - \pi/3 - (\pi/2 + \psi) = \pi/6 - \psi$. By the law of sines on the angles opposite to the sides $\overline{AH}$ and $\overline{HI}$, results the following

$$|HI| = \frac{|AH|\sin(\widehat{A})}{\sin(\widehat{I})} = \frac{s\sin\left(\frac{\pi}{6} - \psi\right)}{\sin\left(\frac{\pi}{3}\right)} = \frac{2s\sin\left(\frac{\pi}{6} - \psi\right)}{\sqrt{3}}. \tag{B.18}$$

Thus,

$$n_l^- = \left\lfloor \frac{2s\sin\left(\frac{\pi}{6} - \psi\right)}{\sqrt{3}d} \right\rfloor.$$

- Case $\psi > \pi/6$. Figure B.10 illustrates this case. The side EH has an angle greater than zero with the y_h-axis. The pink line on the left side is an example of

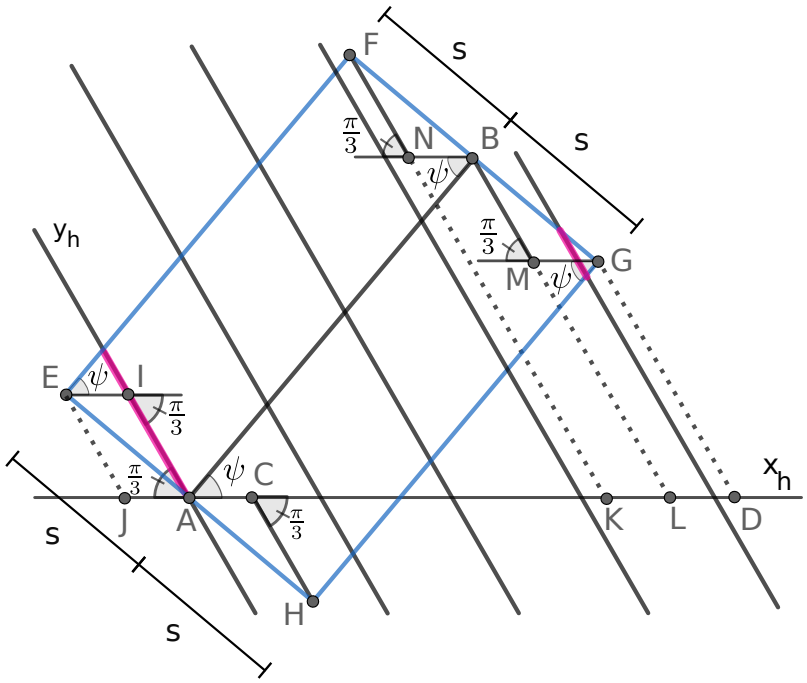


Figure B.10: Triangles AIE, HCA, FNB and BMG and the rectangle $EFGH$ for $\psi > \pi/6$.

one satisfying Lemma 6, while the one on the right side, Lemma 7. The triangles AIE, HCA, FNB and BMG are congruent, because their respective angles are equal – due to parallelism – and $|\overline{EA}| = |\overline{AH}| = |\overline{GB}| = |\overline{FB}| = s$. Except for $\overleftrightarrow{EJ}, \overleftrightarrow{CH}, \overleftrightarrow{FK}, \overleftrightarrow{BL}$ and $\overleftrightarrow{GD}$, the lines parallel-to-y_h are distant by d on the projection over the x-axis and can have robots on them. The reasoning is similar to the previous case, but now using $\triangle EIA$. Then, $|\overline{EA}| = s, \widehat{E} = \pi/2 - \psi$, $\widehat{I} = 2\pi/3$ and $\widehat{A} = \pi - \widehat{I} - \widehat{E} = \pi - 2\pi/3 - (\pi/2 - \psi) = \psi - \pi/6$. Consequently,

$$n_l^- = \left\lfloor \frac{|\overline{EI}|}{d} \right\rfloor = \left\lfloor \frac{2s \sin\left(\psi - \frac{\pi}{6}\right)}{\sqrt{3}d} \right\rfloor .$$

For the final result in (B.14), the absolute value inside the sine function is used to combine both cases. □

The previous lemma has the calculations for the interval of an integer x_h values needed for counting the robots inside the rectangle. The next lemma presents the equation for the number of robots at the rectangular part (N_R) ranging from these integer x_h values. Although the proposition that is now being proved gives the

throughput in terms of θ, this number is first going to be calculated in terms of ψ.

Lemma 5. *For $\psi \in [0, \pi/3)$,*

$$N_R(t,\psi) = \sum_{x_h=-n_l^-}^{n_l^+-1} \left(\lfloor Y_2^R(x_h) \rfloor - \lceil Y_1^R(x_h) \rceil + 1 \right).$$

If for some x_h $\left\lfloor Y_2^R(x_h) \right\rfloor < \left\lceil Y_1^R(x_h) \right\rceil$, the respective summand for this x_h is zero.

Proof. By the previous lemma and knowing that the positions of the robots are integer coordinates over the hexagonal grid coordinate space,

$$N_R(t,\psi) = \sum_{x_h=-n_l^-}^{n_l^+-1} \sum_{y_h=\lceil Y_1^R(x_h) \rceil}^{\lfloor Y_2^R(x_h) \rfloor} 1 = \sum_{x_h=-n_l^-}^{n_l^+-1} \left(\lfloor Y_2^R(x_h) \rfloor - \lceil Y_1^R(x_h) \rceil + 1 \right).$$

since (B.11) and (B.12) give the minimum (Y_1^R) and maximum (Y_2^R) y_h coordinates for a given x_h value such that the robot is inside the rectangle. Note that the last summation can only be used when $\left\lfloor Y_2^R(x_h) \right\rfloor \geq \left\lceil Y_1^R(x_h) \right\rceil$, otherwise a negative number of robots would be accounted. □

In special, for $\psi = \pi/6$, by (B.11) and (B.12),

$$N(t,\pi/6) = \sum_{x_h=0}^{\left\lfloor \frac{2(vt-s)}{\sqrt{3}d} \right\rfloor} \left(\left\lfloor \frac{\sqrt{3}y_2 + dx_h}{2d} \right\rfloor - \left\lceil \frac{\sqrt{3}y_1 + dx_h}{2d} \right\rceil + 1 \right). \tag{B.19}$$

If $\psi \neq \pi/6$, each parallel-to-y_h-axis line intersects two segments of the rectangle $EFGH$. The y_h-components of the two intersections of a rectangle side and such lines are the values of $Y_1^R(x_h)$ and $Y_2^R(x_h)$ for a given x_h. Hence, the set of x_h integer values $\{-n_l^-, \ldots, n_l^+ - 1\}$ will be cut in disjoint subsets based on the max and min outcomes of (B.11) and (B.12). That is, $Y_1^R(x_h)$ and $Y_2^R(x_h)$, respectively; equivalently, which two sides of the rectangle the parallel-to-y_h-axis line corresponding to $(x_h, 0)$ intersects. The following lemmas describe each subset: $\{-n_l^-, \ldots, n_l^-\}$ in Lemma 6; $\{n_l^- + 1, \ldots, K' - 1\}$ in Lemma 9; $\{K', \ldots, n_l^+ - 1\}$ in Lemma 7, for an integer K' defined later.

Lemma 6. *Consider parallel-to-y_h-axis lines inside the rectangle $EFGH$ intersecting the x_h-axis at $(x_h, 0)$, for $x_h \in \mathbb{Z}$. The two following statements are equivalent:*

(I) If $\psi < \pi/6$,

$$Y_1^R(x_h) = \frac{\frac{2y_1}{d} + 2\tan(\psi)x_h}{\sqrt{3} + \tan(\psi)} \text{ and } Y_2^R(x_h) = \frac{-2x_h}{\sqrt{3}\tan(\psi) - 1}, \tag{B.20}$$

and, if $\psi > \pi/6$,

$$Y_1^R(x_h) = \frac{-2x_h}{\sqrt{3}\tan(\psi) - 1} \text{ and } Y_2^R(x_h) = \frac{\frac{2y_2}{d} + 2\tan(\psi)x_h}{\sqrt{3} + \tan(\psi)}. \tag{B.21}$$

(II) $x_h \in \{-n_l^-, \ldots, n_l^-\}$.

Proof. $(I) \Rightarrow (II)$: Let $\psi < \pi/6$. By (B.11) and (B.12), (B.20) is equivalent to

$$\frac{\sin(\psi)x_h - \frac{s}{d}}{\cos\left(\frac{\pi}{6} - \psi\right)} \geq \frac{\frac{vt-s}{d} - \cos(\psi)x_h}{\sin\left(\psi - \frac{\pi}{6}\right)} \text{ and } \frac{\sin(\psi)x_h + \frac{s}{d}}{\cos\left(\frac{\pi}{6} - \psi\right)} \geq \frac{-\cos(\psi)x_h}{\sin\left(\psi - \frac{\pi}{6}\right)}.$$

From the second inequality,

$$\begin{aligned}
&\frac{\sin(\psi)x_h + \frac{s}{d}}{\cos\left(\frac{\pi}{6} - \psi\right)} \geq \frac{-\cos(\psi)x_h}{\sin\left(\psi - \frac{\pi}{6}\right)} \Leftrightarrow \left(\frac{\sin(\psi)}{\cos\left(\frac{\pi}{6} - \psi\right)} + \frac{\cos(\psi)}{\sin\left(\psi - \frac{\pi}{6}\right)}\right) x_h \geq -\frac{s}{d\cos\left(\frac{\pi}{6} - \psi\right)} \\
&\Leftrightarrow \left(\frac{\sin(\psi)\sin\left(\psi - \frac{\pi}{6}\right) + \cos(\psi)\cos\left(\frac{\pi}{6} - \psi\right)}{\cos\left(\frac{\pi}{6} - \psi\right)\sin\left(\psi - \frac{\pi}{6}\right)}\right) x_h \geq -\frac{s\sin\left(\psi - \frac{\pi}{6}\right)}{d\cos\left(\frac{\pi}{6} - \psi\right)\sin\left(\psi - \frac{\pi}{6}\right)} \\
&\quad\Leftrightarrow \left(\frac{\sin(\psi)\sin\left(\psi - \frac{\pi}{6}\right) + \cos(\psi)\cos\left(\psi - \frac{\pi}{6}\right)}{\cos\left(\frac{\pi}{6} - \psi\right)\sin\left(\psi - \frac{\pi}{6}\right)}\right) x_h \geq -\frac{s\sin\left(\psi - \frac{\pi}{6}\right)}{d\cos\left(\frac{\pi}{6} - \psi\right)\sin\left(\psi - \frac{\pi}{6}\right)} \\
&\quad\Leftrightarrow \left(\frac{\cos(\psi - (\psi - \frac{\pi}{6}))}{\cos\left(\frac{\pi}{6} - \psi\right)\sin\left(\psi - \frac{\pi}{6}\right)}\right) x_h \geq -\frac{s\sin\left(\psi - \frac{\pi}{6}\right)}{d\cos\left(\frac{\pi}{6} - \psi\right)\sin\left(\psi - \frac{\pi}{6}\right)} \\
&\quad\Leftrightarrow \left(\frac{\cos(\psi - \psi + \frac{\pi}{6})}{\cos\left(\frac{\pi}{6} - \psi\right)\sin\left(\psi - \frac{\pi}{6}\right)}\right) x_h \geq -\frac{s\sin\left(\psi - \frac{\pi}{6}\right)}{d\cos\left(\frac{\pi}{6} - \psi\right)\sin\left(\psi - \frac{\pi}{6}\right)} \\
&\quad\Leftrightarrow \left(\frac{\cos(\frac{\pi}{6})}{\cos\left(\frac{\pi}{6} - \psi\right)\sin\left(\psi - \frac{\pi}{6}\right)}\right) x_h \geq -\frac{s\sin\left(\psi - \frac{\pi}{6}\right)}{d\cos\left(\frac{\pi}{6} - \psi\right)\sin\left(\psi - \frac{\pi}{6}\right)} \\
&\quad\Leftrightarrow \left(\frac{\frac{\sqrt{3}}{2}}{\cos\left(\frac{\pi}{6} - \psi\right)\sin\left(\psi - \frac{\pi}{6}\right)}\right) x_h \geq -\frac{s\sin\left(\psi - \frac{\pi}{6}\right)}{d\cos\left(\frac{\pi}{6} - \psi\right)\sin\left(\psi - \frac{\pi}{6}\right)} \\
&\quad\Leftrightarrow x_h \leq \frac{-s\sin\left(\psi - \frac{\pi}{6}\right)}{\frac{\sqrt{3}}{2}d} \Leftrightarrow x_h \leq \frac{-2s\sin\left(\psi - \frac{\pi}{6}\right)}{\sqrt{3}d} \Leftrightarrow x_h \leq= \frac{2s\sin\left(\frac{\pi}{6} - \psi\right)}{\sqrt{3}d}.
\end{aligned}$$

The change of inequality sign above is due to $\cos\left(\frac{\pi}{6}-\psi\right)\sin\left(\psi-\frac{\pi}{6}\right)<0$ for $\psi<\pi/6$. As $x_h\in\mathbb{Z}$,

$$x_h \leq \left\lfloor \frac{2s\sin\left(\frac{\pi}{6}-\psi\right)}{\sqrt{3}d} \right\rfloor = n_l^-.$$

The lower value on x_h is obtained by Lemma 4, as to be inside the rectangle $EFGH$ $x_h \geq -n_l^-$. For $\psi>\pi/6$, the same result is obtained by a similar reasoning, but without changing the inequality sign since in this case $\cos\left(\frac{\pi}{6}-\psi\right)\sin\left(\psi-\frac{\pi}{6}\right)>0$.

$(II)\Rightarrow(I)$: From (B.6), (B.7) (i.e., the line equations for $\overleftrightarrow{HG}$, $\overleftrightarrow{EH}$ and $\overleftrightarrow{EF}$), (B.11) and (B.12) (i.e., the definitions of Y_1^R and Y_2^R), if $\psi<\pi/6$,

$$(x_h, Y_1^R(x_h)) \in \overleftrightarrow{HG} \Leftrightarrow Y_1^R(x_h) = \frac{\frac{2y_1}{d}+2\tan(\psi)x_h}{\sqrt{3}+\tan(\psi)},$$

$$(x_h, Y_2^R(x_h)) \in \overleftrightarrow{EH} \Leftrightarrow Y_2^R(x_h) = \frac{-2x_h}{\sqrt{3}\tan(\psi)-1},$$

and, if $\psi>\pi/6$,

$$(x_h, Y_1^R(x_h)) \in \overleftrightarrow{EH} \Leftrightarrow Y_1^R(x_h) = \frac{-2x_h}{\sqrt{3}\tan(\psi)-1},$$

$$(x_h, Y_2^R(x_h)) \in \overleftrightarrow{EF} \Leftrightarrow Y_2^R(x_h) = \frac{\frac{2y_2}{d}+2\tan(\psi)x_h}{\sqrt{3}+\tan(\psi)}.$$

Then, this part is proved by showing that for all $x_h\in\{-n_l^-,\ldots,n_l^-\}$, the line parallel to the y_h-axis intercepting the point $(x_h,0)$ intercepts both sides $\overline{EH}$ and $\overline{HG}$ (and no other), if $\psi<\pi/6$ (Figure B.9), and, if $\psi>\pi/6$, both sides $\overline{EH}$ and $\overline{EF}$ (and no other) (Figure B.10).

- Case $\psi<\pi/6$: Figure B.9 shows the triangles HIA, ACE and BMG inside the rectangle $EFGH$. As the robots are over the parallel lines to the y_h-axis, which are distant by d when projected over the x-axis, the objective is to know how many such parallel lines intersect $\overline{HI}$ (equivalently, how many such lines intersect $\overline{JA}$ due to parallelism) or $\overline{AC}$. For such parallel lines that intersect $\overline{HI}$, Lemma 4 showed that for every $x_h\in\{-n_l^-,\ldots,-1\}$ the line parallel to y_h-axis intersecting $(x_h,0)$ is inside the rectangle. Also, these lines intersect the sides $\overline{EH}$ and $\overline{HG}$, as any line parallel to $\overleftrightarrow{AI}$ which is on its left side intersects the sides $\overline{EH}$ and $\overline{HG}$ if it is inside the rectangle. For the case where such parallel lines intersect $\overline{AC}$, the maximum integer value, M, must be known such that these parallel lines still intersect the sides $\overline{EH}$ and $\overline{HG}$ for any $x_h\in\{0,\ldots,M\}$. Starting from point A (that is, when $x_h=0$),

$$M = \left\lfloor \frac{|\overline{AC}|}{d} \right\rfloor.$$

It is given that $|\overline{AH}| = |\overline{EA}| = s$, $\overleftrightarrow{AI} \parallel \overleftrightarrow{EC}$, $\overleftrightarrow{HI} \parallel \overleftrightarrow{AC}$, and $\overleftrightarrow{AH} \parallel \overleftrightarrow{AE}$ (as E, A and H are collinear), then $\widehat{IHA} = \widehat{CAE}$, $\widehat{AIH} = \widehat{ECA}$, and $\widehat{HAI} = \widehat{AEC}$. Thus, $\triangle HIA \cong \triangle ACE$, then $|\overline{AC}| = |\overline{HI}|$, whose value has been previously calculated in Lemma 4, leading to

$$M = \left\lfloor \frac{2s \sin\left(\frac{\pi}{6} - \psi\right)}{\sqrt{3}d} \right\rfloor = n_l^-.$$

Hence, for any $x_h \in \{0, \ldots, n_l^-\}$, those parallel lines intersect the sides $\overline{EH}$ and $\overline{HG}$.

- Case $\psi > \pi/6$: Figure B.10 illustrates this case. The reasoning is similar to the previous case, but using that $\triangle AIE \cong \triangle HCA$. As the value for $|\overline{EI}|/d$ also has been calculated in Lemma 4 for this figure, then

$$M = \left\lfloor \frac{2s \sin\left(\psi - \frac{\pi}{6}\right)}{\sqrt{3}d} \right\rfloor = n_l^-.$$

Consequently, for any $x_h \in \{-n_l^-, \ldots, n_l^-\}$, the parallel-to-$y_h$-axis line at $(x_h, 0)$ intersects the sides $\overline{EH}$ and $\overline{EF}$ in this case.

□

The next lemma will define the integer K' mentioned before. This number will be compared with the integer x_h coordinate of the point $(n_l^+ - 1, 0)$ intersected by the rightmost parallel-to-y_h-axis line inside the rectangle $EFGH$. Assuming $\theta \neq \pi/6$, if this rightmost line intersects a point on the x_h-axis with an integer coordinate less than K', then no parallel-to-y_h-axis line intersects the rectangle right side $\overline{FG}$. However, if the intersection point coordinate is greater than or equal to K', then at least one parallel line crosses $\overline{FG}$.

Lemma 7. *Consider parallel-to-y_h-axis lines inside the rectangle $EFGH$ intersecting the x_h-axis at $(x_h, 0)$, for $x_h \in \mathbb{Z}$, and $K' = \left\lceil \frac{2(vt-s)\cos(\psi-\pi/6) - 2s\sin(|\psi-\pi/6|)}{\sqrt{3}d} \right\rceil$. Then, the two statements below are equivalent:*

(I) If $\psi < \pi/6$

$$Y_1^R(x_h) = \frac{\frac{2(vt-s)}{d\cos(\psi)} - 2x_h}{\sqrt{3}\tan(\psi) - 1} \text{ and } Y_2^R(x_h) = \frac{\frac{2y_2}{d} + 2\tan(\psi)x_h}{\sqrt{3} + \tan(\psi)}, \tag{B.22}$$

and, if $\psi > \pi/6$

$$Y_1^R(x_h) = \frac{\frac{2y_1}{d} + 2\tan(\psi)x_h}{\sqrt{3} + \tan(\psi)} \text{ and } Y_2^R(x_h) = \frac{\frac{2(vt-s)}{d\cos(\psi)} - 2x_h}{\sqrt{3}\tan(\psi) - 1}. \tag{B.23}$$

(II) $x_h \in \{K', \ldots, n_l^+ - 1\}$.

Proof. $(I) \Rightarrow (II)$: By contrapositive, assume $x_h \notin \{K', \ldots, n_l^+ - 1\}$. By Lemma 4, there is no $x_h > n_l^+ - 1$, so $x_h < K'$. For the case of $\psi < \pi/6$, observe in Figure B.9 the point K on the x_h-axis. This point corresponds to the intersection of $\overleftrightarrow{MG}$ on the x_h-axis, which is the first parallel-to-y_h-axis crossing the rectangle right side $\overline{FG}$. The point D on the x_h-axis is the projection of the point F on this axis. By (B.17), $|\overline{AD}| = \frac{2\cos(\pi/6-\psi)(vt-s)+2s\sin(|\psi-\pi/6|)}{\sqrt{3}}$. Because of the parallelism, $|\overline{MN}| = |\overline{KD}|$. Due to the congruence of triangles ACE, HIA, BMG and BNF and (B.18), $|\overline{BM}| = |\overline{BN}| = |\overline{HI}| = \frac{2s\sin(|\psi-\pi/6|)}{\sqrt{3}}$. Thus, $|\overline{KD}| = |\overline{MN}| = |\overline{BM}| + |\overline{BN}| = \frac{4s\sin(|\psi-\pi/6|)}{\sqrt{3}}$. Since $|\overline{AK}| = |\overline{AD}| - |\overline{KD}| = \frac{2\cos(\pi/6-\psi)(vt-s)-2s\sin(|\psi-\pi/6|)}{\sqrt{3}}$, the point K is located on the (x_h, y_h) coordinate space at $\left(\frac{2\cos(\pi/6-\psi)(vt-s)}{\sqrt{3}d} - \frac{2s\sin(|\psi-\pi/6|)}{\sqrt{3}d}, 0\right)$, as K is on the x-axis and to convert it to (x_h, y_h) coordinate space it is only needed to divide the x-coordinate by d. On the x_h-axis, the nearest point on the right of K with integer x_h is $(\lceil K \rceil, 0) = (K', 0)$. As it is assumed $x_h < K'$, no parallel-to-y_h-axis crossing a integer $(x_h, 0)$ point inside the rectangle intersects $\overline{FG}$. Thus, no such parallel line has $Y_1^R(x_h) = \frac{\frac{2(vt-s)}{d\cos(\psi)}-2x_h}{\sqrt{3}\tan(\psi)-1}$, which is the y_h-coordinate of the intersection of this line with $\overleftrightarrow{FG}$.

In the case of $\psi > \pi/6$, using a similar argument as in Figure B.10 leads to the desired result, but here $|\overline{NB}| + |\overline{MG}| = |\overline{KD}|$ and the congruence is between the triangles AIE, HCA, FNB and BMG. As it is assumed $x_h < K'$, no parallel-to-y_h-axis intersecting a integer point $(x_h, 0)$ inside the rectangle crosses $\overline{FG}$, so for such line $Y_2^R(x_h) \neq \frac{\frac{2(vt-s)}{d\cos(\psi)}-2x_h}{\sqrt{3}\tan(\psi)-1}$.

$(II) \Rightarrow (I)$: If $x_h \in \{K', \ldots, n_l^+ - 1\}$ then the lines parallel-to-y_h-axis inside the rectangle intersecting the x_h-axis at $(x_h, 0)$ are on the right of point K or intersecting it. Hence, these lines intersect $\overline{EF}$ and $\overline{FG}$, if $\psi < \pi/6$. By applying (B.6), (B.7) (for the line equations for $\overleftrightarrow{EF}$ and $\overleftrightarrow{FG}$), (B.11) and (B.12) (for the definitions of Y_1^R and Y_2^R), (B.22) is obtained. A similar argument is used in the case of $\psi > \pi/6$, but for $\overline{FG}$ and $\overline{HG}$ intersections, yielding (B.23). □

The lemma below characterises when a parallel-to-y_h-axis line touches only the sides EH and FG of the rectangle. Intuitively, if this happens, a rectangle with a

small width is obtained. Thus, on rectangles with a large width, no such lines are crossing the sides EH and FG, for $\psi \neq \pi/6$. This lemma will be used on the Lemma 9, for completing the disjoint subsets based on the possible max and min outcomes of Y_1^R and Y_2^R.

Lemma 8. *If $vt - s > 2s\tan(|\psi - \frac{\pi}{6}|)$, then there is not a $x_h \in \{-n_l^-, \ldots, n_l^+ - 1\}$ such that,*

$$Y_1^R(x_h) = \frac{\frac{2(vt-s)}{d\cos(\psi)} - 2x_h}{\sqrt{3}\tan(\psi) - 1} \text{ and } Y_2^R(x_h) = \frac{-2x_h}{\sqrt{3}\tan(\psi) - 1}, \text{ if } \psi < \pi/6;$$

$$Y_1^R(x_h) = \frac{-2x_h}{\sqrt{3}\tan(\psi) - 1} \text{ and } Y_2^R(x_h) = \frac{\frac{2(vt-s)}{d\cos(\psi)} - 2x_h}{\sqrt{3}\tan(\psi) - 1}, \text{ if } \psi > \pi/6.$$

Proof. This proof is by contrapositive. Assume $\psi < \pi/6$. By (B.11) and (B.12), there is an x_h such that

$$\frac{\frac{y_1}{d} + \tan(\psi)x_h}{\frac{\sqrt{3}+\tan(\psi)}{2}} \leq \frac{\frac{vt-s}{d\cos(\psi)} - x_h}{\frac{\sqrt{3}\tan(\psi)-1}{2}} \text{ and } \frac{\frac{y_2}{d} + \tan(\psi)x_h}{\frac{\sqrt{3}+\tan(\psi)}{2}} \geq \frac{-x_h}{\frac{\sqrt{3}\tan(\psi)-1}{2}}.$$

Since $\frac{\sqrt{3}\tan(\psi)-1}{2} < 0$, the signs of inequalities change, then the following implication is obtained

$$\begin{aligned}
&\frac{\frac{y_1}{d}\frac{\sqrt{3}\tan(\psi)-1}{2}}{\frac{\sqrt{3}+\tan(\psi)}{2}} - \frac{vt-s}{d\cos(\psi)} \geq -x_h - \frac{\tan(\psi)x_h\frac{\sqrt{3}\tan(\psi)-1}{2}}{\frac{\sqrt{3}+\tan(\psi)}{2}} \text{ and} \\
&\frac{\frac{y_2}{d}\frac{\sqrt{3}\tan(\psi)-1}{2}}{\frac{\sqrt{3}+\tan(\psi)}{2}} \leq -x_h - \frac{\tan(\psi)x_h\frac{\sqrt{3}\tan(\psi)-1}{2}}{\frac{\sqrt{3}+\tan(\psi)}{2}} \\
\Rightarrow &\frac{\frac{y_2}{d}\frac{\sqrt{3}\tan(\psi)-1}{2}}{\frac{\sqrt{3}+\tan(\psi)}{2}} \leq \frac{\frac{y_1}{d}\frac{\sqrt{3}\tan(\psi)-1}{2}}{\frac{\sqrt{3}+\tan(\psi)}{2}} - \frac{vt-s}{d\cos(\psi)},
\end{aligned}$$

by the transitivity of $\leq$ under the real numbers. Also, the following equivalences is obtained

$$
\begin{aligned}
&\frac{\frac{y_2}{d}\frac{\sqrt{3}\tan(\psi)-1}{2}}{\frac{\sqrt{3}+\tan(\psi)}{2}} \leq \frac{\frac{y_1}{d}\frac{\sqrt{3}\tan(\psi)-1}{2}}{\frac{\sqrt{3}+\tan(\psi)}{2}} - \frac{vt-s}{d\cos(\psi)} \\
\Leftrightarrow\ &\frac{vt-s}{d\cos(\psi)} \leq \frac{\frac{y_1}{d}\frac{\sqrt{3}\tan(\psi)-1}{2}}{\frac{\sqrt{3}+\tan(\psi)}{2}} - \frac{\frac{y_2}{d}\frac{\sqrt{3}\tan(\psi)-1}{2}}{\frac{\sqrt{3}+\tan(\psi)}{2}} \\
\Leftrightarrow\ &\frac{vt-s}{d\cos(\psi)} \leq \frac{y_1-y_2}{d}\frac{\sqrt{3}\tan(\psi)-1}{\sqrt{3}+\tan(\psi)} \\
\Leftrightarrow\ &\frac{vt-s}{d\cos(\psi)} \leq -\frac{2s}{d\cos(\psi)}\frac{\sqrt{3}\tan(\psi)-1}{\sqrt{3}+\tan(\psi)} \qquad \text{[By (B.4)]} \\
\Leftrightarrow\ &\frac{vt-s}{2s} \leq \frac{1-\sqrt{3}\tan(\psi)}{\sqrt{3}+\tan(\psi)} \Leftrightarrow \frac{vt-s}{2s} \leq \frac{1}{\tan(\pi/3+\psi)} \\
\Leftrightarrow\ &\frac{vt-s}{2s} \leq \cot(\pi/3+\psi) \Leftrightarrow \frac{vt-s}{2s} \leq -\tan(\psi+5\pi/6) \\
\Leftrightarrow\ &vt-s \leq 2s\tan(\pi/6-\psi).
\end{aligned}
$$

It is used above the equalities $\tan(a+b) = \frac{\tan(a)+\tan(b)}{1-\tan(a)\tan(b)}$, $\cot(a) = -\tan(a+\pi/2)$ and $-\tan(\pi-a) = \tan(a)$ for any real a and b.

For the case $\psi > \pi/6$, using similar arguments the same result is obtained, but the signs of inequalities are not changed due to $\frac{\sqrt{3}\tan(\psi)-1}{2} > 0$ in this case. The conclusion is reached after combining the two cases using absolute values inside the tangent. □

The next lemma completes the properties of $N_R(t,\psi)$ that are useful for calculating its limit when t tends to infinity.

Lemma 9. *Let* $K' = \left\lceil \frac{2(vt-s)\cos(\psi-\pi/6)-2s\sin(|\psi-\pi/6|)}{\sqrt{3}d} \right\rceil$. *If* $vt-s > 2s\tan(|\psi-\pi/6|)$, *then* $x_h \in \{n_l^- + 1, \ldots, K'-1\}$ *if and only if*

$$
Y_1^R(x_h) = \frac{\frac{y_1}{d} + \tan(\psi)x_h}{\frac{\sqrt{3}+\tan(\psi)}{2}} \text{ and } Y_2^R(x_h) = \frac{\frac{y_2}{d} + \tan(\psi)x_h}{\frac{\sqrt{3}+\tan(\psi)}{2}}.
$$

Proof. Excluding the case when $\psi = \pi/6$, (B.11) and (B.12) give four combinations of possible outcomes for the values of $Y_1^R(x_h)$ and $Y_2^R(x_h)$ based on the results of min and max. When $vt-s > 2s\tan(|\psi-\pi/6|)$, by Lemma 8, there is not the case when they are on the sides EH and FG. For the given values of x_h on the hypothesis, neither Lemma 6 nor Lemma 7 applies, excluding other two combinations of results for $Y_1^R(x_h)$ and $Y_2^R(x_h)$. Finally, Lemma 4 shows that every parallel-to-y_h-axis line crosses the x_h-axis at $(x_h, 0)$ for $x_h \in \{-n_l^-, \ldots, n_l^+ - 1\}$, so the remaining combination yields the desired equivalence. □

Now the calculation of $N_S(t,\theta)$ is presented. Here θ is being used instead of $\psi = \pi/3 - \theta$ for easiness of presentation. Denote (l_x, l_y) the position of the last robot inside a rectangle of width $vt - s$ and height $2s$ whose left side is at (x_0, y_0). Here *last* means the robot with highest x coordinate value. However, if two robots have the same x coordinate value, take the robot whose y coordinate is nearer to y_0. Let Z be the set of robot positions inside the rectangle above for $vt - s > 0$.

Lemma 10. *Let* $c_x = x_0 + vt - s$, *and*

$$(l_x, l_y) = \begin{cases} \operatorname{argmin}_{(x,y)\in Z} |vt - s + x_0 - x| + |y_0 - y| & \text{if } t > \frac{s}{v}, \\ (x_0, y_0) & \text{otherwise.} \end{cases}$$

Then,

$$N_S(t,\theta) = \sum_{x_h=B}^{U} \left(\lfloor Y_2^S(x_h) \rfloor - \lceil Y_1^S(x_h) \rceil + 1 \right),$$

for $\lfloor Y_2^S(x_h) \rfloor \geq \lceil Y_1^S(x_h) \rceil$ *(if for some* x_h $\lfloor Y_2^S(x_h) \rfloor < \lceil Y_1^S(x_h) \rceil$, *assume the respective summand for this* x_h *being zero),*

$$B = \begin{cases} \left\lceil \dfrac{2(\sin(\pi/3-\theta)(c_x - l_x) + \cos(\pi/3-\theta)(y_0 - l_y - s))}{\sqrt{3}d} \right\rceil, & \text{if } t > \dfrac{s}{v}, \\ \left\lceil -\dfrac{2\sqrt{2svt - (vt)^2}}{\sqrt{3}d} \sin\left(\theta + \dfrac{\pi}{6}\right) \right\rceil, & \text{otherwise,} \end{cases} \tag{B.24}$$

if $t > \frac{s}{v}$ *or* $\arctan\left(\frac{\frac{s}{2} - \sin(\theta)(vt-s)}{\frac{\sqrt{3}s}{2} + \cos(\theta)(vt-s)}\right) \leq \frac{\pi}{2} - \theta$,

$$U = \left\lfloor \frac{2(\sin(\pi/3-\theta)(c_x - l_x) + \cos(\pi/3-\theta)(y_0 - l_y) + s)}{\sqrt{3}d} \right\rfloor, \tag{B.25}$$

otherwise,

$$U = \left\lfloor \frac{2\sqrt{2svt - (vt)^2}}{\sqrt{3}d} \cos\left(\theta - \frac{\pi}{3}\right) \right\rfloor. \tag{B.26}$$

Also,

$$Y_1^S(x_h) = \frac{dx_h - C_{-\theta,x} + \sqrt{3}C_{-\theta,y} - \sqrt{\Delta(x_h)}}{2d},$$

$$Y_2^S(x_h) = \begin{cases} \min(L(x_h), C_2(x_h)) - 1, & \text{if } \min(L(x_h), C_2(x_h)) = \lfloor L(x_h) \rfloor \\ & \text{and } t > \dfrac{s}{v}, \\ \min(L(x_h), C_2(x_h)), & \text{otherwise,} \end{cases}$$

for

$$C_{-\theta} = \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{bmatrix} \begin{bmatrix} c_x - l_x \\ y_0 - l_y \end{bmatrix},$$

$$\Delta(x_h) = 4s^2 - \left(\sqrt{3}(dx_h - C_{-\theta,x}) - C_{-\theta,y}\right)^2,$$

$$C_2(x_h) = \frac{dx_h - C_{-\theta,x} + \sqrt{3}C_{-\theta,y} + \sqrt{\Delta(x_h)}}{2d},$$

and

$$L(x_h) = \begin{cases} \dfrac{\sin\left(\frac{\pi}{2} - \theta\right)(dx_h - C_{-\theta,x}) + \cos\left(\frac{\pi}{2} - \theta\right) C_{-\theta,y}}{d\sin\left(\frac{5\pi}{6} - \theta\right)}, & \text{if } t > \dfrac{s}{v}, \\ \dfrac{\sin\left(\frac{\pi}{2} - \theta\right) x_h}{\sin\left(\frac{5\pi}{6} - \theta\right)}, & \text{otherwise.} \end{cases}$$

Proof. Assume $t > \frac{s}{v}$, as shown in Figure B.3 (III). The robots are located in the usual Euclidean space. However, instead of it, this proof uses a similar coordinate system transformation for positioning the robots in a hexagonal grid with integer coordinates, similarly to how it was performed in the rectangular part. As in the previous lemmas, call this coordinate system space coordinates (x_h, y_h). However, here a (x_h, y_h) coordinate system is being used with a different origin and inclination.

In order to do so, first redefine a (x_g, y_g) coordinate space, that is, perform rotation by $-\theta$ on the usual Euclidean space about (l_x, l_y). The origin of the (x_g, y_g) coordinate system is at (l_x, l_y). The transformation for (x_g, y_g) coordinate system used here is similar to the depicted in the Figure B.4, but here $-\theta$ and (l_x, l_y) are being used instead of $-\psi$ and (x_0, y_0), i.e.,

$$\begin{bmatrix} x_g \\ y_g \end{bmatrix} = \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{bmatrix} \begin{bmatrix} x - l_x \\ y - l_y \end{bmatrix}.$$

As the coordinate space (x_g, y_g) is already translated to the point (l_x, l_y), the transformation from the new (x_h, y_h) to the new (x_g, y_g) is the same as in (B.3), repeated below for convenience:

$$\begin{bmatrix} x_g \\ y_g \end{bmatrix} = \begin{bmatrix} d & -\frac{d}{2} \\ 0 & \frac{\sqrt{3}d}{2} \end{bmatrix} \begin{bmatrix} x_h \\ y_h \end{bmatrix}. \tag{B.27}$$

Despite these differences, the notation (x_g, y_g) and (x_h, y_h) will be kept as before for a clean presentation.

Figure B.11 shows how the semicircle with centre at $C = (c_x, c_y) = (x_0 + vt - s, y_0)$ will be after the rotation by $-\theta$ about (l_x, l_y), that is,

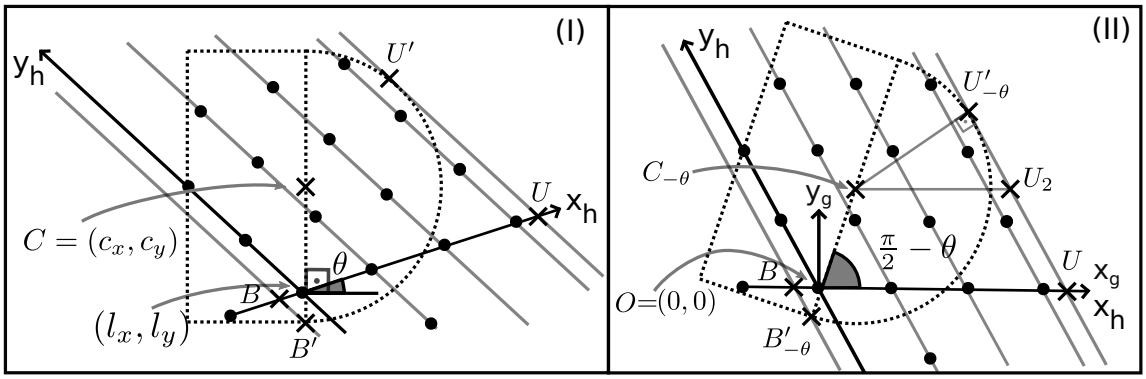


Figure B.11: Semicircle for counting the robots after the rotation on the coordinate space.

$$C_{-\theta} = \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{bmatrix} \begin{bmatrix} c_x - l_x \\ c_y - l_y \end{bmatrix}. \tag{B.28}$$

In the space (I) in Figure B.11, the robots are in the standard coordinate system and the semicircle with centre at $C = (c_x, c_y)$ has the lowest point at $B^{'}$. $\overleftrightarrow{CB'}$ has angle $\frac{\pi}{2}$ with the usual x-axis, however the x_h-axis here has angle θ with it. In (II) it is rotated by $-\theta$ with (l_x, l_y) as centre of rotation. After this rotation, $B^{'}$, $U^{'}$ and C become $B^{'}_{-\theta}$, $U^{'}_{-\theta}$ and $C_{-\theta}$, respectively, and $\overleftrightarrow{C_{-\theta}B^{'}_{-\theta}}$ has angle $\frac{\pi}{2} - \theta$ in relation to the x_g-axis and x_h-axis, which are now coincident lines despite their scale being different. B and U are the minimum and maximum values of the x_h-axis coordinate for a line parallel to the y_h-axis on the hexagonal grid coordinate system. Hereafter the subscript $-\theta$ is on every point presented on the usual Euclidean space to denote the corresponding point on the (x_g, y_g) coordinate space.

The upper and lower values, U and B, of x_h lying on the semicircle are computed first. For getting the U value on the x_h-axis, draw a line parallel to the y_h-axis on the rightmost semicircle boundary at the point $U^{'}$ in order to reach the x_h-axis (Figure B.11 (I)). The corresponding point on the (x_g, y_g) space is denoted by $U^{'}_{-\theta}$ (Figure B.11 (II)). $U^{'}_{-\theta}$ is computed, then its x_h-value on the hexagonal grid coordinate system. $\triangle U^{'}_{-\theta} C_{-\theta} U_2$ in Figure B.11 (II) has $|U^{'}_{-\theta} C_{-\theta}| = s$ and $\widehat{U^{'}_{-\theta} C_{-\theta} U_2} = \pi - \widehat{C_{-\theta} U^{'}_{-\theta} U_2} - \widehat{U^{'}_{-\theta} U_2 C_{-\theta}} = \pi - \pi/2 - \pi/3 = \pi/6$. Hence,

$$
\begin{aligned}
U^{'}_{-\theta} &= C_{-\theta} + s(\cos(\pi/6), \sin(\pi/6)) = C_{-\theta} + \left(\frac{\sqrt{3}s}{2}, \frac{s}{2}\right) \\
&= \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{bmatrix} \begin{bmatrix} c_x - l_x \\ c_y - l_y \end{bmatrix} + \begin{bmatrix} \frac{\sqrt{3}s}{2} \\ \frac{s}{2} \end{bmatrix} \\
&= \Big(\cos(-\theta)(c_x - l_x) - \sin(-\theta)(c_y - l_y) + \frac{\sqrt{3}s}{2}, \\
&\qquad \sin(-\theta)(c_x - l_x) + \cos(-\theta)(c_y - l_y) + \frac{s}{2} \Big) \\
&= \Big(\cos(\theta)(c_x - l_x) + \sin(\theta)(c_y - l_y) + \frac{\sqrt{3}s}{2}, \\
&\qquad \cos(\theta)(c_y - l_y) - \sin(\theta)(c_x - l_x) + \frac{s}{2} \Big).
\end{aligned}
$$

The inverse transformation from (B.27) is

$$
\begin{bmatrix} x_h \\ y_h \end{bmatrix} = \begin{bmatrix} \frac{1}{d} & \frac{1}{\sqrt{3}d} \\ 0 & \frac{2}{\sqrt{3}d} \end{bmatrix} \begin{bmatrix} x_g \\ y_g \end{bmatrix}. \tag{B.29}
$$

Applying the transformation of (B.29) to the point $U^{'}_{-\theta}$ its x_h-axis coordinate is

$$
\begin{aligned}
U &= \frac{1}{d}\left(\cos(\theta)(c_x - l_x) + \sin(\theta)(c_y - l_y) + \frac{\sqrt{3}s}{2}\right) + \\
&\quad \frac{1}{\sqrt{3}d}\left(\cos(\theta)(c_y - l_y) - \sin(\theta)(c_x - l_x) + \frac{s}{2}\right) \\
&= \frac{1}{d}\left(\cos(\theta)(c_x - l_x) + \sin(\theta)(c_y - l_y)\right) + \frac{\sqrt{3}s}{2d} + \\
&\quad \frac{1}{\sqrt{3}d}\left(\cos(\theta)(c_y - l_y) - \sin(\theta)(c_x - l_x)\right) + \frac{s}{2\sqrt{3}d} \\
&= \frac{1}{d}\left(\cos(\theta)(c_x - l_x) + \sin(\theta)(c_y - l_y)\right) + \frac{3s + s}{2\sqrt{3}d} + \\
&\quad \frac{1}{\sqrt{3}d}\left(\cos(\theta)(c_y - l_y) - \sin(\theta)(c_x - l_x)\right) \\
&= \frac{\cos(\theta)(c_x - l_x) + \sin(\theta)(c_y - l_y)}{d} + \frac{\cos(\theta)(c_y - l_y) - \sin(\theta)(c_x - l_x)}{\sqrt{3}d} + \frac{2s}{\sqrt{3}d}
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{1}{\sqrt{3}d}(\sqrt{3}\cos(\theta)(c_x - l_x) + \sqrt{3}\sin(\theta)(c_y - l_y) + \cos(\theta)(c_y - l_y) \\
&\quad - \sin(\theta)(c_x - l_x)) + \frac{2s}{\sqrt{3}d} \\
&= \frac{2}{\sqrt{3}d}\left(\frac{\sqrt{3}}{2}\cos(\theta)(c_x - l_x) + \frac{\sqrt{3}}{2}\sin(\theta)(c_y - l_y) + \frac{1}{2}\cos(\theta)(c_y - l_y)\right. \\
&\quad \left. - \frac{1}{2}\sin(\theta)(c_x - l_x)\right) + \frac{2s}{\sqrt{3}d} \\
&= \frac{2}{\sqrt{3}d}\left(\left(\frac{\sqrt{3}}{2}\cos(\theta) - \frac{1}{2}\sin(\theta)\right)(c_x - l_x) + \left(\frac{\sqrt{3}}{2}\sin(\theta) + \frac{1}{2}\cos(\theta)\right)(c_y - l_y)\right) \\
&\quad + \frac{2s}{\sqrt{3}d} \\
&= \frac{2}{\sqrt{3}d}((\sin(\pi/3)\cos(\theta) - \cos(\pi/3)\sin(\theta))(c_x - l_x) + (\sin(\pi/3)\sin(\theta) \\
&\quad + \cos(\pi/3)\cos(\theta))(c_y - l_y)) + \frac{2s}{\sqrt{3}d} \\
&= \frac{2}{\sqrt{3}d}(\sin(\pi/3 - \theta)(c_x - l_x) + \cos(\pi/3 - \theta)(c_y - l_y)) + \frac{2s}{\sqrt{3}d} \\
&= \frac{2(\sin(\pi/3 - \theta)(c_x - l_x) + \cos(\pi/3 - \theta)(c_y - l_y) + s)}{\sqrt{3}d} \qquad \text{(B.30)}
\end{aligned}
$$

As the integer coordinate less or equal to this value is needed, the floor function is applied to yield the desired result in (B.25).

For getting the B value on the x_h-axis, draw a line parallel to the y_h-axis on the lower semicircle corner at the point $B^{'}$ in order to reach the x_h-axis (Figure B.11 (I)). A calculation similar to the previous paragraph is performed but using $B^{'}_{-\theta}$ (Figure B.11 (II)). It is obtained $\widehat{C_{-\theta}OU} = \pi/2 - \theta$ (as this is the same angle of $\overleftrightarrow{CB'}$ with x_h-axis in Figure B.11 (I) which coincides with x_g-axis in the Figure B.11 (II)). Then, as the vector $C_{-\theta}B^{'}_{-\theta}$ is pointed downwards, it has negative angle with the x_g-axis, that is, $-\widehat{B_{-\theta}OU} = -(\pi - \widehat{C_{-\theta}OU}) = -(\pi - (\pi/2 - \theta)) = -\pi/2 - \theta$ with x_g-axis. Also, $\left|\overline{C_{-\theta}B^{'}_{-\theta}}\right| = s$. Consequently,

$$
\overrightarrow{C_{-\theta}B^{'}_{-\theta}} = B^{'}_{-\theta} - C_{-\theta} = s(\cos(-\pi/2 - \theta), \sin(-\pi/2 - \theta)) \Leftrightarrow
$$

$$
\begin{aligned}
B^{'}_{-\theta} &= C_{-\theta} + s(\cos(-\pi/2-\theta), \sin(-\pi/2-\theta)) \\
&= C_{-\theta} + s(\sin(-\theta), -\cos(-\theta)) \\
&= C_{-\theta} - s(\sin(\theta), \cos(\theta)) \\
&= \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{bmatrix} \begin{bmatrix} c_x - l_x \\ c_y - l_y \end{bmatrix} - \begin{bmatrix} s\sin(\theta) \\ s\cos(\theta) \end{bmatrix} \\
&= (\cos(\theta)(c_x - l_x) + \sin(\theta)(c_y - l_y) - s\sin(\theta), \\
&\quad \cos(\theta)(c_y - l_y) - \sin(\theta)(c_x - l_x) - s\cos(\theta)) \\
&= (\cos(\theta)(c_x - l_x) + \sin(\theta)(c_y - l_y - s), \\
&\quad \cos(\theta)(c_y - l_y - s) - \sin(\theta)(c_x - l_x)).
\end{aligned}
$$

Using (B.29) on $B^{'}_{-\theta}$,

$$
\begin{aligned}
B &= \frac{1}{d}\left(\cos(\theta)(c_x - l_x) + \sin(\theta)(c_y - l_y - s)\right) + \\
&\quad \frac{1}{\sqrt{3}d}\left(\cos(\theta)(c_y - l_y - s) - \sin(\theta)(c_x - l_x)\right) \\
&= \frac{\cos(\theta)(c_x - l_x) + \sin(\theta)(c_y - l_y - s)}{d} + \frac{\cos(\theta)(c_y - l_y - s) - \sin(\theta)(c_x - l_x)}{\sqrt{3}d} \\
&= \frac{1}{\sqrt{3}d}\Big(\sqrt{3}\cos(\theta)(c_x - l_x) + \sqrt{3}\sin(\theta)(c_y - l_y - s) + \cos(\theta)(c_y - l_y - s) - \\
&\quad \sin(\theta)(c_x - l_x)\Big) \\
&= \frac{2}{\sqrt{3}d}\left(\frac{\sqrt{3}}{2}\cos(\theta)(c_x - l_x) + \frac{\sqrt{3}}{2}\sin(\theta)(c_y - l_y - s) + \frac{1}{2}\cos(\theta)(c_y - l_y - s) - \right. \\
&\quad \left. \frac{1}{2}\sin(\theta)(c_x - l_x)\right) \\
&= \frac{2}{\sqrt{3}d}\left(\frac{\sqrt{3}}{2}\cos(\theta)(c_x - l_x) - \frac{1}{2}\sin(\theta)(c_x - l_x) + \frac{\sqrt{3}}{2}\sin(\theta)(c_y - l_y - s) + \right. \\
&\quad \left. \frac{1}{2}\cos(\theta)(c_y - l_y - s)\right) \\
&= \frac{2}{\sqrt{3}d}((\sin(\pi/3)\cos(\theta) - \cos(\pi/3)\sin(\theta))(c_x - l_x) + (\sin(\pi/3)\sin(\theta) + \\
&\quad \cos(\pi/3)\cos(\theta))(c_y - l_y - s)) \\
&= \frac{2(\sin(\pi/3 - \theta)(c_x - l_x) + \cos(\pi/3 - \theta)(c_y - l_y - s))}{\sqrt{3}d}.
\end{aligned}
$$

Then, the ceiling function is applied on this value to get an integer coordinate greater or equal to it in order to obtain (B.24) for $t > \frac{s}{v}$.

On the hexagonal grid coordinate system, for each x_h from B to U, it is needed to find the minimum and maximum y_h – namely $Y_1^S(x_h)$ and $Y_2^S(x_h)$, respectively – of a line parallel to y_h-axis intercepting the x_h-axis and lying on the semicircle. Depending on the angle of $\overleftrightarrow{C_{-\theta}B'_{-\theta}}$ with the x_h-axis, the minimum and maximum y_h can be either on the semicircle arc or $\overleftrightarrow{C_{-\theta}B'_{-\theta}}$. Due to $\theta \in [0, \pi/3)$, the angle of $\overleftrightarrow{C_{-\theta}B'_{-\theta}}$ is in $(\frac{\pi}{6}, \frac{\pi}{2}]$. Thus, the minimum y_h value is at the semicircle arc, otherwise the minimum angle of $\overleftrightarrow{C_{-\theta}B'_{-\theta}}$ would be $2\pi/3$, which is the y_h-axis angle with the x_h-axis. However, the maximum y_h value could be either on $\overleftrightarrow{C_{-\theta}B'_{-\theta}}$ or on the circle, thus take the lowest, since the y_h value on the boundary of the semicircle is wanted.

Let $C_1(x_h)$ and $C_2(x_h)$ be functions that respectively return the lowest and the highest y_h value at the circle centred at $C_{-\theta}$ and radius s for a x_h coordinate value of a parallel-to-y_h-axis line assuming it intersects the circle. Then, a point (x_g, y_g) on the Euclidean space is on that circle if

$$(x_g - C_{-\theta,x})^2 + (y_g - C_{-\theta,y})^2 = s^2 \Leftrightarrow$$

$$\left(dx_h - \frac{dy_h}{2} - C_{-\theta,x}\right)^2 + \left(\frac{\sqrt{3}dy_h}{2} - C_{-\theta,y}\right)^2 = s^2,$$

by (B.27).

Isolating y_h and solving the two degree polynomial it is obtained

$$y_{h_1} = C_1(x_h) = \frac{dx_h - C_{-\theta,x} + \sqrt{3}C_{-\theta,y} - \sqrt{\Delta(x_h)}}{2d} \text{ and} \tag{B.31}$$

$$y_{h_2} = C_2(x_h) = \frac{dx_h - C_{-\theta,x} + \sqrt{3}C_{-\theta,y} + \sqrt{\Delta(x_h)}}{2d}, \tag{B.32}$$

for $0 \leq \Delta(x_h) = 4s^2 - \left(\sqrt{3}(dx_h - C_{-\theta,x}) - C_{-\theta,y}\right)^2$. $\Delta(x_h)$ cannot be negative, otherwise the lines would not intersect this circle, contradicting the assumption.

Denote $L(x_h)$ a function that returns the y_h component of the line $\overleftrightarrow{C_{-\theta}B_{-\theta}}$ for a given x_h. The $\overleftrightarrow{C_{-\theta}B_{-\theta}}$ equation for a point in the space (x_g, y_g) is

$$
\begin{aligned}
&\tan\left(\frac{\pi}{2}-\theta\right)=\frac{y_g-C_{-\theta,y}}{x_g-C_{-\theta,x}}\\
&\Leftrightarrow \frac{\sin\left(\frac{\pi}{2}-\theta\right)}{\cos\left(\frac{\pi}{2}-\theta\right)}=\frac{y_g-C_{-\theta,y}}{x_g-C_{-\theta,x}}\\
&\Leftrightarrow \sin\left(\frac{\pi}{2}-\theta\right)\left(dx_h-\frac{dy_h}{2}-C_{-\theta,x}\right)=\cos\left(\frac{\pi}{2}-\theta\right)\left(\frac{\sqrt{3}dy_h}{2}-C_{-\theta,y}\right)\\
&\Leftrightarrow \sin\left(\frac{\pi}{2}-\theta\right)(dx_h-C_{-\theta,x})-\sin\left(\frac{\pi}{2}-\theta\right)\frac{dy_h}{2}=\cos\left(\frac{\pi}{2}-\theta\right)\frac{\sqrt{3}dy_h}{2}\\
&\quad-\cos\left(\frac{\pi}{2}-\theta\right)C_{-\theta,y}\\
&\Leftrightarrow \frac{\sqrt{3}dy_h}{2}\cos\left(\frac{\pi}{2}-\theta\right)+\sin\left(\frac{\pi}{2}-\theta\right)\frac{dy_h}{2}=\sin\left(\frac{\pi}{2}-\theta\right)(dx_h-C_{-\theta,x})\\
&\quad+\cos\left(\frac{\pi}{2}-\theta\right)C_{-\theta,y}\\
&\Leftrightarrow d\left(\frac{\sqrt{3}}{2}\cos\left(\frac{\pi}{2}-\theta\right)+\sin\left(\frac{\pi}{2}-\theta\right)\frac{1}{2}\right)y_h=\sin\left(\frac{\pi}{2}-\theta\right)(dx_h-C_{-\theta,x})\\
&\quad+\cos\left(\frac{\pi}{2}-\theta\right)C_{-\theta,y}\\
&\Leftrightarrow y_h=\frac{\sin\left(\frac{\pi}{2}-\theta\right)(dx_h-C_{-\theta,x})+\cos\left(\frac{\pi}{2}-\theta\right)C_{-\theta,y}}{d\left(\frac{\sqrt{3}}{2}\cos\left(\frac{\pi}{2}-\theta\right)+\sin\left(\frac{\pi}{2}-\theta\right)\frac{1}{2}\right)}\\
&\Leftrightarrow y_h=\frac{\sin\left(\frac{\pi}{2}-\theta\right)(dx_h-C_{-\theta,x})+\cos\left(\frac{\pi}{2}-\theta\right)C_{-\theta,y}}{d\left(\sin\left(\frac{\pi}{3}\right)\cos\left(\frac{\pi}{2}-\theta\right)+\sin\left(\frac{\pi}{2}-\theta\right)\cos\left(\frac{\pi}{3}\right)\right)}\\
&\Leftrightarrow y_h=\frac{\sin\left(\frac{\pi}{2}-\theta\right)(dx_h-C_{-\theta,x})+\cos\left(\frac{\pi}{2}-\theta\right)C_{-\theta,y}}{d\sin\left(\frac{\pi}{3}+\frac{\pi}{2}-\theta\right)}\\
&\Leftrightarrow L(x_h)=y_h=\frac{\sin\left(\frac{\pi}{2}-\theta\right)(dx_h-C_{-\theta,x})+\cos\left(\frac{\pi}{2}-\theta\right)C_{-\theta,y}}{d\sin\left(\frac{5\pi}{6}-\theta\right)}.
\end{aligned}
$$

$Y_1^S(x_h) = C_1(x_h)$ and $Y_2^S(x_h)$ can be either $\min(L(x_h), C_2(x_h))$ or $\min(L(x_h), C_2(x_h)) - 1$. As $t > \frac{s}{v}$, there can be a number of robots inside the rectangle $N_R(t,\theta) \geq 1$. If, for some x_h, $Y'(x_h) = \min(L(x_h), C_2(x_h)) = \lfloor L(x_h) \rfloor$, then the robot on $(x_h, Y'(x_h))$ is on the line $\overleftrightarrow{C_{-\theta}B'_{-\theta}}$. As this line belongs to the rectangle, the robot was already counted by $N_R(t,\theta)$. Hence,

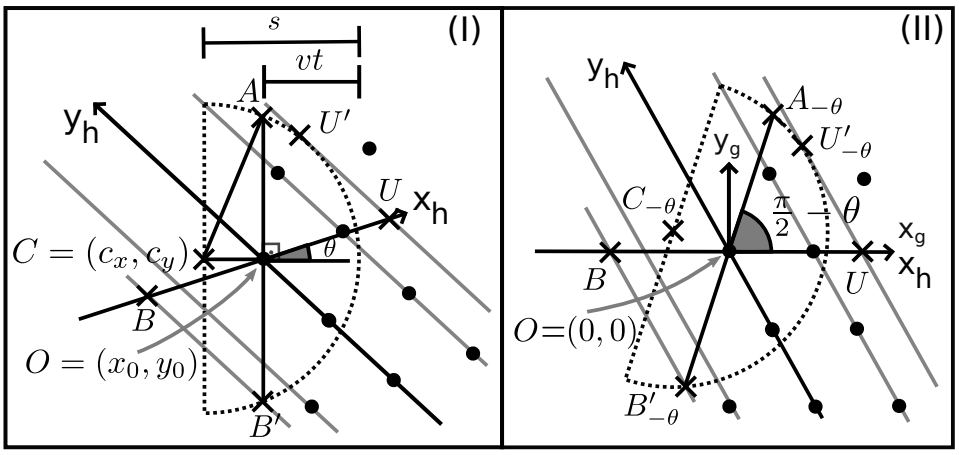


Figure B.12: Similar to the coordinate spaces of Figure B.11, but for $t \leq \frac{s}{v}$.

$$Y_2^S(x_h) = \begin{cases} \min(L(x_h), C_2(x_h)) - 1, & \text{if } \min(L(x_h), C_2(x_h)) = \lfloor L(x_h) \rfloor \\ & \text{and } t > \dfrac{s}{v}, \\ \min(L(x_h), C_2(x_h)), & \text{otherwise.} \end{cases}$$

The number of robots inside the semicircle is the number of integer coordinates (x_h, y_h) for x_h ranging from B to U and $y_h \in \left[\left\lceil Y_1^S(x_h)\right\rceil, \left\lfloor Y_2^S(x_h)\right\rfloor\right]$ for each x_h. Thus,

$$N_S(t,\theta) = \sum_{x_h=B}^{U} \sum_{y_h=\lceil Y_1^S(x_h)\rceil}^{\lfloor Y_2^S(x_h)\rfloor} 1 = \sum_{x_h=B}^{U} \left(\lfloor Y_2^S(x_h)\rfloor - \lceil Y_1^S(x_h)\rceil + 1\right).$$

Heed that the last summation can only be used when $\left\lfloor Y_2^S(x_h)\right\rfloor \geq \left\lceil Y_1^S(x_h)\right\rceil$, otherwise a negative number of robots would be summed.

Now, assume $t \leq \frac{s}{v}$. Then, the semicircle has centre at $C = (c_x, c_y) = (x_0 - (s - vt), y_0)$ as shown in the Figure B.12. In this figure, the rotation and hexagonal grid system centres are now (x_0, y_0). Notice also in (I) that $\triangle CAO$ is right with hypotenuse $\overline{CA}$ measuring s, and the horizontal cathetus $\overline{CO}$ measures $s - vt$. Now, as there is no rectangle part, consider the *last* robot of the rectangular part being the first robot to arrive at the target region, so $(l_x, l_y) = (x_0, y_0)$, and, by (B.28),

$$C_{-\theta} = \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{bmatrix} \begin{bmatrix} c_x - x_0 \\ c_y - y_0 \end{bmatrix}. \tag{B.33}$$

In the usual Euclidean coordinate space before the rotation about (x_0, y_0), consider the line $\overleftrightarrow{OA}$ perpendicular to the x-axis at $O = (x_0, y_0)$. This line represents the

perpendicular axis such that all the robots from it to the arc of the semicircle on its right are counted. From Figure B.12 (I),

$$r' = |\overline{AO}| = \sqrt{|\overline{CA}|^2 - |\overline{CO}|^2} = \sqrt{s^2 - (s - vt)^2} = \sqrt{2svt - (vt)^2}.$$

After the rotation by $-\theta$ about the point O, the maximum value for x_h is defined by the point U. The point U is chosen depending on the angles $\widehat{U'_{-\theta}OU}$ and $\widehat{A_{-\theta}OU}$. When the angle $\widehat{U'_{-\theta}OU}$ is greater than $\widehat{A_{-\theta}OU}$, the value of U is calculated in relation to $A_{-\theta}$, because the line parallel to y_h-axis intercepting $U'_{-\theta}$ is not inside the semicircle below $\overleftrightarrow{OA_{-\theta}}$ as shown in Figure B.13. It is only considered here robots inside the semicircle below the line $\overleftrightarrow{OA_{-\theta}}$, otherwise the robot on O would not be the first robot by assumption. In this case, any line parallel to y_h-axis crossing the semicircle below $\overline{OA_{-\theta}}$ must have its x_h-axis coordinate less than or equal to U, for example Q projected from P. For comparison, Figure B.12 (II) illustrates an example where U is chosen as the x_h-axis intersection with the line parallel to y_h-axis at $U'_{-\theta}$. As before, for the case $t > \frac{s}{v}$ (Figure B.11 (II)), the angle of $\overline{C_{-\theta}U'_{\theta}}$ in relation to x_h-axis is $\pi/6$, consequently,

$$\begin{aligned}
U'_{-\theta} &= C_{-\theta} + s\left(\cos\left(\frac{\pi}{6}\right), \sin\left(\frac{\pi}{6}\right)\right) \\
&= \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{bmatrix} \begin{bmatrix} c_x - x_0 \\ c_y - y_0 \end{bmatrix} + \begin{bmatrix} \frac{\sqrt{3}s}{2} \\ \frac{s}{2} \end{bmatrix} \\
&= \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} vt - s \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{\sqrt{3}s}{2} \\ \frac{s}{2} \end{bmatrix} \\
&= \left(\frac{\sqrt{3}s}{2} + \cos(\theta)(vt - s), \frac{s}{2} - \sin(\theta)(vt - s)\right),
\end{aligned}$$

from (B.33), and $\widehat{U'_{-\theta}OU}$ measures

$$\arctan\left(\frac{U'_{-\theta,y}}{U'_{-\theta,x}}\right) = \arctan\left(\frac{\frac{s}{2} - \sin(\theta)(vt - s)}{\frac{\sqrt{3}s}{2} + \cos(\theta)(vt - s)}\right).$$

$\widehat{A_{-\theta}OU}$ measures $\frac{\pi}{2} - \theta$, as show in Figure B.12 (II). Thence,

$$A_{-\theta} = \left(r' \cos\left(\frac{\pi}{2} - \theta\right), r' \sin\left(\frac{\pi}{2} - \theta\right)\right).$$

If $\arctan\left(\frac{U'_{-\theta,y}}{U'_{-\theta,x}}\right) \leq \widehat{A_{-\theta}OU} = \frac{\pi}{2} - \theta$, apply (B.29) on $U'_{-\theta}$ to get its x_h-axis coordinate

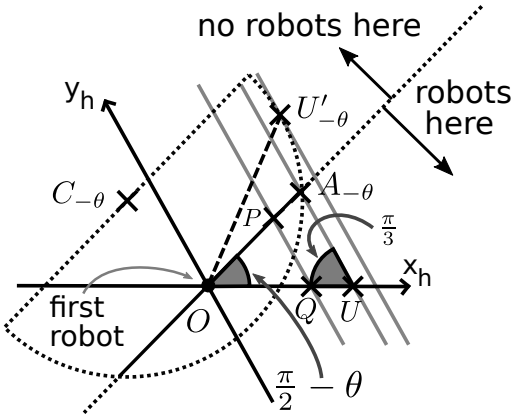


Figure B.13: An example of when the angle $\widehat{U'_{-\theta}OU}$ is greater than $\widehat{A_{-\theta}OU}$.

$$
\begin{aligned}
U &= \frac{1}{d}\left(\frac{\sqrt{3}s}{2} + \cos(\theta)(vt-s)\right) + \frac{1}{\sqrt{3}d}\left(\frac{s}{2} - \sin(\theta)(vt-s)\right) \\
&= \cos(\theta)\frac{vt-s}{d} + \frac{\sqrt{3}s}{2d} - \sin(\theta)\frac{vt-s}{\sqrt{3}d} + \frac{s}{2\sqrt{3}d} \\
&= \cos(\theta)\frac{vt-s}{d} - \sin(\theta)\frac{vt-s}{\sqrt{3}d} + \frac{\sqrt{3}\sqrt{3}s}{2\sqrt{3}d} + \frac{s}{2\sqrt{3}d} \\
&= \cos(\theta)\frac{vt-s}{d} - \sin(\theta)\frac{vt-s}{\sqrt{3}d} + \frac{3s}{2\sqrt{3}d} + \frac{s}{2\sqrt{3}d} \\
&= \cos(\theta)\frac{vt-s}{d} - \sin(\theta)\frac{vt-s}{\sqrt{3}d} + \frac{4s}{2\sqrt{3}d} \\
&= \cos(\theta)\frac{vt-s}{d} - \sin(\theta)\frac{vt-s}{\sqrt{3}d} + \frac{2s}{\sqrt{3}d} \\
&= \sqrt{3}\cos(\theta)\frac{vt-s}{\sqrt{3}d} - \sin(\theta)\frac{vt-s}{\sqrt{3}d} + \frac{2s}{\sqrt{3}d} \\
&= 2\left(\frac{\sqrt{3}}{2}\cos(\theta) - \frac{1}{2}\sin(\theta)\right)\frac{vt-s}{\sqrt{3}d} + \frac{2s}{\sqrt{3}d} \\
&= \frac{2\sin(\pi/3-\theta)(vt-s)}{\sqrt{3}d} + \frac{2s}{\sqrt{3}d},
\end{aligned}
$$

followed by applying floor function to it, as the integer coordinate less or equal to this value is needed. This is the same as (B.30) by using $(l_x, l_y) = (x_0, y_0)$, then (B.25) also applies when $\arctan\left(\frac{\frac{s}{2}-\sin(\theta)(vt-s)}{\frac{\sqrt{3}s}{2}+\cos(\theta)(vt-s)}\right) \leq \frac{\pi}{2} - \theta$.

If $\arctan\left(\frac{U'_{-\theta,y}}{U'_{-\theta,x}}\right) > \frac{\pi}{2} - \theta$, then there are no robots to consider on the parallel lines to y_h-axis between $U'_{-\theta}$ and $A_{-\theta}$, otherwise the robot at (x_0, y_0) would not be the first to arrive at the target region. Thus, if $\arctan\left(\frac{U'_{-\theta,y}}{U'_{-\theta,x}}\right) > \frac{\pi}{2} - \theta$, the x_h-coordinate for the point $A_{-\theta}$ on the hexagonal grid space is used, that is,

$$\begin{aligned} U &= \frac{1}{d}\left(r'\cos\left(\frac{\pi}{2}-\theta\right)\right) + \frac{1}{\sqrt{3}d}\left(r'\sin\left(\frac{\pi}{2}-\theta\right)\right) \\ &= \frac{2r}{\sqrt{3}d}\left(\frac{\sqrt{3}}{2}\cos\left(\frac{\pi}{2}-\theta\right) + \frac{1}{2}\sin\left(\frac{\pi}{2}-\theta\right)\right) \\ &= \frac{2r}{\sqrt{3}d}\left(\frac{\sqrt{3}}{2}\sin\left(\theta\right) + \frac{1}{2}\cos\left(\theta\right)\right) \\ &= \frac{2r}{\sqrt{3}d}\cos\left(\theta - \frac{\pi}{3}\right). \end{aligned}$$

then apply the floor function to yield the desired result in (B.26).

Now the minimum value for an integer x_h will be found such that a parallel-to-y_h-axis line is inside the semicircle and starting from the right of $\overleftrightarrow{OA}$ or on it. For the calculation of B, from Figure B.12 (II), similarly to how it was previously done,

$$\begin{aligned} B'_{-\theta} &= O + r'(\cos(-(\pi/2+\theta)), \sin(-(\pi/2+\theta)) \\ &= (r'\cos(\pi/2+\theta), -r'\sin(\pi/2+\theta)) \\ &= (-r'\sin(\theta), -r'\cos(\theta)), \end{aligned}$$

and, by (B.29), as B is the x_h-coordinate of the $B_{-\theta}$,

$$\begin{aligned} B &= \frac{1}{d}\left(-r'\sin(\theta)\right) + \frac{1}{\sqrt{3}d}\left(-r'\cos(\theta)\right) = -\frac{2r}{\sqrt{3}d}\left(\frac{1}{2}\cos(\theta) + \frac{\sqrt{3}}{2}\sin(\theta)\right) \\ &= -\frac{2r}{\sqrt{3}d}\left(\sin(\pi/6)\cos(\theta) + \cos(\pi/6)\sin(\theta)\right) \\ &= -\frac{2r}{\sqrt{3}d}\sin\left(\theta + \frac{\pi}{6}\right). \end{aligned}$$

Also, apply the ceiling function to yield the desired result in (B.24).

In this case, $C_1(x_h)$ and $C_2(x_h)$ are equal to (B.31) and (B.32), but $L(x_h)$ is different from the previous case. The line $\overleftrightarrow{OA_{-\theta}}$ for a point (x_g, y_g) in the Euclidean space is

$$y_g = \tan\left(\frac{\pi}{2}-\theta\right)x_g \Leftrightarrow y_g = \frac{\sin\left(\frac{\pi}{2}-\theta\right)}{\cos\left(\frac{\pi}{2}-\theta\right)}x_g$$

$$
\begin{aligned}
&\Leftrightarrow \frac{\sqrt{3}dy_h}{2} = \frac{\sin\left(\frac{\pi}{2}-\theta\right)}{\cos\left(\frac{\pi}{2}-\theta\right)}\left(dx_h - \frac{dy_h}{2}\right)\\
&\Leftrightarrow \frac{\sqrt{3}dy_h}{2}\cos\left(\frac{\pi}{2}-\theta\right) + \sin\left(\frac{\pi}{2}-\theta\right)\frac{dy_h}{2} = \sin\left(\frac{\pi}{2}-\theta\right)dx_h\\
&\Leftrightarrow y_h = \frac{2\sin\left(\frac{\pi}{2}-\theta\right)x_h}{\sqrt{3}\cos\left(\frac{\pi}{2}-\theta\right)+\sin\left(\frac{\pi}{2}-\theta\right)}\\
&\Leftrightarrow y_h = \frac{\sin\left(\frac{\pi}{2}-\theta\right)x_h}{\sin(\pi/3)\cos\left(\frac{\pi}{2}-\theta\right)+\cos(\pi/3)\sin\left(\frac{\pi}{2}-\theta\right)}\\
&\Leftrightarrow y_h = \frac{\sin\left(\frac{\pi}{2}-\theta\right)x_h}{\sin(\pi/3+\pi/2-\theta)}\\
&\Leftrightarrow L(x_h) = y_h = \frac{\sin\left(\frac{\pi}{2}-\theta\right)x_h}{\sin\left(\frac{5\pi}{6}-\theta\right)}.
\end{aligned}
$$

□

It follows that $\lim_{t\to\infty} f_h(t,\theta) = \lim_{t\to\infty}\frac{N_R(t,\theta)}{t} + \lim_{t\to\infty}\frac{N_S(t,\theta)-1}{t}$, by Definition 2. As shown below, this limit needs only the rectangle part, because N_S is limited by a semicircle with finite radius.

Lemma 11.

$$\lim_{t\to\infty}\frac{N_S(t,\theta)-1}{t} = 0.$$

Proof. As $t\to\infty$, $t > \frac{s}{v}$. By Lemma 10, $c_x = x_0+vt-s$, which is the x-axis coordinate of the right side of the rectangle. The robots are distant by d, so the last robot must be at most distant by d from the point (c_x, y_0). Hence, $x_0+vt-s-d \le l_x \le x_0+vt-s$, and $y_0-d \le l_y \le y_0+d$, so $0 = c_x-(x_0+vt-s) \le c_x-l_x \le c_x-(x_0+vt-s-d) = d$ and $-d \le y_0-l_y \le d$. Then, $-d \le C_{-\theta,x}, C_{-\theta,y} \le d$. Also, $\theta \in [0,\pi/3)$, so $-1/2 \le \cos(\pi/3-\theta) \le 1$. Thus,

$$
\begin{aligned}
B &= \left\lceil \frac{2(\sin(\pi/3-\theta)(c_x-l_x)+\cos(\pi/3-\theta)(y_0-l_y-s))}{\sqrt{3}d}\right\rceil\\
&\ge \left\lceil \frac{2(\cos(\pi/3-\theta)(-d-s))}{\sqrt{3}d}\right\rceil = \left\lceil \frac{-2\cos(\pi/3-\theta)(1+\frac{s}{d})}{\sqrt{3}}\right\rceil \ge \left\lceil \frac{-2(1+\frac{s}{d})}{\sqrt{3}}\right\rceil\\
&= \left\lceil -\frac{2}{\sqrt{3}} - \frac{s}{\sqrt{3}d}\right\rceil \ge -\frac{2}{\sqrt{3}} - \frac{s}{\sqrt{3}d},
\end{aligned}
$$

$$
\begin{aligned}
U &= \left\lfloor \frac{2(\sin(\pi/3-\theta)(c_x-l_x)+\cos(\pi/3-\theta)(y_0-l_y)+s)}{\sqrt{3}d} \right\rfloor \\
&\leq \left\lfloor \frac{2(\sin(\pi/3-\theta)d+\cos(\pi/3-\theta)d+s)}{\sqrt{3}d} \right\rfloor \leq \left\lfloor \frac{2(2d+s)}{\sqrt{3}d} \right\rfloor = \left\lfloor \frac{4}{\sqrt{3}} + \frac{2s}{\sqrt{3}d} \right\rfloor \\
&\leq \frac{4}{\sqrt{3}} + \frac{2s}{\sqrt{3}d},
\end{aligned}
$$

and for any integer $x_h \in [B, U]$, as $\Delta(x_h)$ cannot be negative,

$$
\begin{aligned}
0 \leq \Delta(x_h) &= 4s^2 - \left(\sqrt{3}(dx_h - C_{-\theta,x}) - C_{-\theta,y}\right)^2 \leq 4s^2, \\
\lceil Y_1^S(x_h) \rceil \geq Y_1^S(x_h) &= \frac{dx_h - C_{-\theta,x} + \sqrt{3}C_{-\theta,y} - \sqrt{\Delta(x_h)}}{2d} \\
&\geq \frac{dx_h - d - \sqrt{3}d - 2s}{2d} = \frac{x_h - 1 - \sqrt{3}}{2} - \frac{s}{d},
\end{aligned}
$$

and

$$
\begin{aligned}
\lfloor Y_2^S(x_h) \rfloor &\leq Y_2^S(x_h) \leq \min(L(x_h), C_2(x_h)) \leq C_2(x_h) \\
&= \frac{dx_h - C_{-\theta,x} + \sqrt{3}C_{-\theta,y} + \sqrt{\Delta(x_h)}}{2d} \leq \frac{dx_h + d + \sqrt{3}d + 2s}{2d} \\
&= \frac{x_h + 1 + \sqrt{3}}{2} + \frac{s}{d}.
\end{aligned}
$$

Thus,

$$
\begin{aligned}
0 \leq N_S(t,\theta) &= \sum_{x_h=B}^{U} \left(\lfloor Y_2^S(x_h) \rfloor - \lceil Y_1^S(x_h) \rceil + 1 \right) \\
&\leq \sum_{x_h=B}^{U} \left(\frac{x_h + 1 + \sqrt{3}}{2} + \frac{s}{d} - \left(\frac{x_h - 1 - \sqrt{3}}{2} - \frac{s}{d} \right) + 1 \right) \\
&= \sum_{x_h=B}^{U} \left(\frac{x_h + 1 + \sqrt{3}}{2} + \frac{s}{d} - \frac{x_h - 1 - \sqrt{3}}{2} + \frac{s}{d} + 1 \right) \\
&= \sum_{x_h=B}^{U} \left(\frac{\sqrt{3}}{2} + \frac{1}{2} + \frac{2s}{d} + \frac{1}{2} + \frac{\sqrt{3}}{2} + 1 \right) \\
&= \sum_{x_h=B}^{U} \left(\frac{2s}{d} + \frac{\sqrt{3} + 1 + 1 + \sqrt{3}}{2} + 1 \right) \\
&= \sum_{x_h=B}^{U} \left(\frac{2s}{d} + \frac{2\sqrt{3} + 4}{2} \right)
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{x_h=B}^{U} \left(\frac{2s}{d} + \sqrt{3} + 2 \right) = (U - B + 1) \left(\frac{2s}{d} + \sqrt{3} + 2 \right) \\
&\leq \left(\frac{4}{\sqrt{3}} + \frac{2s}{\sqrt{3}d} - \left(-\frac{2}{\sqrt{3}} - \frac{s}{\sqrt{3}d} \right) + 1 \right) \left(\frac{2s}{d} + \sqrt{3} + 2 \right) \\
&= \left(\frac{4}{\sqrt{3}} + \frac{2s}{\sqrt{3}d} + \frac{2}{\sqrt{3}} + \frac{s}{\sqrt{3}d} + 1 \right) \left(\frac{2s}{d} + \sqrt{3} + 2 \right) \\
&= \left(2\sqrt{3} + \frac{\sqrt{3}s}{d} + 1 \right) \left(\frac{2s}{d} + \sqrt{3} + 2 \right) \\
\Rightarrow 0 = \lim_{t\to\infty} \frac{-1}{t} &\leq \lim_{t\to\infty} \frac{N_S(t,\theta) - 1}{t} \\
&\leq \lim_{t\to\infty} \frac{1}{t} \left(\left(2\sqrt{3} + \frac{\sqrt{3}s}{d} + 1 \right) \left(\frac{2s}{d} + \sqrt{3} + 2 \right) - 1 \right) = 0.
\end{aligned}
$$

Hence, the result follows from the sandwich theorem. □

As $\lim_{t\to\infty} \frac{N_S(t,\theta) - 1}{t} = 0$, hereafter only the limit for the number of robots inside the rectangle is calculated. By Lemmas 5 to 9, if $n_l^+ - 1 < K'$

$$
\begin{aligned}
\lim_{t\to\infty} f_h(t,\psi) = \lim_{t\to\infty} \frac{1}{t} &\sum_{x_h=-n_l^-}^{n_l^-} \left(\lfloor Y_2^R(x_h) \rfloor - \lceil Y_1^R(x_h) \rceil + 1 \right) \\
&+ \lim_{t\to\infty} \frac{1}{t} \sum_{x_h=n_l^-+1}^{n_l^+-1} \left(\lfloor Y_2^R(x_h) \rfloor - \lceil Y_1^R(x_h) \rceil + 1 \right),
\end{aligned}
$$

otherwise,

$$
\begin{aligned}
\lim_{t\to\infty} f_h(t,\psi) = \lim_{t\to\infty} \frac{1}{t} &\sum_{x_h=-n_l^-}^{n_l^-} \left(\lfloor Y_2^R(x_h) \rfloor - \lceil Y_1^R(x_h) \rceil + 1 \right) \\
&+ \lim_{t\to\infty} \frac{1}{t} \sum_{x_h=n_l^-+1}^{K'-1} \left(\lfloor Y_2^R(x_h) \rfloor - \lceil Y_1^R(x_h) \rceil + 1 \right) \\
&+ \lim_{t\to\infty} \frac{1}{t} \sum_{x_h=K'}^{n_l^+-1} \left(\lfloor Y_2^R(x_h) \rfloor - \lceil Y_1^R(x_h) \rceil + 1 \right).
\end{aligned}
$$

To clarify, the third summation is zero in the case of $n_l^+ - 1 < K'$, while the second summation goes until $\min(n_l^+ - 1, K' - 1)$ in both cases. Each one will be individually

solved assuming $\psi \neq \pi/6$. Later, it will be seen that the final result holds for $\psi = \pi/6$ as well. The following lemmas will be useful soon.

Lemma 12. *Assume $\psi \neq \pi/6$.*

$$\lim_{t\to\infty} \frac{1}{t} \sum_{x_h=-n_l^-}^{n_l^-} \left(\lfloor Y_2^R(x_h) \rfloor - \lceil Y_1^R(x_h) \rceil + 1 \right) = 0.$$

Proof. As for any x, $x - 1 < \lfloor x \rfloor \leq x \leq \lceil x \rceil < x + 1$,

$$\begin{aligned}
&\lim_{t\to\infty} \frac{1}{t} \sum_{x_h=-n_l^-}^{n_l^-} \left(Y_2^R(x_h) - Y_1^R(x_h) - 1 \right) \\
&< \lim_{t\to\infty} \frac{1}{t} \sum_{x_h=-n_l^-}^{n_l^-} \left(\lfloor Y_2^R(x_h) \rfloor - \lceil Y_1^R(x_h) \rceil + 1 \right) \\
&\leq \lim_{t\to\infty} \frac{1}{t} \sum_{x_h=-n_l^-}^{n_l^-} \left(Y_2^R(x_h) - Y_1^R(x_h) + 1 \right).
\end{aligned}$$

By Lemma 6, the first and last summations do not depend on t, so both sides have limit equal to 0. By the sandwich theorem, the result is obtained. □

Lemma 13. *Assume $\psi \neq \pi/6$. For $K' = \left\lceil \frac{2(vt-s)\cos(\psi-\pi/6) - 2s\sin(|\psi-\pi/6|)}{\sqrt{3}d} \right\rceil$,*

$$\lim_{t\to\infty} \frac{1}{t} \sum_{x_h=K'}^{n_l^+-1} \left(\lfloor Y_2^R(x_h) \rfloor - \lceil Y_1^R(x_h) \rceil + 1 \right) = 0.$$

Proof. If $K' > n_l^+ - 1$, this limit is already zero, so this proof is focused on the other case. Analogously to the previous lemma,

$$\begin{aligned}
&\lim_{t\to\infty} \frac{1}{t} \sum_{x_h=K'}^{n_l^+-1} \left(Y_2^R(x_h) - Y_1^R(x_h) - 1 \right) \\
&< \lim_{t\to\infty} \frac{1}{t} \sum_{x_h=K'}^{n_l^+-1} \left(\lfloor Y_2^R(x_h) \rfloor - \lceil Y_1^R(x_h) \rceil + 1 \right) \qquad \text{(B.34)} \\
&\leq \lim_{t\to\infty} \frac{1}{t} \sum_{x_h=K'}^{n_l^+-1} \left(Y_2^R(x_h) - Y_1^R(x_h) + 1 \right).
\end{aligned}$$

For any constant c,

$$\lim_{t\to\infty}\frac{1}{t}\sum_{x_h=K'}^{n_l^+-1} c = 0, \tag{B.35}$$

because the number of x_h indexes in the summation is limited by a finite number of integer outcomes that depends on t. In other words, the number of indexes in the above summation is $n_l^+ - K'$ such that $\frac{4s\sin(|\psi-\pi/6|)}{\sqrt{3}d} - 1 < n_l^+ - K' \leq \frac{4s\sin(|\psi-\pi/6|)}{\sqrt{3}d} + 1$. The last inequality is obtained by counting how many x_h are used in the summation and knowing that $2y - 1 < \lfloor x+y \rfloor - \lceil x-y \rceil + 1 \leq 2y+1$ for any $x, y \in \mathbb{R}$. Thus, for any t, $n_l^+ - K'$ can only range from $\left\lceil \frac{4s\sin(|\psi-\pi/6|)}{\sqrt{3}d} \right\rceil - 1$ to $\left\lfloor \frac{4s\sin(|\psi-\pi/6|)}{\sqrt{3}d} \right\rfloor + 1$. This yields to three possible integer numbers, if $\frac{4s\sin(|\psi-\pi/6|)}{\sqrt{3}d} \in \mathbb{Z}$, or four, otherwise. Thus, a finite range of outcomes, none of them having t. Hence, for all outcomes, the limit on the left side of (B.35) is zero.

Assume $\psi > \pi/6$ (for $\psi < \pi/6$ the result is the same). From Lemma 7,

$$\begin{aligned} Y_2^R(x_h) - Y_1^R(x_h) &= \frac{\frac{2(vt-s)}{d\cos(\psi)} - 2x_h}{\sqrt{3}\tan(\psi) - 1} - \frac{\frac{2y_1}{d} + 2\tan(\psi)x_h}{\sqrt{3} + \tan(\psi)} \\ &= \frac{\frac{2(vt-s)}{d\cos(\psi)}}{\sqrt{3}\tan(\psi) - 1} - \frac{\frac{2y_1}{d}}{\sqrt{3} + \tan(\psi)} - \left(\frac{2}{\sqrt{3}\tan(\psi) - 1} \right. \\ &\left. + \frac{2\tan(\psi)}{\sqrt{3} + \tan(\psi)} \right) x_h. \end{aligned} \tag{B.36}$$

For the second term above, by (B.35), $\lim_{t\to\infty} \sum_{x_h=K'}^{n_l^+-1} \frac{\frac{2y_1}{d}}{\sqrt{3} + \tan(\psi)} = 0$.

For the first term,

$$\begin{aligned} \sum_{x_h=K'}^{n_l^+-1} \frac{1}{t} \frac{\frac{2(vt-s)}{d\cos(\psi)}}{\sqrt{3}\tan(\psi) - 1} &= \sum_{x_h=K'}^{n_l^+-1} \frac{2\left(v - \frac{s}{t}\right)}{d\cos(\psi)(\sqrt{3}\tan(\psi) - 1)} \\ = \sum_{x_h=K'}^{n_l^+-1} \frac{2\left(v - \frac{s}{t}\right)}{d(\sqrt{3}\sin(\psi) - \cos(\psi))} &= \sum_{x_h=K'}^{n_l^+-1} \frac{v - \frac{s}{t}}{d\sin(\psi - \pi/6)} \\ = \sum_{x_h=K'}^{n_l^+-1} \frac{v}{d\sin(\psi - \pi/6)} - \frac{1}{t} \sum_{x_h=K'}^{n_l^+-1} &\frac{s}{d\sin(\psi - \pi/6)}, \end{aligned} \tag{B.37}$$

due to $\frac{\sqrt{3}}{2}\sin(\psi) - \frac{1}{2}\cos(\psi) = \sin(\psi - \pi/6)$. Let L' be the number of terms on

the summation of (B.37). As discussed above, L' is an integer in $\left\{ \left\lceil \frac{4s\sin(|\psi-\pi/6|)}{\sqrt{3}d} \right\rceil - 1, \ldots, \left\lfloor \frac{4s\sin(|\psi-\pi/6|)}{\sqrt{3}d} \right\rfloor + 1 \right\}$, so

$$\sum_{x_h=K'}^{n_l^+-1} \frac{1}{t} \frac{\frac{2(vt-s)}{d\cos(\psi)}}{\sqrt{3}\tan(\psi)-1} = \frac{L'v}{d\sin(\psi-\pi/6)} - \frac{1}{t}\sum_{x_h=K'}^{n_l^+-1} \frac{s}{d\sin(\psi-\pi/6)}. \tag{B.38}$$

Also,

$$\begin{aligned}
&\frac{2}{\sqrt{3}\tan(\psi)-1} + \frac{2\tan(\psi)}{\sqrt{3}+\tan(\psi)} = \frac{2(\sqrt{3}+\tan(\psi)) + 2\tan(\psi)(\sqrt{3}\tan(\psi)-1)}{(\sqrt{3}\tan(\psi)-1)(\sqrt{3}+\tan(\psi))} \\
&= \frac{2\sqrt{3}+2\tan(\psi)+2\sqrt{3}\tan^2(\psi)-2\tan(\psi)}{(\sqrt{3}\tan(\psi)-1)(\sqrt{3}+\tan(\psi))} = \frac{2\sqrt{3}+2\sqrt{3}\tan^2(\psi)}{(\sqrt{3}\tan(\psi)-1)(\sqrt{3}+\tan(\psi))} \\
&= \frac{2\sqrt{3}(1+\tan^2(\psi))}{(\sqrt{3}\tan(\psi)-1)(\sqrt{3}+\tan(\psi))} = \frac{2\sqrt{3}\sec^2(\psi)}{(\sqrt{3}\tan(\psi)-1)(\sqrt{3}+\tan(\psi))} \\
&= \frac{2\sqrt{3}}{(\sqrt{3}\sin(\psi)-\cos(\psi))(\sqrt{3}\cos(\psi)+\sin(\psi))} = \frac{\sqrt{3}}{2\sin(\psi-\pi/6)\cos(\psi-\pi/6)}
\end{aligned}$$

as $1+\tan^2(\psi) = \sec^2(\psi)$ and $\frac{\sqrt{3}}{2}\cos(\psi) + \frac{1}{2}\sin(\psi) = \cos(\psi-\pi/6)$. Hence, for the last term in (B.36),

$$\begin{aligned}
&\frac{1}{t}\sum_{x_h=K'}^{n_l^+-1} \frac{\sqrt{3}}{2\sin(\psi-\pi/6)\cos(\psi-\pi/6)} x_h \\
&= \frac{1}{t}\frac{\sqrt{3}}{2\sin(\psi-\pi/6)\cos(\psi-\pi/6)} \frac{(n_l^+-1+K')(n_l^+-K')}{2} \\
&= \frac{\sqrt{3}LG}{4T\sin(\psi-\pi/6)\cos(\psi-\pi/6)},
\end{aligned} \tag{B.39}$$

for an integer $G = n_l^+ - 1 + K'$. As $2x-1 < \lfloor x+y \rfloor + \lceil x-y \rceil < 2x+1$ for any $x, y \in \mathbb{R}$, $G \in \left(\frac{4(vt-s)\cos(\psi-\pi/6)}{\sqrt{3}d} - 1, \frac{4(vt-s)\cos(\psi-\pi/6)}{\sqrt{3}d} + 1 \right)$.

For the lowest bound on G, using (B.38) and (B.39)

$$\begin{aligned}
&\lim_{t\to\infty} \frac{1}{t} \sum_{x_h=K'}^{n_l^+-1} \left(Y_2^R(x_h) - Y_1^R(x_h) \right) \\
&= \lim_{t\to\infty} \frac{1}{t} \sum_{x_h=K'}^{n_l^+-1} \left(\frac{\frac{2(vt-s)}{d\cos(\psi)}}{\sqrt{3}\tan(\psi)-1} - \left(\frac{2}{\sqrt{3}\tan(\psi)-1} + \frac{2\tan(\psi)}{\sqrt{3}+\tan(\psi)} \right) x_h \right)
\end{aligned}$$

$$= \lim_{t\to\infty} \left(\frac{L'v}{d\sin(\psi-\pi/6)} - \frac{1}{t}\sum_{x_h=K'}^{n_l^+-1} \frac{s}{d\sin(\psi-\pi/6)} \right.$$
$$\left. - \frac{\sqrt{3}L'\left(\frac{4(vt-s)\cos(\psi-\pi/6)}{\sqrt{3}d}-1\right)}{4T\sin(\psi-\pi/6)\cos(\psi-\pi/6)} \right)$$
$$= \lim_{t\to\infty} \left(\frac{L'v}{d\sin(\psi-\pi/6)} - \frac{L'\frac{4(vt-s)\cos(\psi-\pi/6)}{d} - \sqrt{3}L'}{4T\sin(\psi-\pi/6)\cos(\psi-\pi/6)} - \frac{1}{t}\sum_{x_h=K'}^{n_l^+-1} \frac{s}{d\sin(\psi-\pi/6)} \right)$$
$$= \lim_{t\to\infty} \left(\frac{L'v}{d\sin(\psi-\pi/6)} - \frac{L'v}{d\sin(\psi-\pi/6)} + \frac{\sqrt{3}L'}{4T\sin(\psi-\pi/6)\cos(\psi-\pi/6)} - \right.$$
$$\left. \frac{1}{t}\sum_{x_h=K'}^{n_l^+-1} \frac{s}{d\sin(\psi-\pi/6)} \right)$$
$$= \lim_{t\to\infty} \left(\frac{\sqrt{3}L'}{4T\sin(\psi-\pi/6)\cos(\psi-\pi/6)} - \frac{1}{t}\sum_{x_h=K'}^{n_l^+-1} \frac{s}{d\sin(\psi-\pi/6)} \right) = 0,$$

due to (B.35) on the second term and, as

$$L' \in \left\{ \left\lceil \frac{4s\sin(|\psi-\pi/6|)}{\sqrt{3}d} \right\rceil - 1, \ldots, \left\lfloor \frac{4s\sin(|\psi-\pi/6|)}{\sqrt{3}d} \right\rfloor + 1 \right\},$$

no element in this finite set has the term t.

For the highest bound on G, the same limit is obtained. Hence, by the sandwich theorem applied on the results for both bounds of G,

$$\lim_{t\to\infty} \frac{1}{t} \sum_{x_h=K'}^{n_l^+-1} \left(Y_2^R(x_h) - Y_1^R(x_h) \right) = 0. \tag{B.40}$$

Using (B.40) and (B.35) on the bounds of (B.34) and the sandwich theorem again concludes with the desired value. □

Lemma 14. *Assume* $\psi \neq \pi/6$.

$$\lim_{t\to\infty} \frac{1}{t} \sum_{x_h=n_l^-+1}^{\min(n_l^+-1, K'-1)} \left(\lfloor Y_2^R(x_h) \rfloor - \lceil Y_1^R(x_h) \rceil + 1 \right)$$

exists and is bounded by

$$\left(\frac{4vs}{\sqrt{3}d^2} - \frac{2v\cos(\psi - \pi/6)}{\sqrt{3}d}, \frac{4vs}{\sqrt{3}d^2} + \frac{2v\cos(\psi - \pi/6)}{\sqrt{3}d}\right].$$

Proof. The next lemmas will be useful for proving this lemma.

Lemma 15. *For any $a, b > 0, a\lfloor x \rfloor - b\lfloor y \rfloor < ax - by + a + b$.*

Proof. As mentioned before, by the definition of floor function $\lfloor x \rfloor = x - frac(x)$, where $frac$ is the function that returns the fractional part of the number x, such that $0 \leq frac(x) < 1$ (Graham, Knuth, and Patashnik, 1994),

$$\begin{aligned} a\lfloor x \rfloor - b\lfloor y \rfloor &= ax - afrac(x) - by + bfrac(y) \\ &< ax - by + b - afrac(x) && [\text{because } frac(y) < 1] \\ &< ax - by + b + a && [\text{as } -afrac(x) \leq 0 < a]. \end{aligned}$$

□

Lemma 16. *Let $c, d, A_1, B_1, A_2, B_2 \in \mathbb{R}$, $c > 0$ and $I_1 \in \mathbb{Z}$. Then, the limit below exists:*

$$\lim_{n\to\infty} \sum_{i=I_1+1}^{\lfloor cn+d \rfloor} \frac{frac(-(A_1 i + B_1)) + frac(A_2 i + B_2)}{n}.$$

Proof. For convergence, it is shown that for $R(i) = frac(-(A_1 i + B_1)) + frac(A_2 i + B_2)$, $(a_n)_{n\in\mathbb{N}^*} = \left(\sum_{i=I_1+1}^{\lfloor cn+d \rfloor} \frac{R(i)}{n}\right)_{n\in\mathbb{N}^*}$ is a Cauchy sequence. Take $\epsilon > 0$ and choose $N > \frac{4|I_1 - d + 1|}{\epsilon}$. Let $n, m \in \mathbb{N}^*$ and $n > m > N$. Then,

$$\begin{aligned} |a_n - a_m| &= \left| \sum_{i=I_1+1}^{\lfloor cn+d \rfloor} \frac{R(i)}{n} - \sum_{i=I_1+1}^{\lfloor cm+d \rfloor} \frac{R(i)}{m} \right| = \left| \frac{1}{nm} \left(m \sum_{i=I_1+1}^{\lfloor cn+d \rfloor} R(i) - n \sum_{i=I_1+1}^{\lfloor cm+d \rfloor} R(i) \right) \right| \\ &= \left| \frac{1}{nm} \left(m \sum_{i=\lfloor cm+d \rfloor+1}^{\lfloor cn+d \rfloor} R(i) + (m-n) \sum_{i=I_1+1}^{\lfloor cm+d \rfloor} R(i) \right) \right| && [\text{as } \lfloor cn+d \rfloor > \lfloor cm+d \rfloor] \\ &< 2 \left| \frac{m(\lfloor cn+d \rfloor - (\lfloor cm+d \rfloor + 1) + 1) + (m-n)(\lfloor cm+d \rfloor - (I_1+1) + 1)}{nm} \right| \\ & && [\text{as } R(i) < 2 \text{ for any } i] \end{aligned}$$

$$
\begin{aligned}
&= 2\left|\frac{m\lfloor cn+d\rfloor - m(\lfloor cm+d\rfloor+1)+m+m\lfloor cm+d\rfloor - m(I_1+1)+m-n\lfloor cm+d\rfloor}{nm}\right.\\
&\quad \left.+\frac{n(I_1+1)-n}{nm}\right|\\
&= 2\left|\frac{m\lfloor cn+d\rfloor - m\lfloor cm+d\rfloor - m+m+m\lfloor cm+d\rfloor - mK-m+m-n\lfloor cm+d\rfloor}{nm}\right.\\
&\quad \left.+\frac{nK+n-n}{nm}\right|\\
&= 2\left|\frac{m\lfloor cn+d\rfloor - mK-n\lfloor cm+d\rfloor+nK}{nm}\right| = 2\left|\frac{m\lfloor cn+d\rfloor - n\lfloor cm+d\rfloor-(m-n)I_1}{nm}\right|\\
&< 2\left|\frac{m(cn+d)-n(cm+d)+m+n-(m-n)I_1}{nm}\right| \qquad \text{[Lemma 15]}\\
&= 2\left|\frac{(m-n)(d-I_1)+m+n}{nm}\right|\\
&= 2\left|\frac{(n-m)(I_1-d)+m+n}{nm}\right| < 2\left|\frac{(n+m)(I_1-d)+m+n}{nm}\right|\\
&= 2\left|\frac{(m+n)(I_1-d+1)}{nm}\right| = 2|I_1-d+1|\frac{m+n}{nm} = 2|I_1-d+1|\left(\frac{1}{n}+\frac{1}{m}\right)\\
&< 2|I_1-d+1|\frac{2}{N} = \frac{4|I_1-d+1|}{N} < \epsilon.
\end{aligned}
$$

□

To prove the existence of the limit in Lemma 16, note that $\lceil x\rceil = x + frac(-x)$, for any real number x (heed that using this definition of $frac$, $frac(1.7) = 0.7$ and $frac(-1.7) = 0.3$), because

$$
\begin{aligned}
frac(-x) &= -x - \lfloor -x\rfloor && \text{[def. of } \lfloor -x\rfloor]\\
&= -x - (-\lceil x\rceil) && [\lfloor -x\rfloor = -\lceil x\rceil]\\
&= -x + \lceil x\rceil\\
\Leftrightarrow \lceil x\rceil &= x + frac(-x).
\end{aligned}
$$

Thus,

$$
\begin{aligned}
&\lim_{t\to\infty}\frac{1}{t}\sum_{x_h=n_l^-+1}^{\min(n_l^+-1,K'-1)}\left(\lfloor Y_2^R(x_h)\rfloor-\lceil Y_1^R(x_h)\rceil+1\right)\\
&=\lim_{t\to\infty}\frac{1}{t}\sum_{x_h=n_l^-+1}^{\min(n_l^+-1,K'-1)}\left(Y_2^R(x_h)-Y_1^R(x_h)-frac\left(-Y_1^R(x_h)\right)\right.\\
&\left.-frac\left(Y_2^R(x_h)\right)+1\right)\\
&=\lim_{t\to\infty}\frac{1}{t}\sum_{x_h=n_l^-+1}^{\min(n_l^+-1,K'-1)}\left(Y_2^R(x_h)-Y_1^R(x_h)+1\right)-\\
&\lim_{t\to\infty}\frac{1}{t}\sum_{x_h=n_l^-+1}^{\min(n_l^+-1,K'-1)}\left(frac\left(-Y_1^R(x_h)\right)+frac\left(Y_2^R(x_h)\right)\right).
\end{aligned}
$$

The limit of the first term above exists and its value is presented below on (B.43). The existence of the limit for the second term was shown by Lemma 16 for any outcome of $\min(n_l^+-1,K'-1)$, because, if $\left\lfloor\frac{2(vt-s)\cos(\pi/6-\theta)+2s\sin(|\pi/6-\theta|)}{\sqrt{3}d}\right\rfloor=n_l^+-1\leq K'-1$, then $c=\frac{2v\cos(\psi-\pi/6)}{\sqrt{3}d}$ and $d=\frac{2s}{\sqrt{3}d}(\sin(|\pi/6-\theta|)-\cos(\pi/6-\theta))$ on the Lemma 16. If $n_l^+-1>K'-1=\left\lceil\frac{2}{\sqrt{3}d}((vt-s)\cos(\psi-\pi/6)-2s\sin(|\psi-\pi/6|))-1\right\rceil$, as for any x, $\lceil x\rceil=\lfloor x\rfloor$ or $\lceil x\rceil=\lfloor x\rfloor+1$ depending on whether x is an integer or not, then $K'-1=\left\lfloor\frac{2(vt-s)\cos(\psi-\pi/6)}{\sqrt{3}d}-\frac{2s\sin(|\psi-\pi/6|)}{\sqrt{3}d}-1\right\rfloor$ or $K'-1=\left\lfloor\frac{2(vt-s)\cos(\psi-\pi/6)}{\sqrt{3}d}-\frac{2s\sin(|\psi-\pi/6|)}{\sqrt{3}d}\right\rfloor$. For both cases, on the Lemma 16 $c=\frac{2v\cos(\psi-\pi/6)}{\sqrt{3}d}$ as well, but for the former case, $d=-\frac{2s(\sin(|\psi-\pi/6|)+\cos(\pi/6-\theta))}{\sqrt{3}d}-1$, and for the latter, $d=-\frac{2s(\sin(|\psi-\pi/6|)+\cos(\pi/6-\theta))}{\sqrt{3}d}$.

To get the bounds,

$$
\begin{aligned}
&\lim_{t\to\infty}\frac{1}{t}\sum_{x_h=n_l^-+1}^{\min(n_l^+-1,K'-1)}\left(Y_2^R(x_h)-Y_1^R(x_h)-1\right)\\
&<\lim_{t\to\infty}\frac{1}{t}\sum_{x_h=n_l^-+1}^{\min(n_l^+-1,K'-1)}\left(\lfloor Y_2^R(x_h)\rfloor-\lceil Y_1^R(x_h)\rceil+1\right) \qquad \text{(B.41)}\\
&\leq\lim_{t\to\infty}\frac{1}{t}\sum_{x_h=n_l^-+1}^{\min(n_l^+-1,K'-1)}\left(Y_2^R(x_h)-Y_1^R(x_h)+1\right),
\end{aligned}
$$

and by Lemma 9, as $t\to\infty$,

$$\begin{aligned} Y_2^R(x_h) - Y_1^R(x_h) &= \frac{\frac{y_2}{d} + \tan(\psi)x_h}{\frac{\sqrt{3}+\tan(\psi)}{2}} - \frac{\frac{y_1}{d} + \tan(\psi)x_h}{\frac{\sqrt{3}+\tan(\psi)}{2}} \\ &= \frac{\frac{y_2-y_1}{d}}{\frac{\sqrt{3}+\tan(\psi)}{2}} = \frac{\frac{2s}{d\cos(\psi)}}{\frac{\sqrt{3}+\tan(\psi)}{2}} = \frac{4s}{d(\sqrt{3}\cos(\psi)+\sin(\psi))} \\ &= \frac{2s}{d\cos(\psi-\pi/6)}, \end{aligned}$$

by (B.4).

For the first limit at (B.41) in the case of $\min(n_l^+ - 1, K' - 1) = n_l^+ - 1$,

$$\begin{aligned} &\lim_{t\to\infty} \frac{1}{t} \sum_{x_h=n_l^-+1}^{n_l^+-1} \left(Y_2^R(x_h) - Y_1^R(x_h) - 1\right) = \lim_{t\to\infty} \frac{1}{t} \sum_{x_h=n_l^-+1}^{n_l^+-1} \left(\frac{2s}{d\cos(\psi-\pi/6)} - 1\right) \\ &= \lim_{t\to\infty} \frac{1}{t} \left(n_l^+ - n_l^- - 1\right) \left(\frac{2s}{d\cos(\psi-\pi/6)} - 1\right) \\ &= \left(\frac{2s}{d\cos(\psi-\pi/6)} - 1\right) \left(\lim_{t\to\infty} \frac{1}{t} n_l^+ - \lim_{t\to\infty} \frac{1}{t}(n_l^- - 1)\right) \\ &= \left(\frac{2s}{d\cos(\psi-\pi/6)} - 1\right) \lim_{t\to\infty} \frac{1}{t} n_l^+ = \left(\frac{2s}{d\cos(\psi-\pi/6)} - 1\right) \frac{2v\cos(\psi-\pi/6)}{\sqrt{3}d}. \\ &= \frac{4vs}{\sqrt{3}d^2} - \frac{2v\cos(\psi-\pi/6)}{\sqrt{3}d}. \end{aligned} \tag{B.42}$$

Above $\lim_{t\to\infty} \frac{1}{t} n_l^+ = \frac{2v\cos(\psi-\pi/6)}{\sqrt{3}d}$ is derived by using the sandwich theorem and the inequality $x - 1 < \lfloor x \rfloor \leq x$ to get the bounds on n_l^+.

Similarly, for the last limit at (B.41) in the case of $\min(n_l^+ - 1, K' - 1) = n_l^+ - 1$,

$$\lim_{t\to\infty} \frac{1}{t} \sum_{x_h=n_l^-+1}^{n_l^+-1} \left(Y_2^R(x_h) - Y_1^R(x_h) + 1\right) = \frac{4vs}{\sqrt{3}d^2} + \frac{2v\cos(\psi-\pi/6)}{\sqrt{3}d}.$$

The limits above in the case of $\min(n_l^+ - 1, K' - 1) = K' - 1$ yields the same result because of the sandwich theorem, the inequality $x \leq \lceil x \rceil < x + 1$, and

$$\begin{aligned}
&\frac{2v\cos(\psi-\pi/6)}{\sqrt{3}d} = \lim_{t\to\infty}\frac{1}{t}\left(\frac{2(vt-s)\cos(\psi-\pi/6)-2s\sin(|\psi-\pi/6|)}{\sqrt{3}d}\right)\\
&\leq \lim_{t\to\infty}\frac{1}{t}(K'-1) = \lim_{t\to\infty}\frac{1}{t}K'\\
&= \lim_{t\to\infty}\frac{1}{t}\left\lceil\frac{2(vt-s)\cos(\psi-\pi/6)-2s\sin(|\psi-\pi/6|)}{\sqrt{3}d}\right\rceil\\
&< \lim_{t\to\infty}\frac{1}{t}\left(\frac{2(vt-s)\cos(\psi-\pi/6)-2s\sin(|\psi-\pi/6|)}{\sqrt{3}d}+1\right)\\
&= \frac{2v\cos(\psi-\pi/6)}{\sqrt{3}d},
\end{aligned}$$

so, $\lim_{t\to\infty}\frac{1}{t}K' = \lim_{t\to\infty}\frac{1}{t}n_l^+$. Consequently, the limit below exists and

$$\lim_{t\to\infty}\frac{1}{t}\sum_{x_h=n_l^-+1}^{\min(n_l^+-1,K'-1)}\left(Y_2^R(x_h)-Y_1^R(x_h)+1\right) = \frac{4vs}{\sqrt{3}d^2}+\frac{2v\cos(\psi-\frac{\pi}{6})}{\sqrt{3}d}. \tag{B.43}$$

Finally, using the bounds provided by (B.42) and (B.43) the expected result is obtained. □

By Lemmas 12, 13 and 14 it is obtained for $\psi \neq \pi/6$

$$\lim_{t\to\infty} f_h(t,\psi) \in \left(\frac{4vs}{\sqrt{3}d^2}-\frac{2v\cos(\psi-\pi/6)}{\sqrt{3}d}, \frac{4vs}{\sqrt{3}d^2}+\frac{2v\cos(\psi-\pi/6)}{\sqrt{3}d}\right]. \tag{B.44}$$

For $\psi = \pi/6$, by (B.19),

$$\begin{aligned}
\lim_{t\to\infty}\frac{1}{t}\sum_{x_h=0}^{\left\lfloor\frac{2(vt-s)}{\sqrt{3}d}\right\rfloor}\left(\frac{\sqrt{3}y_2+dx_h}{2d}-\frac{\sqrt{3}y_1+dx_h}{2d}-1\right) &< \lim_{t\to\infty} f_h(t,\pi/6)\\
&\leq \lim_{t\to\infty}\frac{1}{t}\sum_{x_h=0}^{\left\lfloor\frac{2(vt-s)}{\sqrt{3}d}\right\rfloor}\left(\frac{\sqrt{3}y_2+dx_h}{2d}-\frac{\sqrt{3}y_1+dx_h}{2d}+1\right),
\end{aligned}$$

with

$$\lim_{t\to\infty}\frac{1}{t}\sum_{x_h=0}^{\left\lfloor\frac{2(vt-s)}{\sqrt{3}d}\right\rfloor}\left(\frac{\sqrt{3}y_2+dx_h}{2d}-\frac{\sqrt{3}y_1+dx_h}{2d}+1\right)$$
$$=\lim_{t\to\infty}\frac{1}{t}\sum_{x_h=0}^{\left\lfloor\frac{2(vt-s)}{\sqrt{3}d}\right\rfloor}\left(\frac{\sqrt{3}s}{d\cos(\pi/6)}+1\right)=\lim_{t\to\infty}\frac{1}{t}\sum_{x_h=0}^{\left\lfloor\frac{2(vt-s)}{\sqrt{3}d}\right\rfloor}\left(\frac{2s}{d}+1\right)$$
$$=\lim_{t\to\infty}\frac{1}{t}\left\lfloor\frac{2(vt-s)}{\sqrt{3}d}+1\right\rfloor\left(\frac{2s}{d}+1\right)=\frac{2v}{\sqrt{3}d}\left(\frac{2s}{d}+1\right),$$

from (B.4) and, as similarly done before, $\lim_{t\to\infty}\frac{1}{t}\left\lfloor\frac{2(vt-s)}{\sqrt{3}d}+1\right\rfloor=\frac{2v}{\sqrt{3}d}$ by using the sandwich theorem and the inequality $x-1<\lfloor x\rfloor\leq x$ to get the bounds on the floor function; and

$$\lim_{t\to\infty}\frac{1}{t}\sum_{x_h=0}^{\left\lfloor\frac{2(vt-s)}{\sqrt{3}d}\right\rfloor}\left(\frac{\sqrt{3}y_2+dx_h}{2d}-\frac{\sqrt{3}y_1+dx_h}{2d}-1\right)=\frac{2v}{\sqrt{3}d}\left(\frac{2s}{d}-1\right).$$

Accordingly,

$$\lim_{t\to\infty}f_h(t,\pi/6)\in\left(\frac{2v}{\sqrt{3}d}\left(\frac{2s}{d}-1\right),\frac{2v}{\sqrt{3}d}\left(\frac{2s}{d}+1\right)\right],$$

which are the same values in (B.44) if $\psi=\pi/6$ is used. Lemmas 4–9, 12, 13 and 14 used ψ, so, after replacing ψ by $\pi/3-\theta$, the proof of the Proposition 6 is concluded.

B.5 Proof of Proposition 8

Using the touch and run strategy, each lane is distant by at least d from each other. However, the minimum distance between robots on the same lane d_o must be checked at the beginning of the curved path, as their distance decreases if assuming constant linear speed. Two cases are distinguished based on Figure B.14:

1. $|\overline{ED}|<d$: Two robots cannot be on the lane curved path;
2. $|\overline{ED}|\geq d$: More than one robot can occupy the lane curved path.

In this figure, the red line represents the trajectory of robots in one lane. α is the central angle for the lane. The dashed blue circle of centre A is the target. C is the centre of the circle of radius r from the circular trajectory. The grey circle of centre C has a radius of $r+d/2$. Points D and E represent the connection between the curved

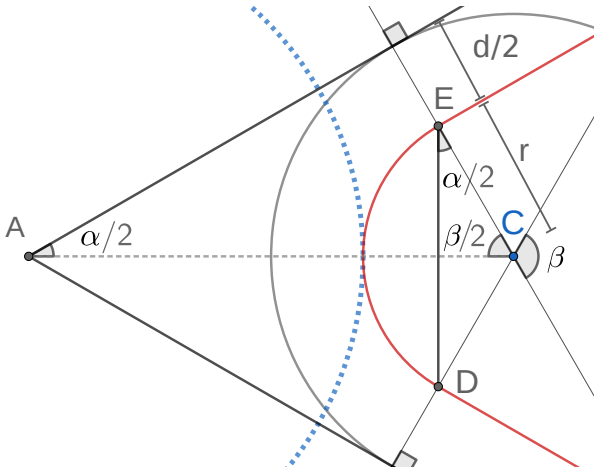


Figure B.14: Relationship between the curved path and the distance between the robots.

path and the straight path. $\beta = \pi - \alpha$ due to the symmetry and the fact that the sum of the angles of $\triangle ECD$ is equal to π.

The two identified cases affect the minimum distance between robots, d_o, such that they can follow the trajectory without decreasing their linear speed. In both cases, they need to satisfy the minimum distance d if they are turning on the curved path. From Figure B.14,

$$|\overline{ED}| = 2r \sin\left(\frac{\beta}{2}\right) = 2r \sin\left(\frac{\pi}{2} - \frac{\alpha}{2}\right) = 2r \cos\left(\frac{\alpha}{2}\right). \tag{B.45}$$

In case 1, in Figure B.15a, two points t and U are defined on the lane such that the distance between them is $|\overline{TU}| = d$ and their distances to the target are equal. The robots R_1 and R_2 are the black dots on the red line representing the trajectory. If the delay between R_1 and R_2 is less than the time for a robot to run from t to U following the red trajectory, there will be some instant in which R_1 and R_2 will be vertically aligned. Their positions at that instant are represented by grey dots in front of them. Hence, their distance would be less than d. The right triangle TVE has side $\overline{TV}$, which can be measured using $\overline{ED}$. The delay between one robot at t and another at U is equal to

$$\Delta t_1 = \frac{|\overline{TE}| + |\widehat{ED}| + |\overline{DU}|}{v},$$

that is, the time for running through the straight line TE, the curved path ED and the straight line DU.

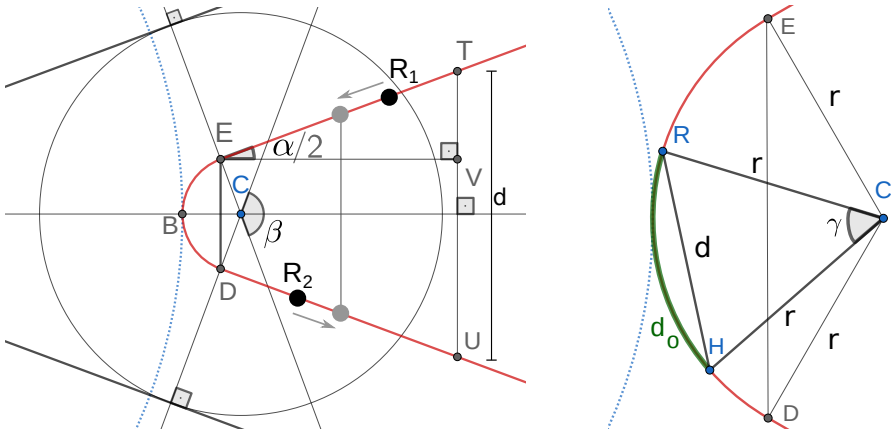


Figure B.15: Enlargements of Figure B.14. (a) $|\overline{ED}| < d$. (b) $|\overline{ED}| \geq d$.

For any delay less than Δt_1 between two robots, say R_1 and R_2, there is an instant of time when R_1 is on the path between B and t and R_2 is on the path between B and U, and they are vertically aligned (Figure B.15a). In this case, the distance between R_1 and R_2 is below $|\overline{TU}|$, so they do not respect the minimum distance d between them. Hence, the minimum delay between two robots in case 1 is Δt_1.

From Figure B.14, $|\widehat{ED}| = r\beta = r(\pi - \alpha)$. For calculating the value of $|\overline{TE}|$ and $|\overline{DU}|$ from Figure B.15a, observe that $|\overline{TE}| = |\overline{DU}|$ by symmetry. Thus,

$$\begin{aligned} |\overline{VT}| &= \frac{d}{2} - \frac{|\overline{ED}|}{2} && \text{[From Figure B.15a]} \\ &= \frac{d}{2} - r\cos\left(\frac{\alpha}{2}\right) && \text{[From (B.45)].} \end{aligned}$$

As $\triangle TVE$ is right, $|\overline{TE}| = \frac{|\overline{VT}|}{\sin(\alpha/2)}$. Thence,

$$\Delta t_1 = \frac{r(\pi - \alpha) + 2\frac{d/2 - r\cos(\alpha/2)}{\sin(\alpha/2)}}{v} = \frac{r(\pi - \alpha)}{v} + \frac{d - 2r\cos(\alpha/2)}{v\sin(\alpha/2)}$$

and

$$d_o = \max\left(d, v\Delta t_1\right) = \max\left(d, r(\pi - \alpha) + \frac{d - 2r\cos\left(\alpha/2\right)}{\sin(\alpha/2)}\right).$$

Above the max function is used because the result of $v\Delta t_1$ can still be less than d, depending on α, r and d.

In case 2, one has to check the minimum distance d when two robots are on the circular part $\widehat{ED}$ in Figure B.15b. Here, d_o denotes the minimum arc length for two

robots located at any two points R and H on $\widehat{ED}$ such that they are distant by at least d. γ is the angle defining the arc d_o for the circle of centre C. From this figure, $\triangle CRH$ is isosceles, so $\gamma = 2\arcsin\left(\frac{d}{2r}\right)$. Thus, to keep constant linear speed, the delay between two robots in this case is

$$\Delta t_2 = \frac{d_o}{v} = \frac{r\gamma}{v} = \frac{2r}{v}\arcsin\left(\frac{d}{2r}\right).$$

Then,

$$d_o = \max\left(d, v\Delta t_2\right) = \max\left(d, 2r\arcsin\left(\frac{d}{2r}\right)\right).$$

The max function is applied for a similar reason as exposed before. After rearranging, (4.17) and (4.18) are obtained.

For calculating the throughput $f_t(K,t)$ for K lanes and a given time t after the arrival of the first robot, the number of robots reaching the target region by the time t is obtained, then the Definition 2 is applied. As it was assumed that the first robot of every lane begins at the same distance from the target, at time $t = 0$ there are K robots simultaneously arriving. Then, after d_o/v units of time, there are K more robots arriving and this keeps happening every d_o/v units of time. Denote $N(K,t)$ the total number of robots that have arrived at the target region from K lanes by time t. Thus,

$$N(K,t) = K\left\lfloor \frac{t}{\frac{d_0}{v}} + 1 \right\rfloor = K\left\lfloor \frac{vt}{d_o} + 1 \right\rfloor,$$

so, by Definition 2,

$$f_t(K,t) = \frac{1}{t}\left(K\left\lfloor \frac{vt}{d_o} + 1 \right\rfloor - 1\right).$$

As for every number x, $\lfloor x \rfloor = x - frac(x)$ and $0 \leq frac(x) < 1$, then distributing $\frac{1}{t}$ for each term,

$$\begin{aligned} f_t(K) &= \lim_{t\to\infty} f_t(K,t) = \lim_{t\to\infty} \frac{1}{t}\left(K\left(\frac{vt}{d_o} + 1\right) - frac\left(\frac{vt}{d_o} + 1\right) - 1\right) \\ &= \lim_{t\to\infty} \frac{K}{t}\left(\frac{vt}{d_o} + 1\right) = \frac{Kv}{d_o}. \end{aligned}$$

Appendix C

Additional graphs for Chapter 4

C.1 Additional screenshots

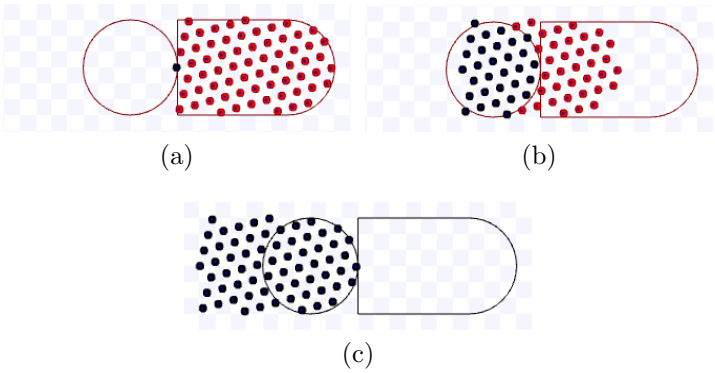
Figures C.1–C.7 presents screenshots from experiments using the hexagonal packing strategy with different values of hexagonal packing angles θ and radius of circular target area s.

Figures C.8–C.9 displays screenshots from experiments using the touch and run strategy with radius of circular target area $s = 6$ m and different values of the number of lanes K.

C.2 Graphs of the time of arrival per robots

Figure C.10 displays the relation between the time of arrival at the target region and the number of robots for the hexagonal packing strategy for $s \in \{3, 6\}$ and $\theta \in \{0, \pi/12, \pi/6, 5\pi/18\}$. They match the experiments of Figure 4.27 in Section 4.6.3.

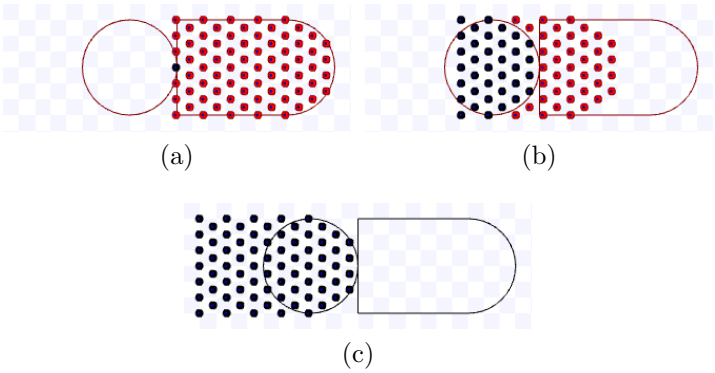
Figure C.11 exhibits the relation between the time of arrival at the target region and the number of robots for the touch and strategy for $s \in \{3, 6\}$ and different values of the number of lanes K. They are on a par with the experiments of Figure 4.30 in Section 4.6.4.



(a) (b)

(c)

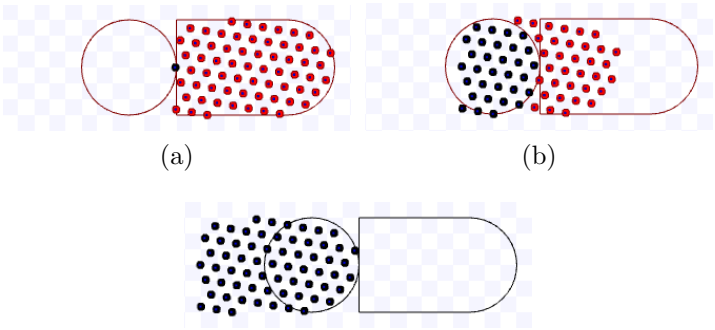
Figure C.1: Simulation on Stage for hexagonal packing strategy using $s = 3$ m, $\theta = \pi/12$ during $t = 10$ s. Available on `https://youtu.be/Wji8XlSQJBQ`, accessed on 12 June 2022. (a) 0 s: beginning of the simulation; (b) After 5 s; (c) 10 s: ending of the simulation.



(a) (b)

(c)

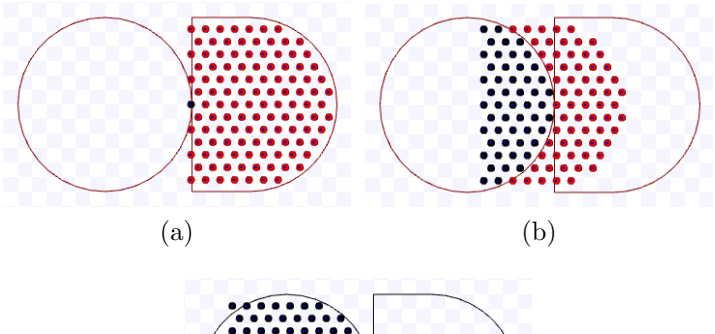
Figure C.2: Simulation on Stage for hexagonal packing strategy using $s = 3$ m, $\theta = \pi/6$ during $t = 10$ s. Available on `https://youtu.be/szOBU8no_sU`, accessed on 12 June 2022. (a) 0 s: beginning of the simulation; (b) After 4.9 s; (c) 10 s: ending of the simulation.



(a) (b)

(c)

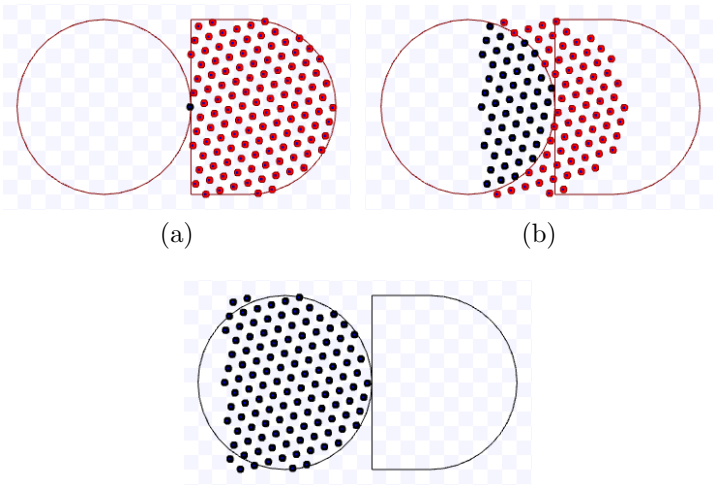
Figure C.3: Simulation on Stage for hexagonal packing strategy using $s = 3$ m, $\theta = 5\pi/18$ during $t = 10$ s. Available on `https://youtu.be/jRLgaF7Te1Q`, accessed on 12 June 2022. (a) 0 s: beginning of the simulation; (b) After 4.9 s; (c) 10 s: ending of the simulation.



(a) (b)

(c)

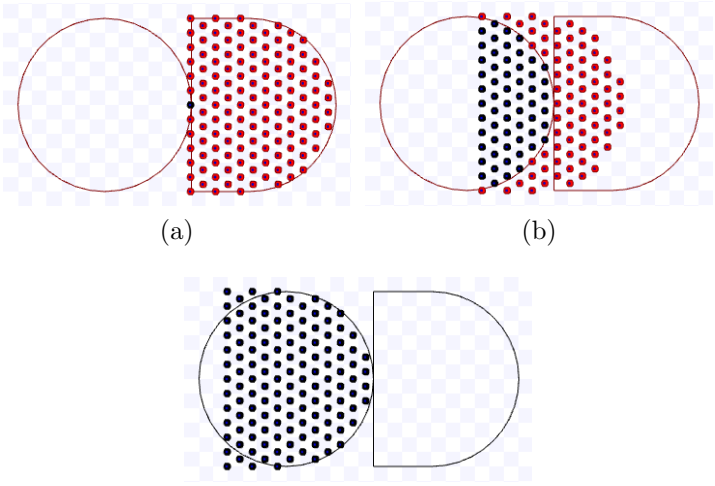
Figure C.4: Simulation on Stage for hexagonal packing strategy using $s = 6$ m, $\theta = 0$ during $t = 9.8$ s. Available on `https://youtu.be/v0FK8YpGrL8`, accessed on 12 June 2022. (a) 0 s: beginning of the simulation; (b) After 4.9 s; (c) 9.8 s: ending of the simulation.



(a) (b)

(c)

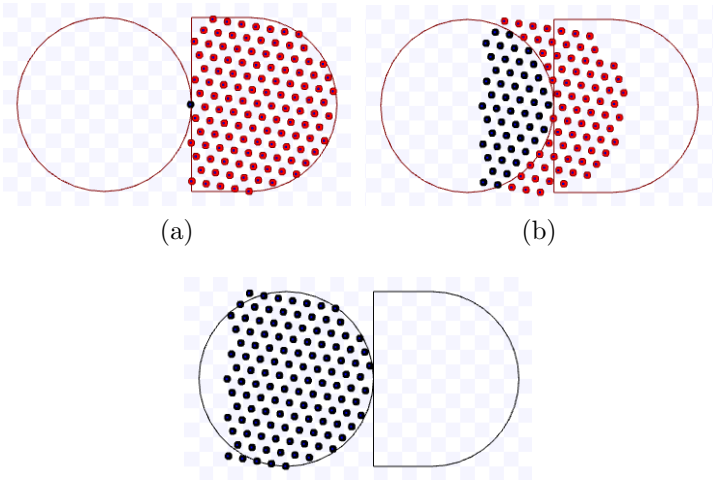
Figure C.5: Simulation on Stage for hexagonal packing strategy using $s = 6$ m, $\theta = \pi/12$ during $t = 10.1$ s. Available on `https://youtu.be/OBS_HADH5OE`, accessed on 12 June 2022. (a) 0 s: beginning of the simulation.; (b) After 5 s; (c) 10.1 s: ending of the simulation..



(a) (b)

(c)

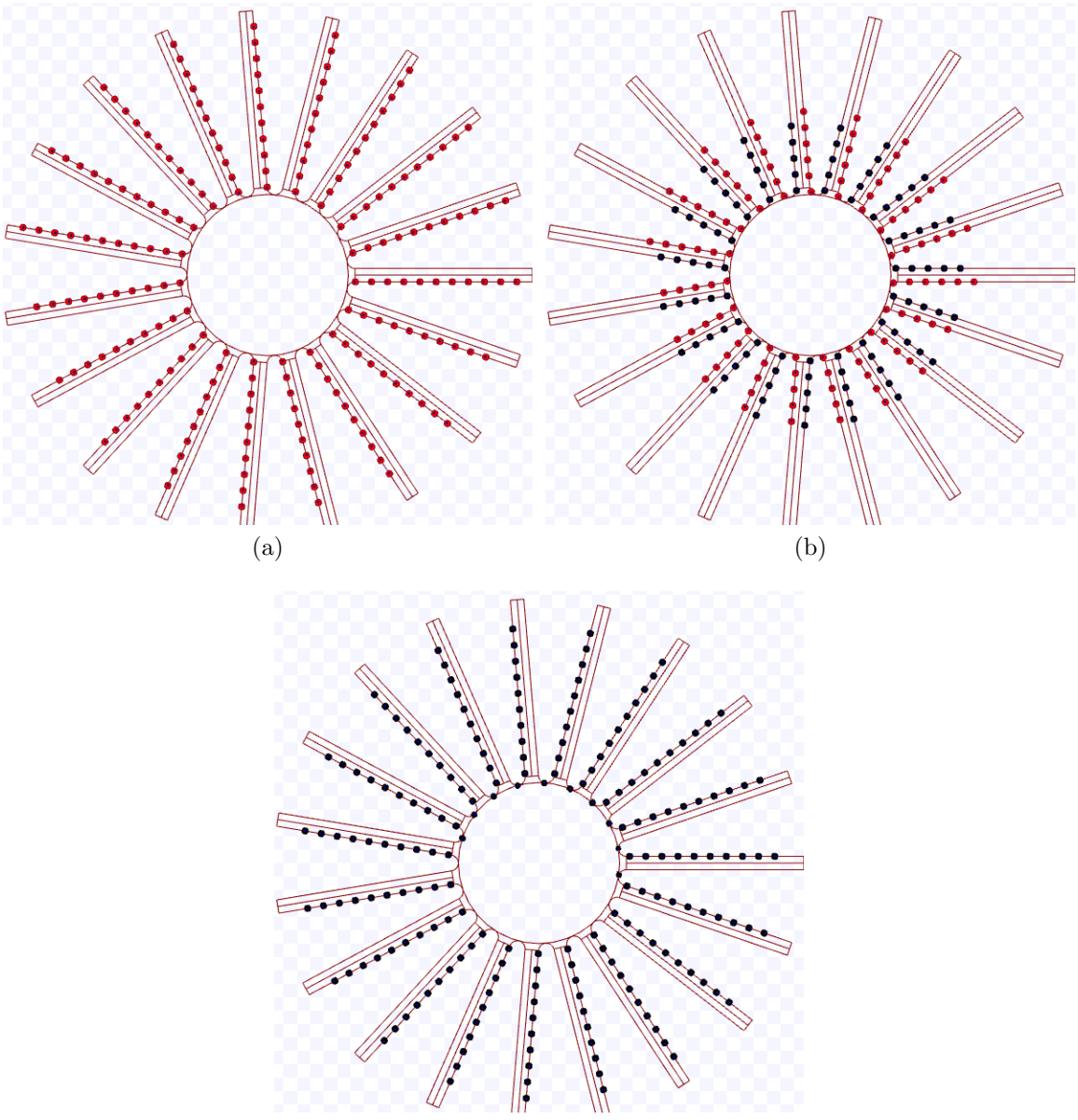
Figure C.6: Simulation on Stage for hexagonal packing strategy using $s = 6$ m, $\theta = \pi/6$ during $t = 10$ s. Available on `https://youtu.be/-KX7ziOp8b0`, accessed on 12 June 2022. (a) 0 s: beginning of the simulation.; (b) After 4.9 s; (c) 10 s: ending of the simulation.



(a) (b)

(c)

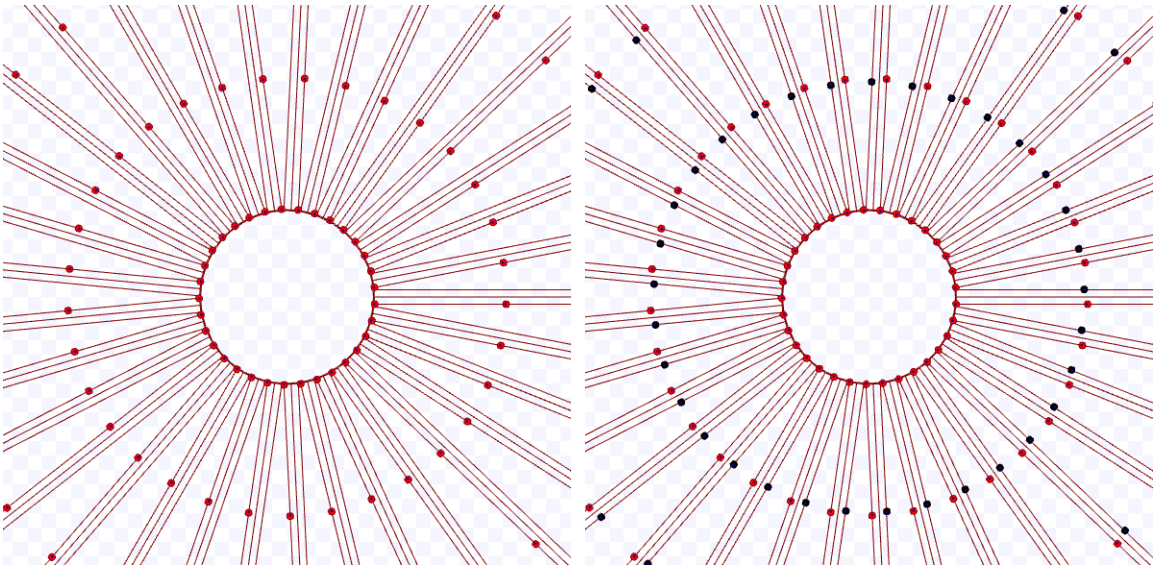
Figure C.7: Simulation on Stage for hexagonal packing strategy using $s = 6$ m, $\theta = 5\pi/18$ during $t = 10$ s. Available on `https://youtu.be/GRYRnH5CrhU`, accessed on 12 June 2022. (a) 0 s: beginning of the simulation; (b) After 4.9 s; (c) 10 s: ending of the simulation.



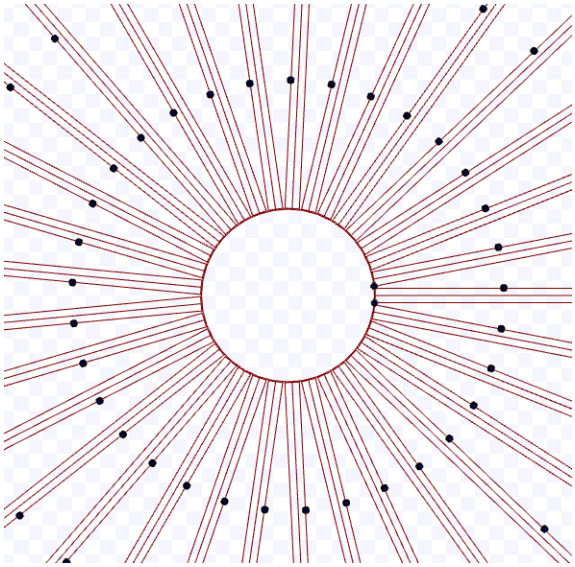
(a) (b)

(c)

Figure C.8: Simulation on Stage for the touch and run strategy using $s = 6$ m, $K = 19$ during $t = 127.4$ s at $v = 0.1$ m/s. Available on `https://youtu.be/xJVoVCIjX5k`, accessed on 12 June 2022. (a) 0 s: beginning of the simulation; (b) After 63.6 s; (c) 127.4 s: ending of the simulation.



(a) (b)



(c)

Figure C.9: Simulation on Stage for the touch and run strategy using $s = 6$ m, $K = 33$ during $t = 548$ s at $v = 0.1$ m/s. Available on `https://youtu.be/-xZz84npKV4`, accessed on 12 June 2022. (a) 0 s: beginning of the simulation; (b) After 274 s; (c) 548 s: ending of the simulation.

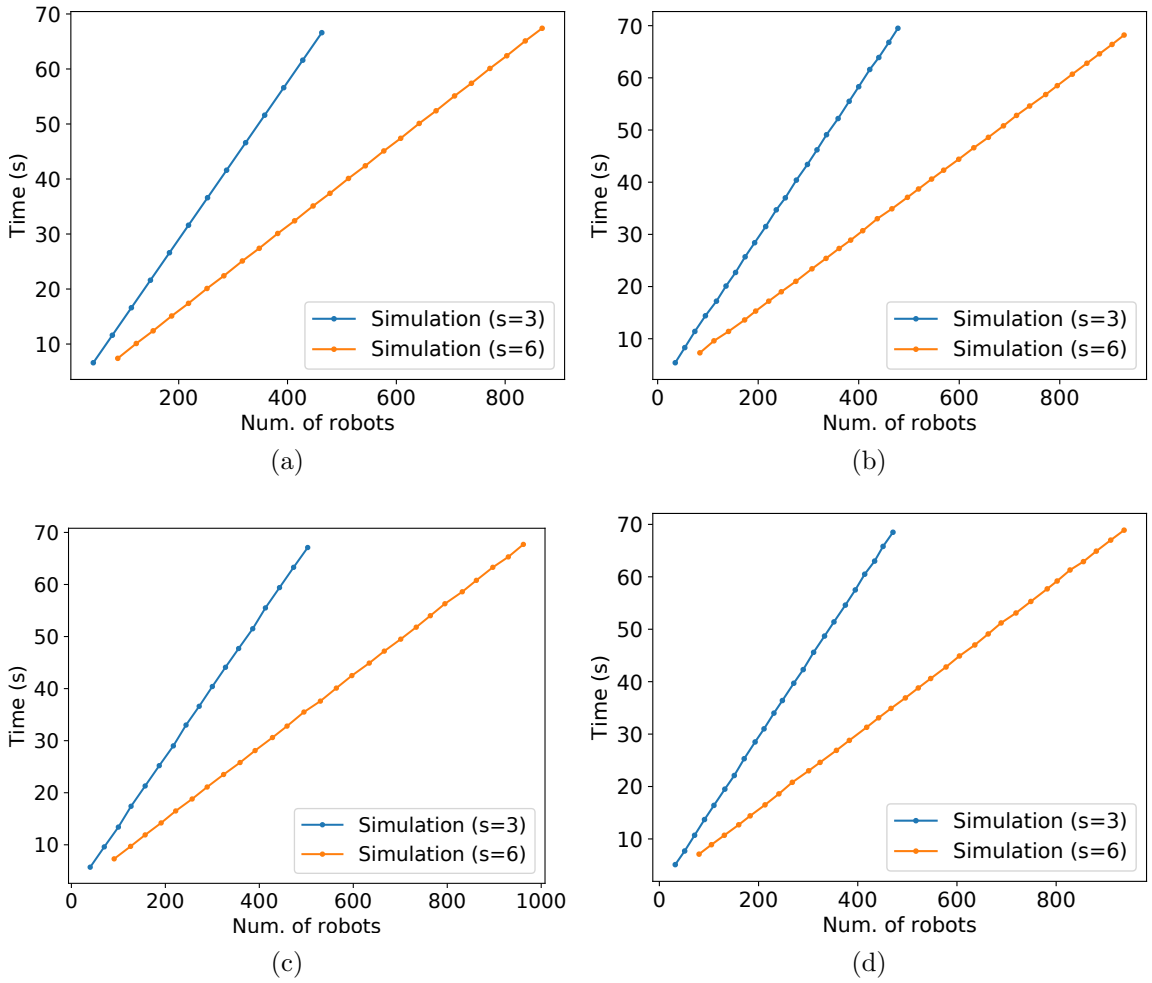
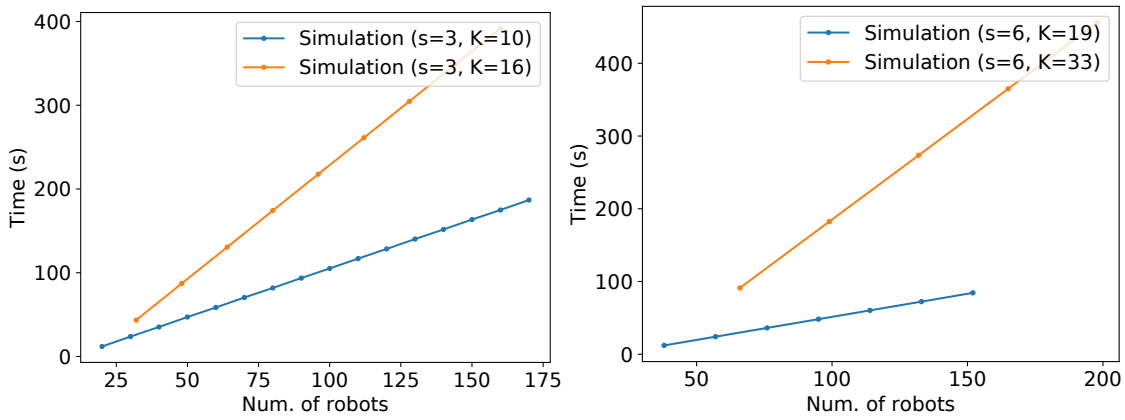


Figure C.10: Time of arrival at the target of the last robot versus the number of robots for the same simulations in Figure 4.27. (a) $\theta = 0$; (b) $\theta = \pi/12$; (c) $\theta = \pi/6$; (d) $\theta = 5\pi/18$.



(a) (b)

Figure C.11: Time of arrival at the target of the last robot versus number of robots for the same simulations in Figure 4.30. (a) $s = 3$ m and $K \in \{10, 16\}$; (b) $s = 6$ m and $K \in \{19, 33\}$.

Appendix D

Additional graphs for Chapter 5

D.1 Parameter K of the TRVF algorithm

Simulations were performed to compare different K values for the defaults parameters for a number of robots ranging from 20 to 300 with increments of 20 with the default values in Table 5.1. The results are in Figure D.1.

Firstly, the results are compared with the touch and run strategy by calculating the asymptotic throughput for the allowed values of K following (4.19) – that is, $K \in \{3, \ldots, 6\}$ – the asymptotic throughput rounded to three decimal places are 0.994, 1.2, 1.099 and 1, respectively. Thus, $K = 4$ would be the best number of lanes for the touch and run strategy. However, as in the TRVF experiments, the distances between robots and linear speeds change over time, and the movement of the robots is influenced by the other robots in their neighbourhood. Therefore, the best K was 5 according to the results in Figure D.1.

Figure D.2 shows a screenshot in the middle of the execution for the allowed K values and 100 holonomic robots. Using $K = 6$, the default values $s = 3$ m and $d = I_d = 3$ m in (4.13) gives $r = 0$, so the TRVF algorithm will run but no curved trajectory is made. In this case, the entrance and exiting straight lanes lie exactly in the target region. This could be a shorter trajectory, as the robots do not have the curved turn, but congestion happens in the target region as they have to turn to leave the target area in a sharp curve (Figure D.2d). As K grows, the curve tends to be sharper. Obligating robots to do sharp curves tends to increase congestion due to their slowdown. Thus, the best K results from the equilibrium between the rise of the number of target accesses and the minimisation of congestion due to slowdowns near the target caused by diminishing the turning radius of the curved path. Observe that, for any K displayed in Figure D.2, some robots are trapped inside the target area. They were pushed inside the area due to repulsive forces from other robots, and they inflicted those forces on the robots getting near the target region from the curved

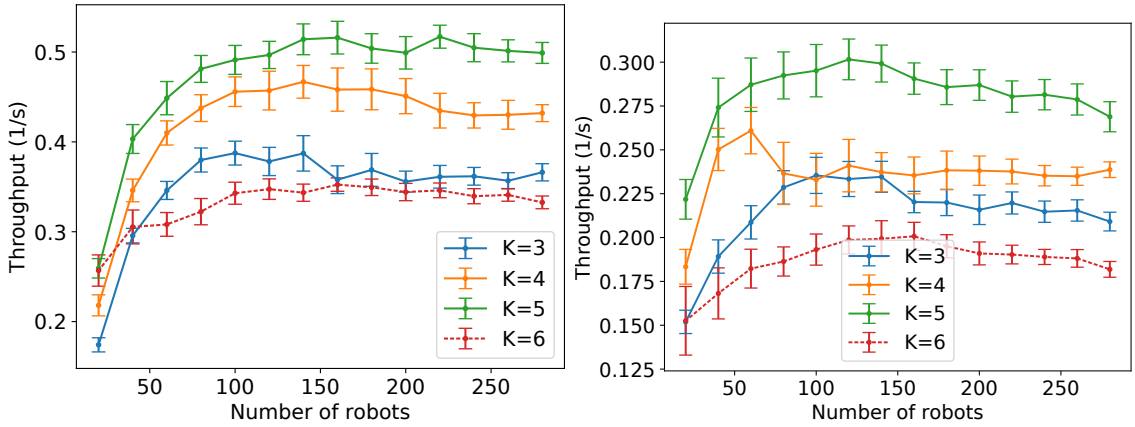


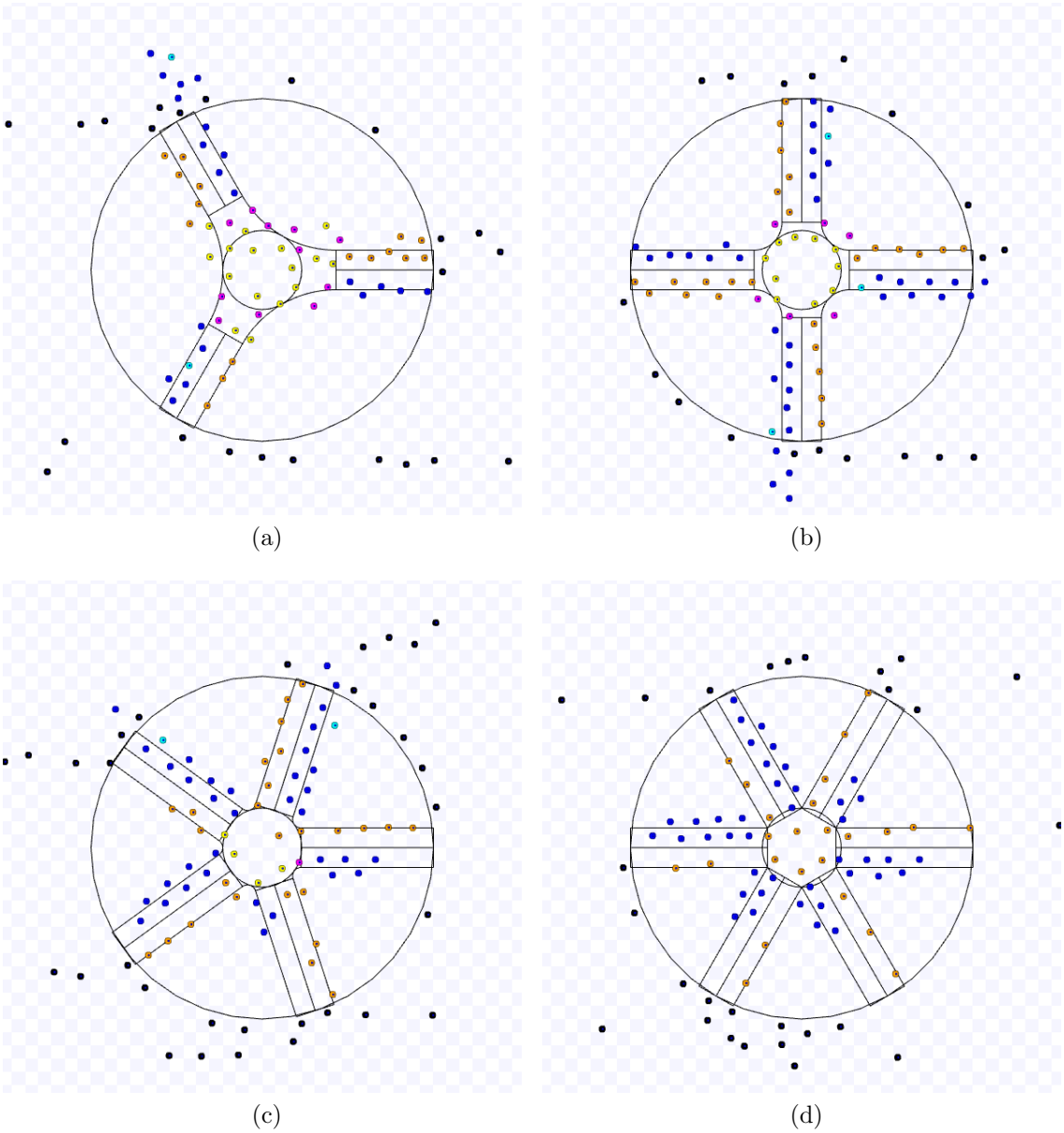
Figure D.1: Throughput of the TRVF algorithm by the number of robots from 20 to 300 in steps of 20 for different values of the number of lanes (K). (a) Holonomic. (b) Non-holonomic.

trajectory, consequently causing congestion as well. However, for the best $K = 5$, congestion is minimised.

D.2 Parameter *Alg* of MT

Figures D.3–D.6 illustrates the comparison of the total simulation time per total number of robots of a MT using SQF and NC as parameter *Alg* with percentages of robots in group A to the total number of individuals ranging from 20% to 90% in steps of 10%. For 30% to 70% for non-holonomic robots (Figures D.3–D.5), NC as the alternative algorithm is significantly better on average until different values from 120 to 180 robots. Note that the number of robots where it starts to take less time decreases with the ratio of ad hoc robots.

Figures D.7–D.10 show the comparison of NC and TRVF as the group A algorithms for ratios varying from 20% to 90% in steps of 10% for holonomic and non-holonomic robots. For 90%, TRVF is significantly better only from 200 to 300 in the holonomic case and from 180 to 300 robots in the non-holonomic cases, respectively (Figure D.10). When using the TRVF as an alternative algorithm, the lanes near the leaving trajectory of the SQF algorithm create congestion. However, with NC, they tend to go directly to the target, opening space near the SQF leaving area.



(a) (b)

(c) (d)

Figure D.2: Screenshots of the TRVF algorithm, for 100 holonomic robots, default values and different numbers of lanes (K) at the middle of the execution. Available on `https://youtu.be/BYd8VncXnCQ`, accessed on 20 December 2022. (a) $K = 3$ and after 202.8 s. Available on `https://youtu.be/0U6ajiBtYw8`, accessed on 20 December 2022. (b) $K = 4$ and after 155.5 s. Available on `https://youtu.be/D-_2VK5JRYg`. (c) $K = 5$ and after 156.1 s. Available on `https://youtu.be/z92SNJ8ugHs`, accessed on 20 December 2022. (d) $K = 6$ and after 177.3 s.

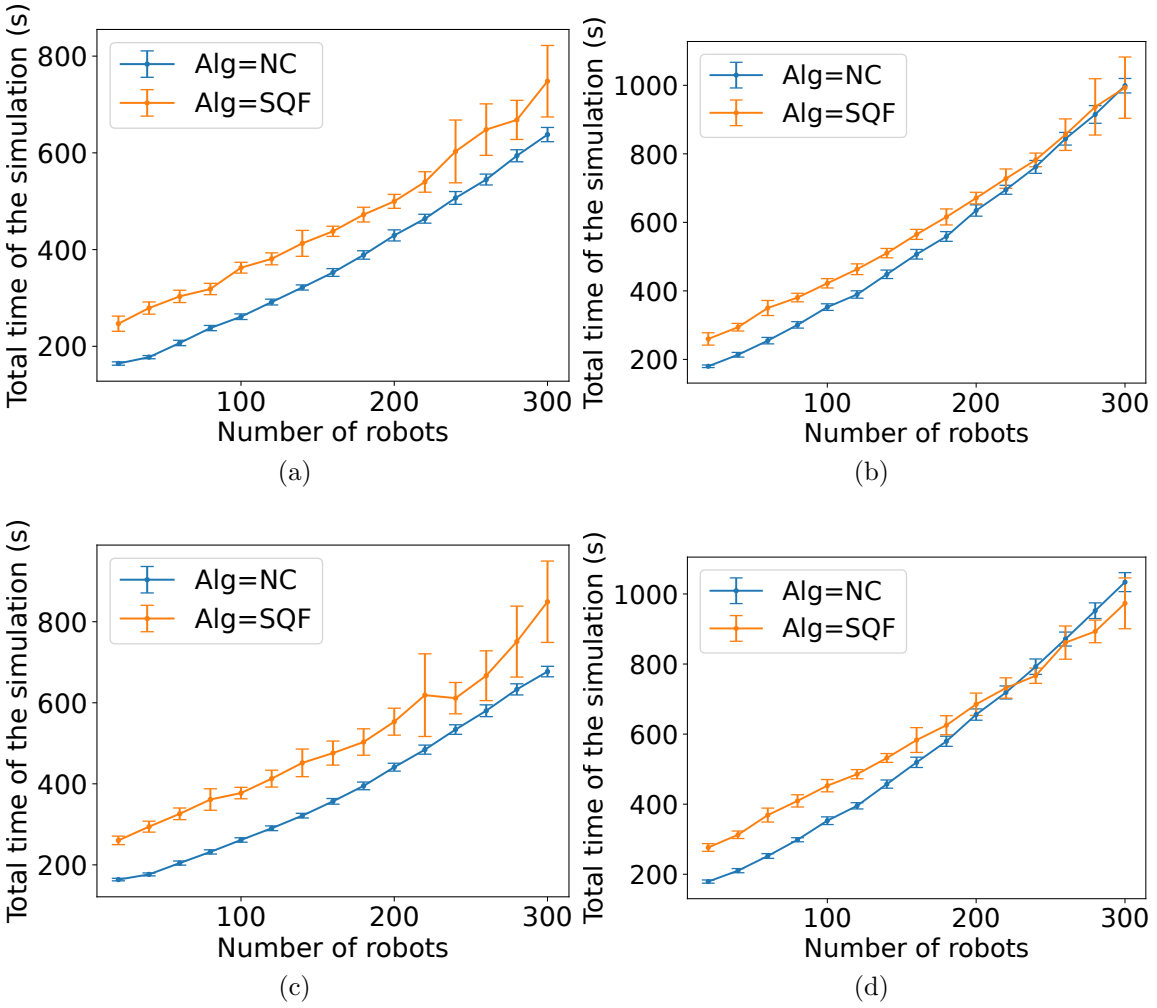


Figure D.3: Comparison of the total simulation time by the number of robots for the experiments with MT using TRVF as the swarm control algorithm and SQF and NC as the alternative algorithm by 20% of the robots for (a) holonomic and (b) non-holonomic robots and 30% for (c) holonomic and (d) non-holonomic robots.

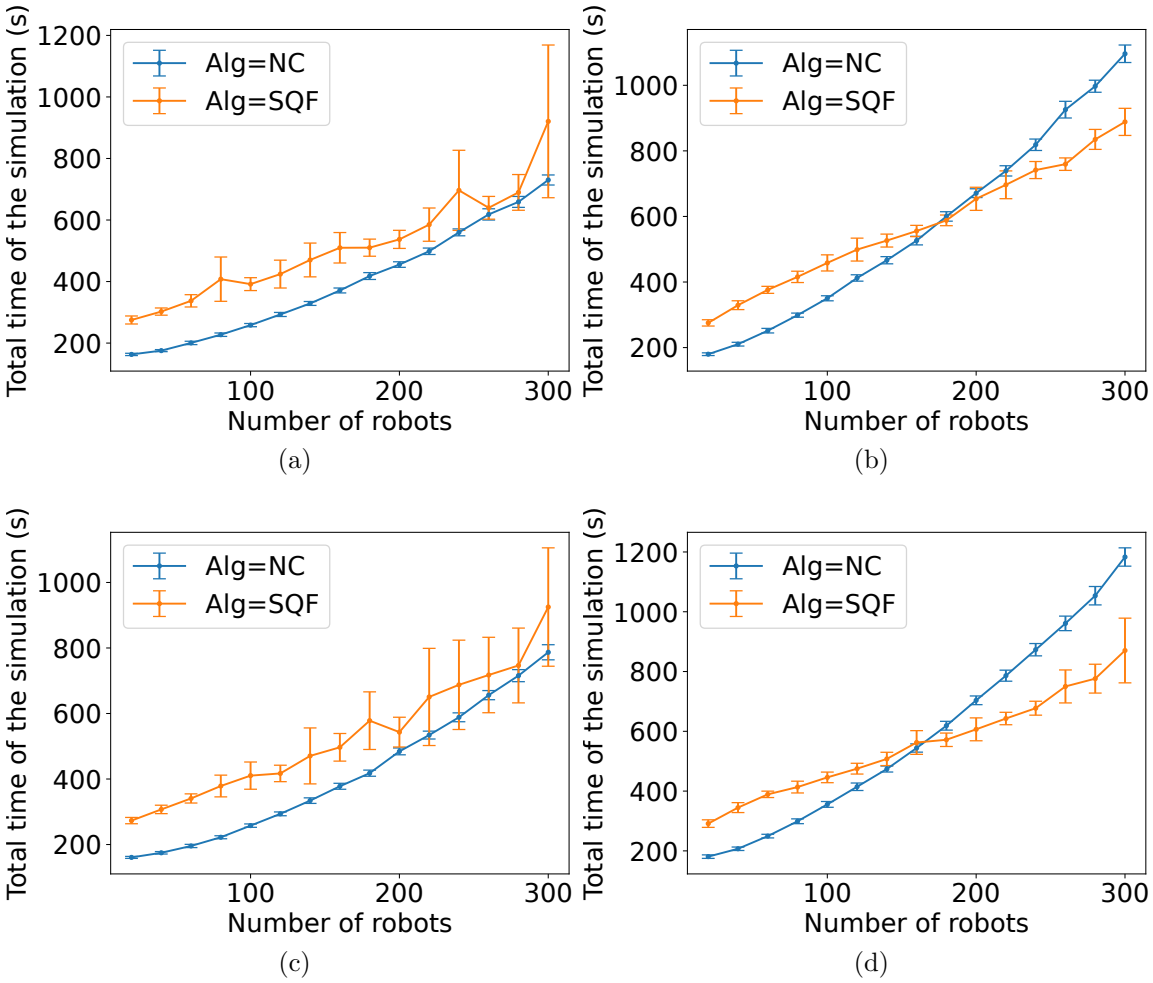


Figure D.4: Comparison of the total simulation time by the number of robots for the experiments with MT using TRVF as the swarm control algorithm and SQF and NC as the alternative algorithm by 40% of the robots for (a) holonomic and (b) non-holonomic robots and 50% for (c) holonomic and (d) non-holonomic robots.

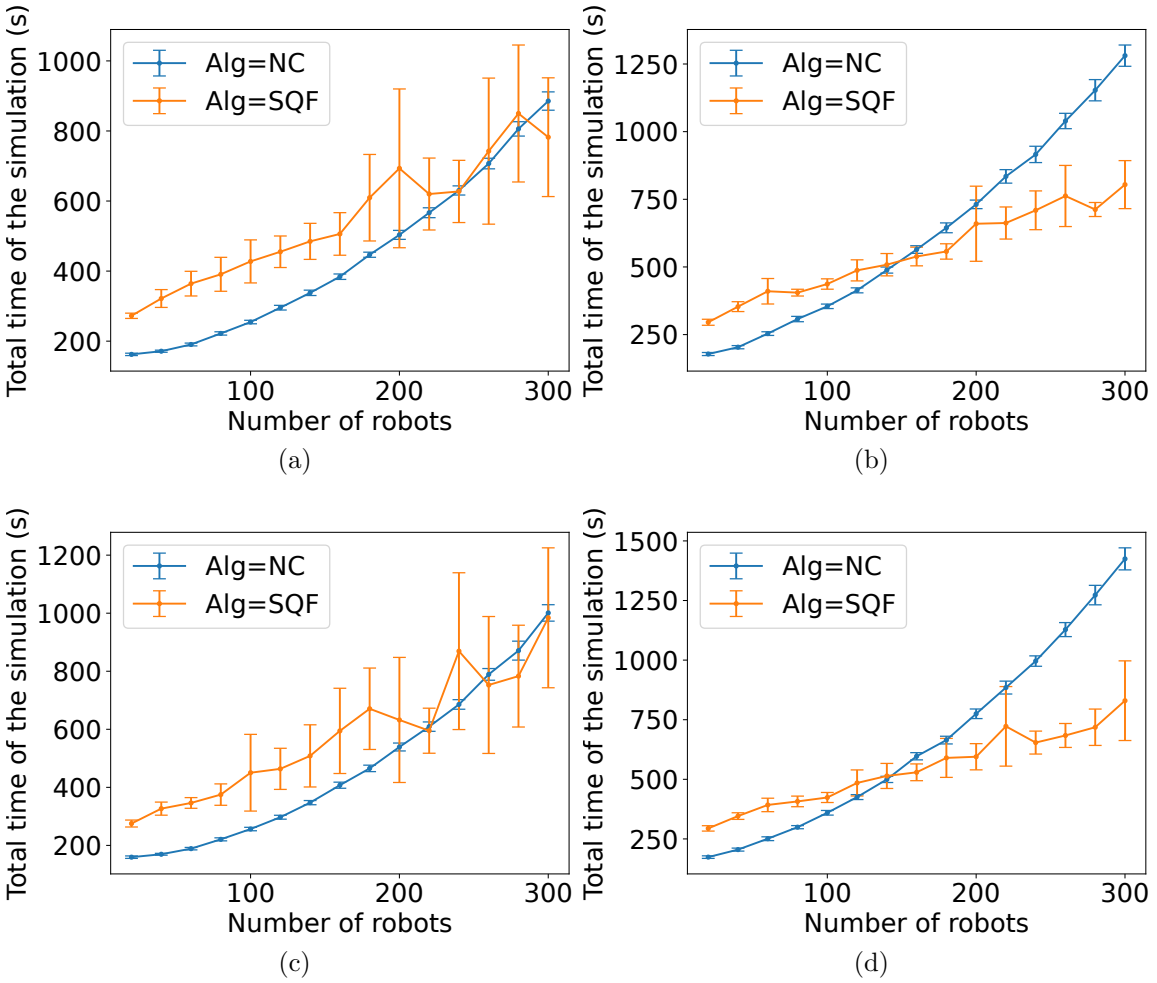


Figure D.5: Comparison of the total simulation time by the number of robots for the experiments with MT using TRVF as the swarm control algorithm and SQF and NC as the alternative algorithm by 60% of the robots for (a) holonomic and (b) non-holonomic robots and 70% for (c) holonomic and (d) non-holonomic robots.

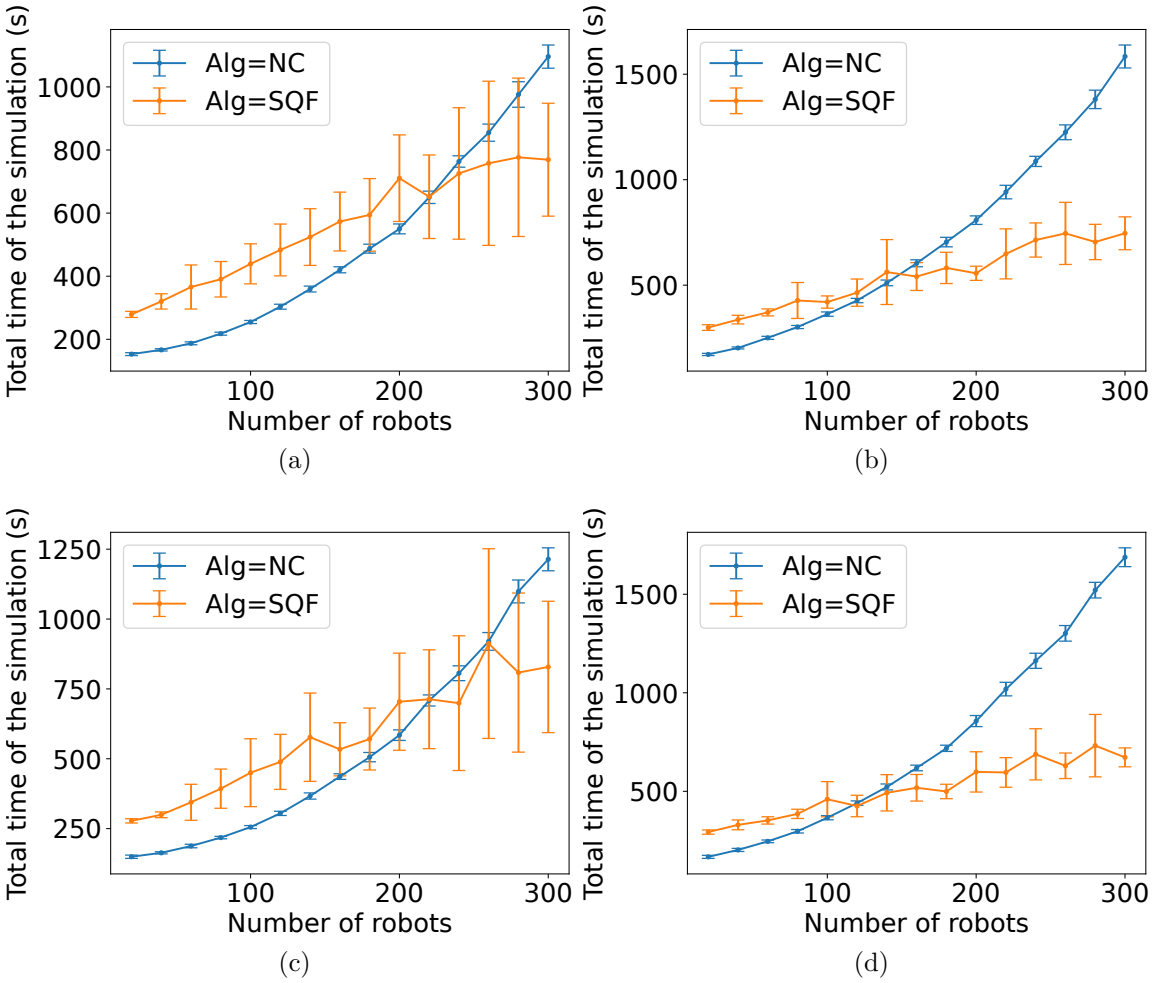


Figure D.6: Comparison of the total simulation time by the number of robots for the experiments with MT using TRVF as the swarm control algorithm and SQF and NC as the alternative algorithm by 80% of the robots for (a) holonomic and (b) non-holonomic robots and 90% for (c) holonomic and (d) non-holonomic robots.

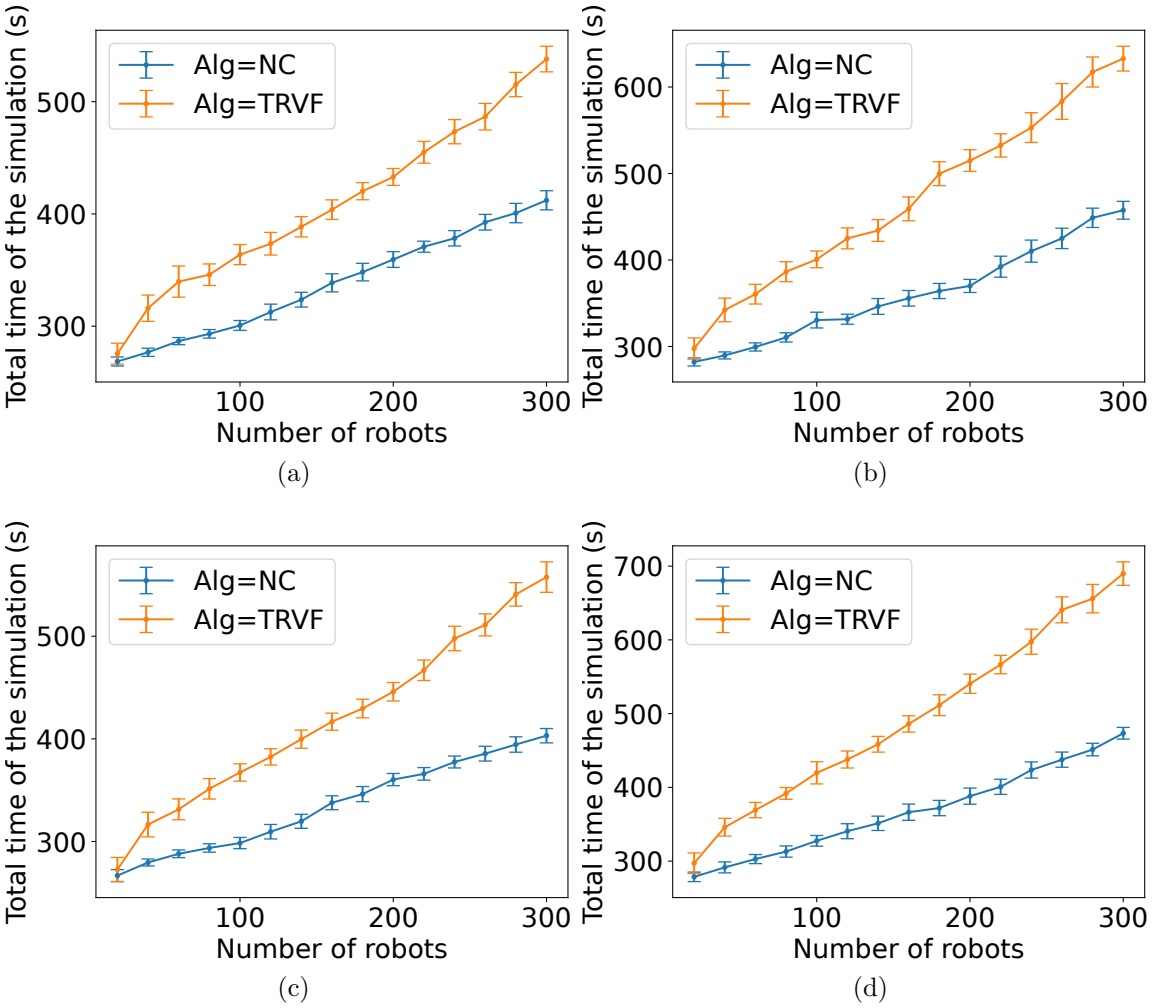


Figure D.7: Comparison of the total simulation time by the number of robots for the experiments with MT using SQF as the swarm control algorithm and TRVF and NC as the alternative algorithm by 20% of the robots for (a) holonomic and (b) non-holonomic robots and 30% for (c) holonomic and (d) non-holonomic robots.

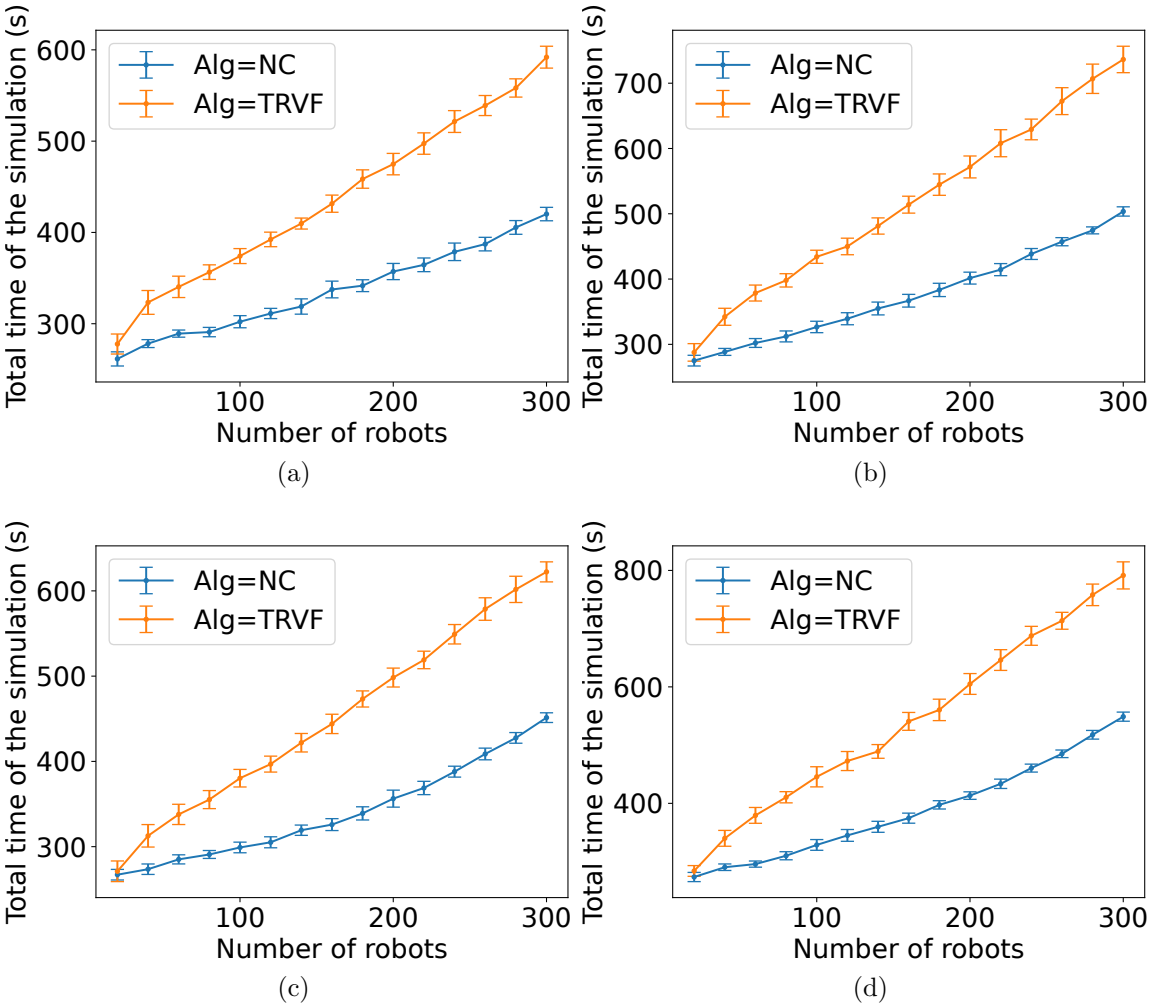


Figure D.8: Comparison of the total simulation time by the number of robots for the experiments with MT using SQF as the swarm control algorithm and TRVF and NC as the alternative algorithm by 40% of the robots for (a) holonomic and (b) non-holonomic robots and 50% for (c) holonomic and (d) non-holonomic robots.

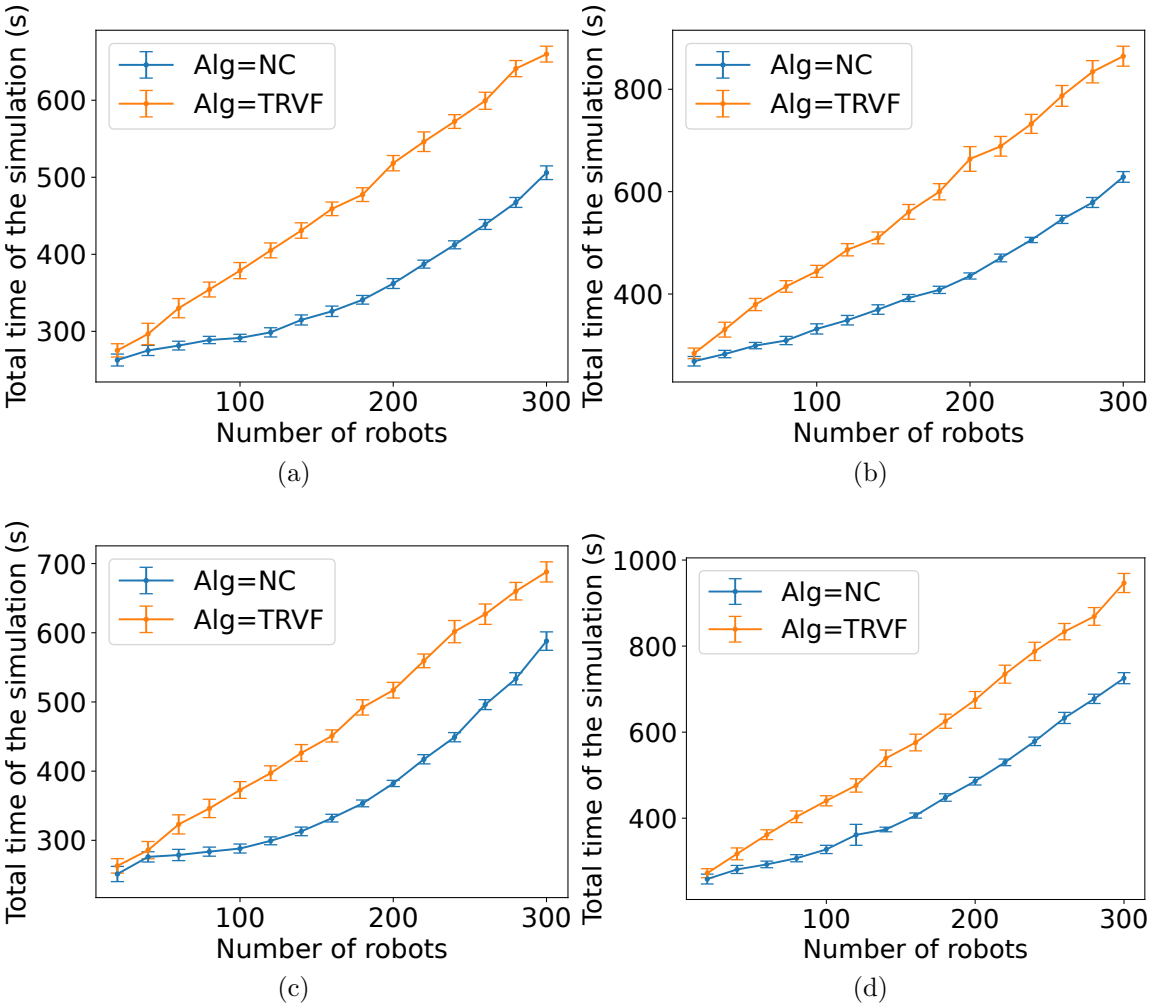


Figure D.9: Comparison of the total simulation time by the number of robots for the experiments with MT using SQF as the swarm control algorithm and TRVF and NC as the alternative algorithm by 60% of the robots for (a) holonomic and (b) non-holonomic robots and 70% for (c) holonomic and (d) non-holonomic robots.

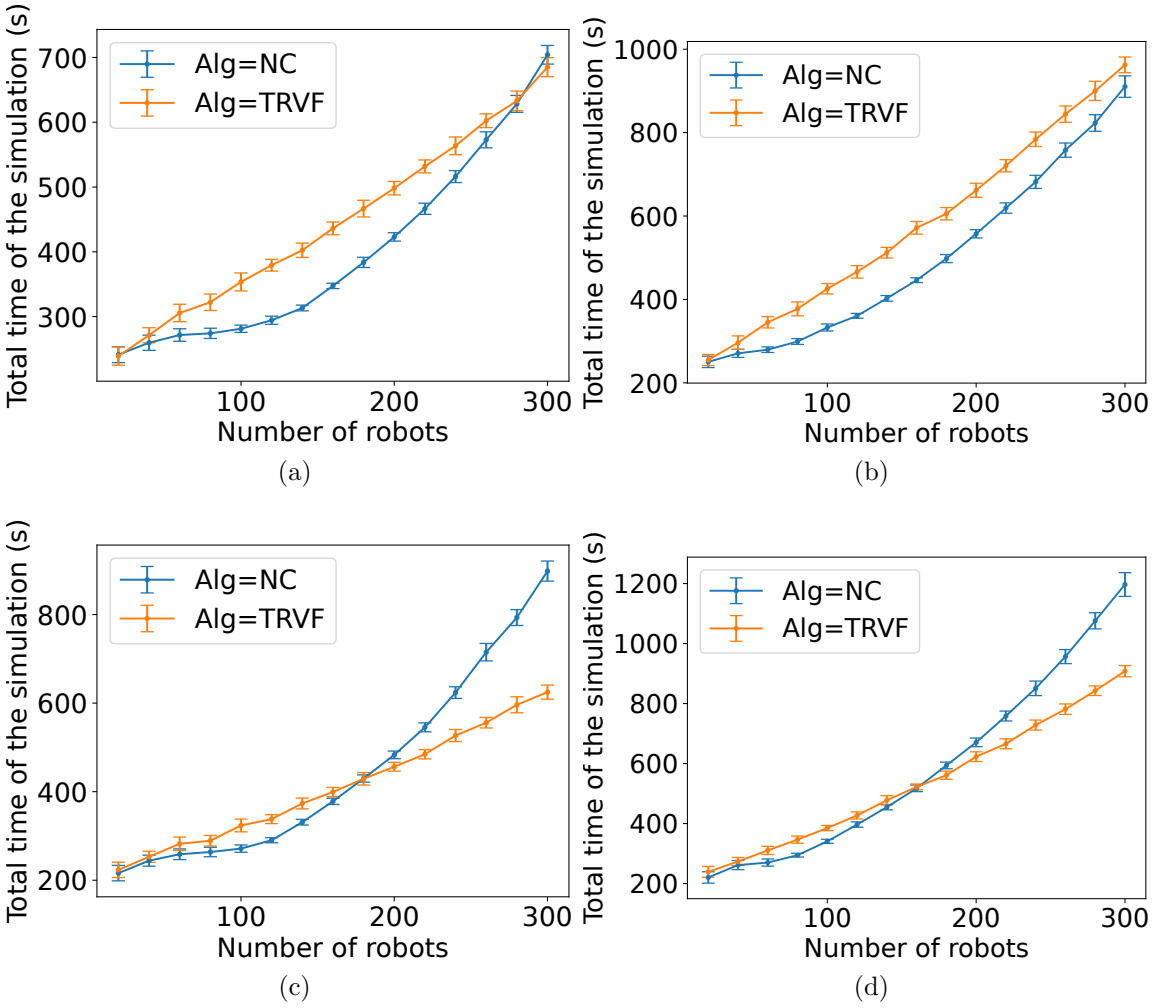


Figure D.10: Comparison of the total simulation time by the number of robots for the experiments with MT using SQF as the swarm control algorithm and TRVF and NC as the alternative algorithm by 80% of the robots for (a) holonomic and (b) non-holonomic robots and 90% for (c) holonomic and (d) non-holonomic robots.

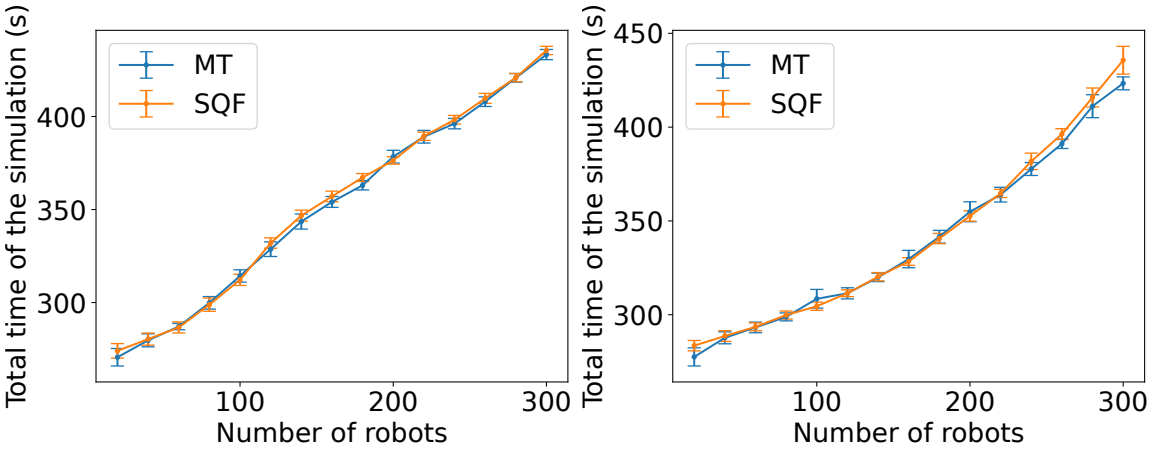


Figure D.11: Comparison of the total simulation time by the number of robots for the experiments with one robot using NC and SQF being used by the others for (a) holonomic and (b) non-holonomic robots.

D.3 Tests with one Ad Hoc Robot in MT.

When assuming only one ad hoc robot, the effect of its presence does not have a big impact in the total simulation time. Figures D.11 and D.12 show the simulation time per number of robots for only one robot in MT with NC and the aware robots using SQF and TRVF for holonomic and non-holonomic cases. For reference, results for the algorithms without any ad hoc robot are presented. Observe that for each number of robots, the mean values for SQF and TRVF with holonomic robots are not significantly different except for 180 robots for the SQF algorithm (Figure D.11a). For the non-holonomic case, the mean values are significantly different for 20, 260 and 300 robots for the SQF algorithm (Figure D.11b) and 40 robots for the TRVF algorithm (Figure D.12b). Despite that, the difference in their means is low.

D.4 Different Ratios of Ad Hoc Robots

Figures D.13–D.20 show the simulation time per number of robots for holonomic and non-holonomic robots from 20% to 90% in steps of 10% of ad hoc robots to the total number of individuals. SQF is the congestion control algorithm in Figures D.13–D.16 and TRVF in Figures D.17–D.20.

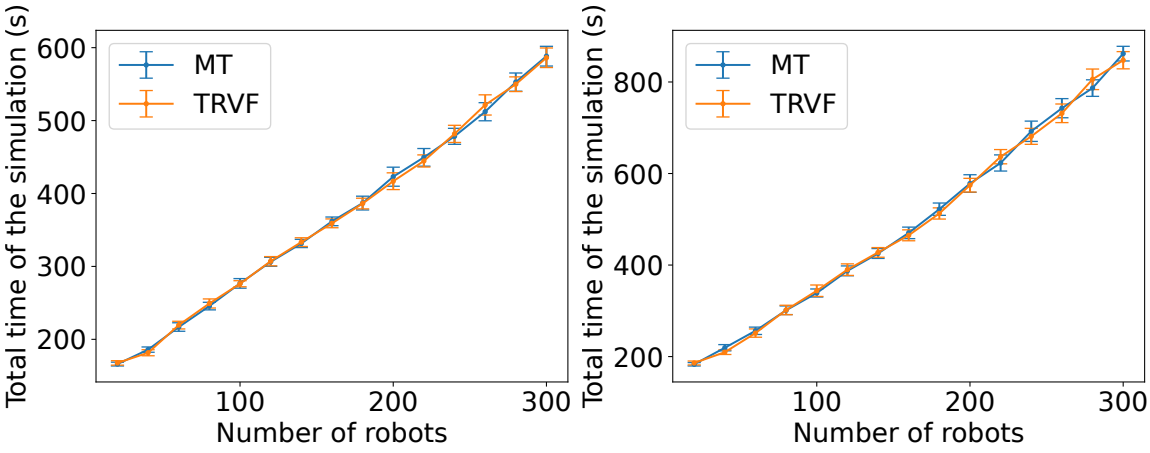


Figure D.12: Comparison of the total simulation time by the number of robots for the experiments with one robot using NC and TRVF being used by the others for (a) holonomic and (b) non-holonomic robots.

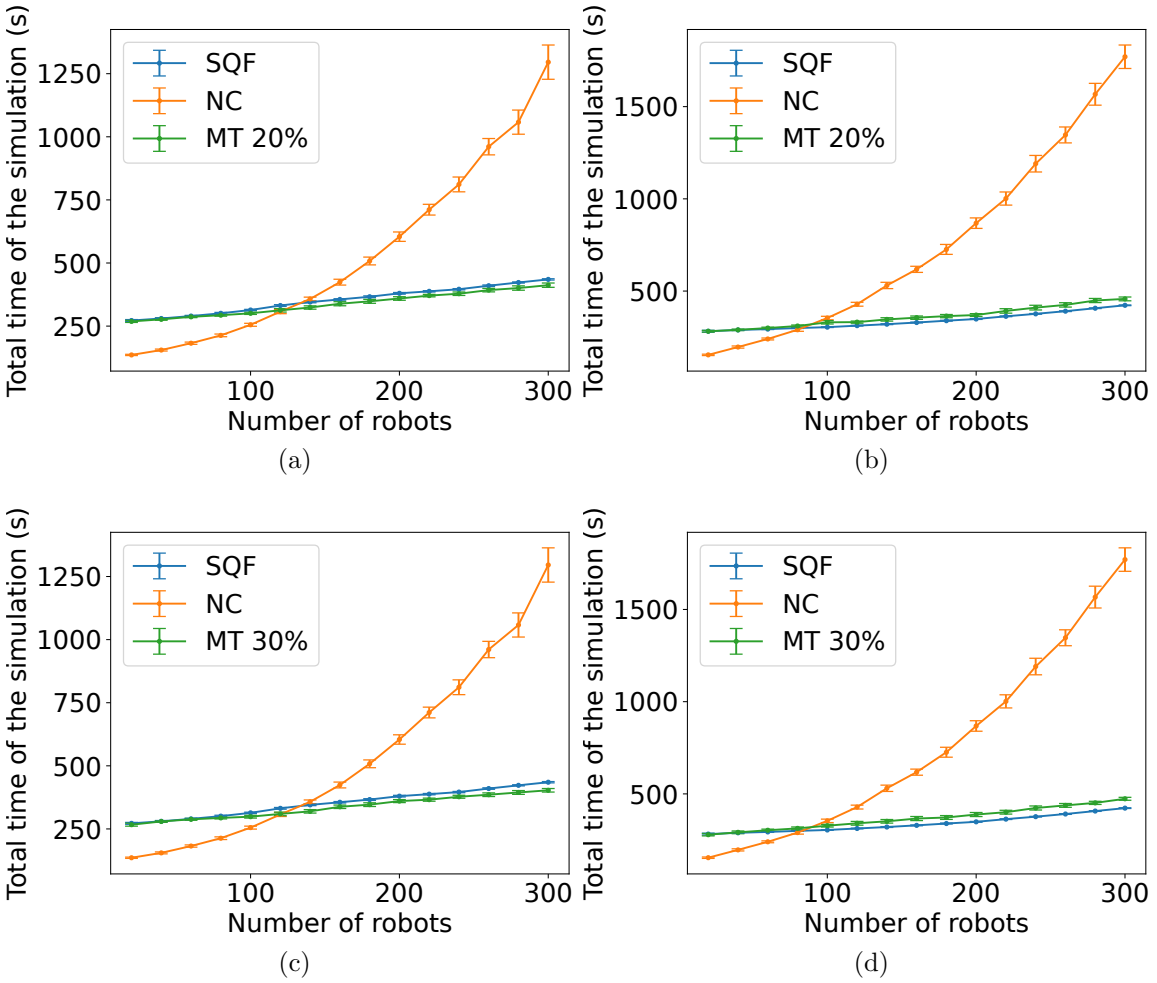


Figure D.13: Comparison of the total simulation time by the number of robots for the experiments with MT using SQF as the swarm control algorithm and NC as the alternative algorithm by 20% of the robots for (a) holonomic and (b) non-holonomic robots and 30% for (c) holonomic and (d) non-holonomic robots.

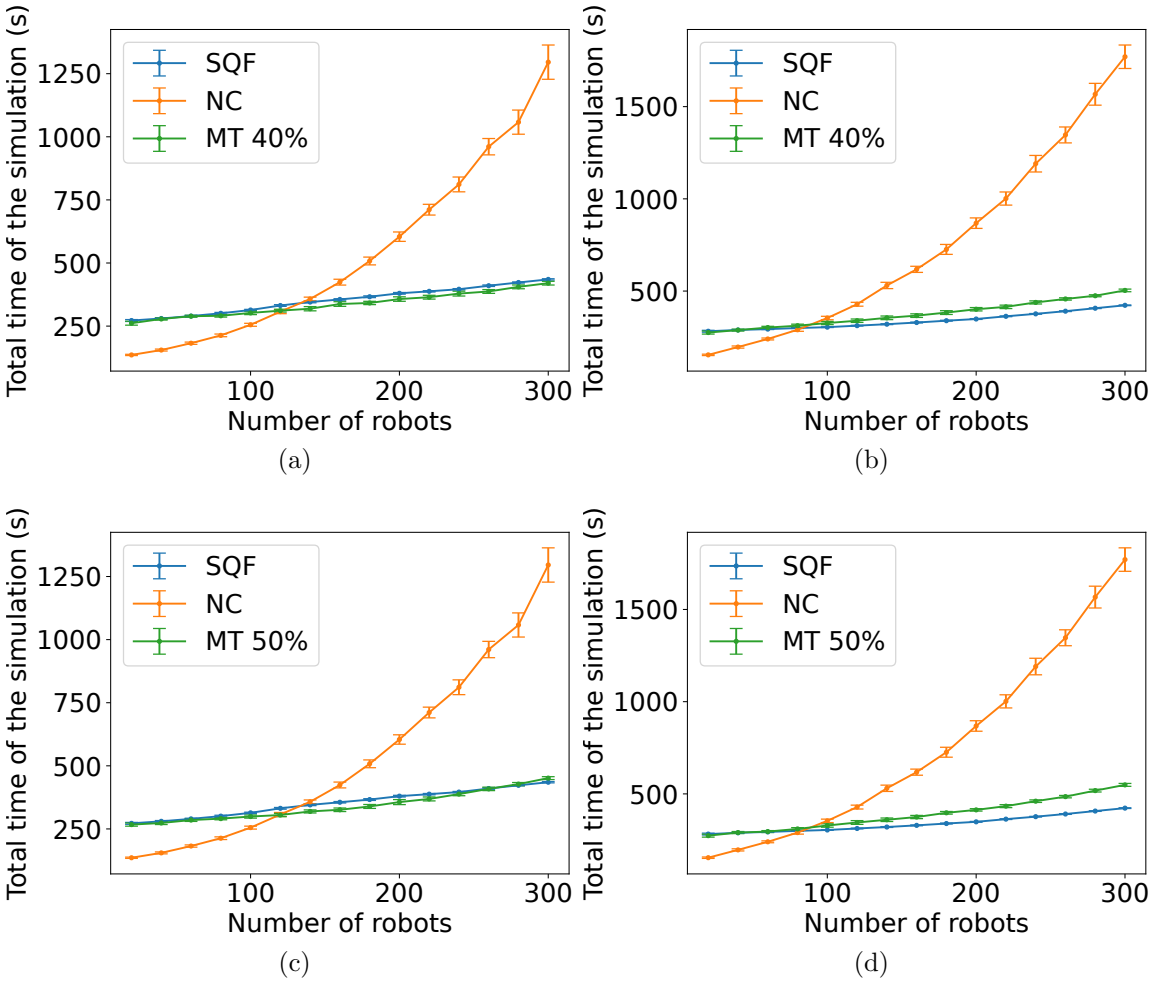


Figure D.14: Comparison of the total simulation time by the number of robots for the experiments with MT using SQF as the swarm control algorithm and NC as the alternative algorithm by 40% of the robots for (a) holonomic and (b) non-holonomic robots and 50% for (c) holonomic and (d) non-holonomic robots.

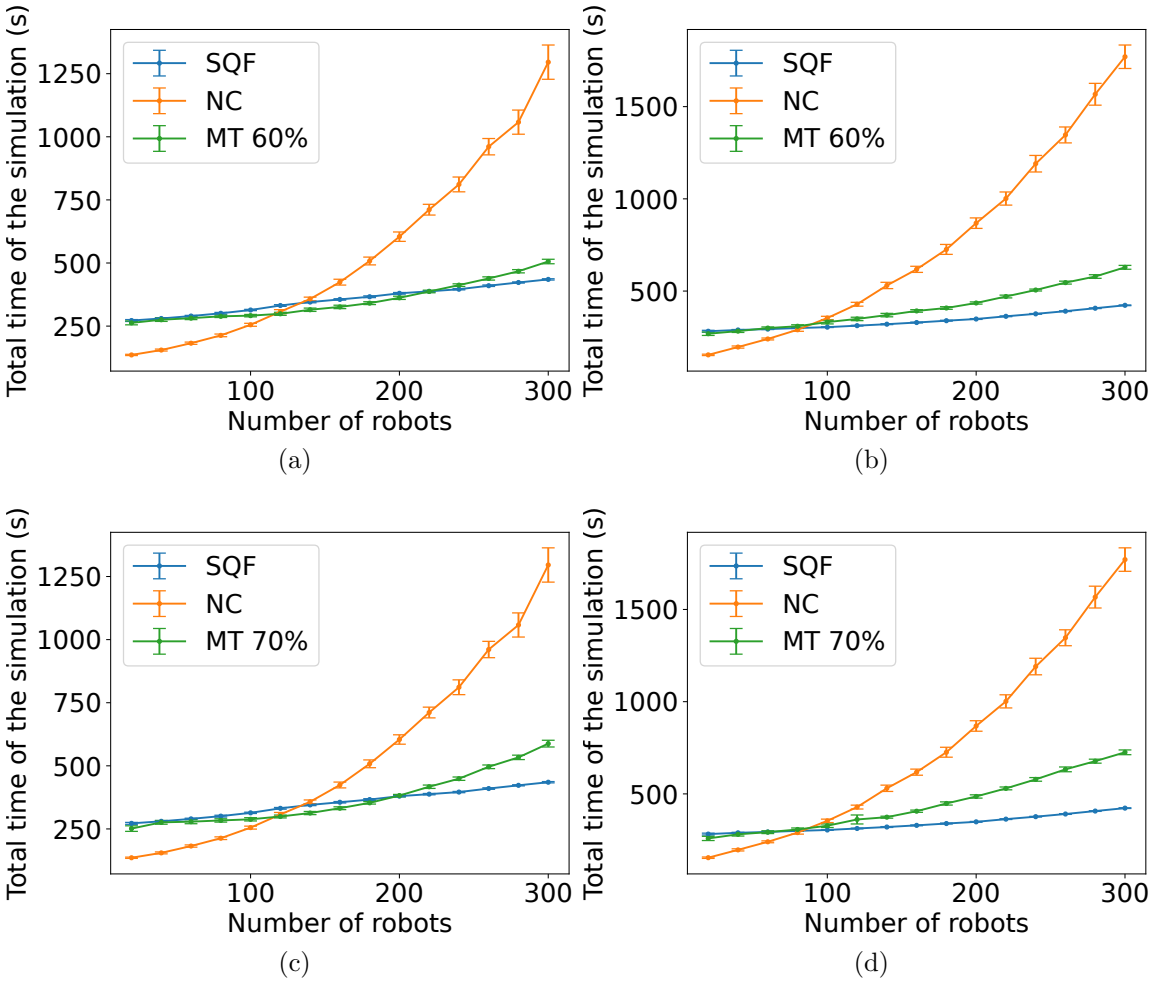


Figure D.15: Comparison of the total simulation time by the number of robots for the experiments with MT using SQF as the swarm control algorithm and NC as the alternative algorithm by 60% of the robots for (a) holonomic and (b) non-holonomic robots and 70% for (c) holonomic and (d) non-holonomic robots.

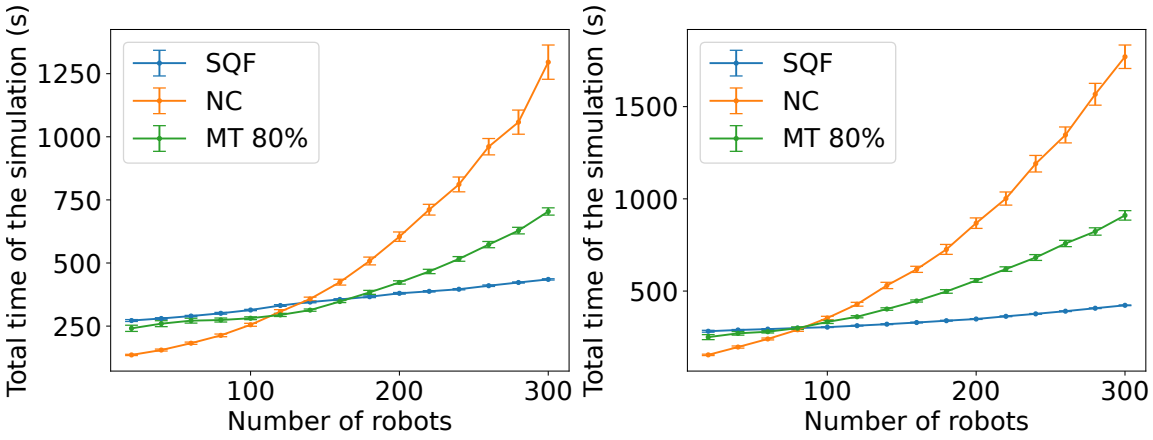


Figure D.16: Comparison of the total simulation time by the number of robots for the experiments with MT using SQF as the swarm control algorithm and NC as the alternative algorithm by 80% of the robots for (a) holonomic and (b) non-holonomic robots and 90% for (c) holonomic and (d) non-holonomic robots.

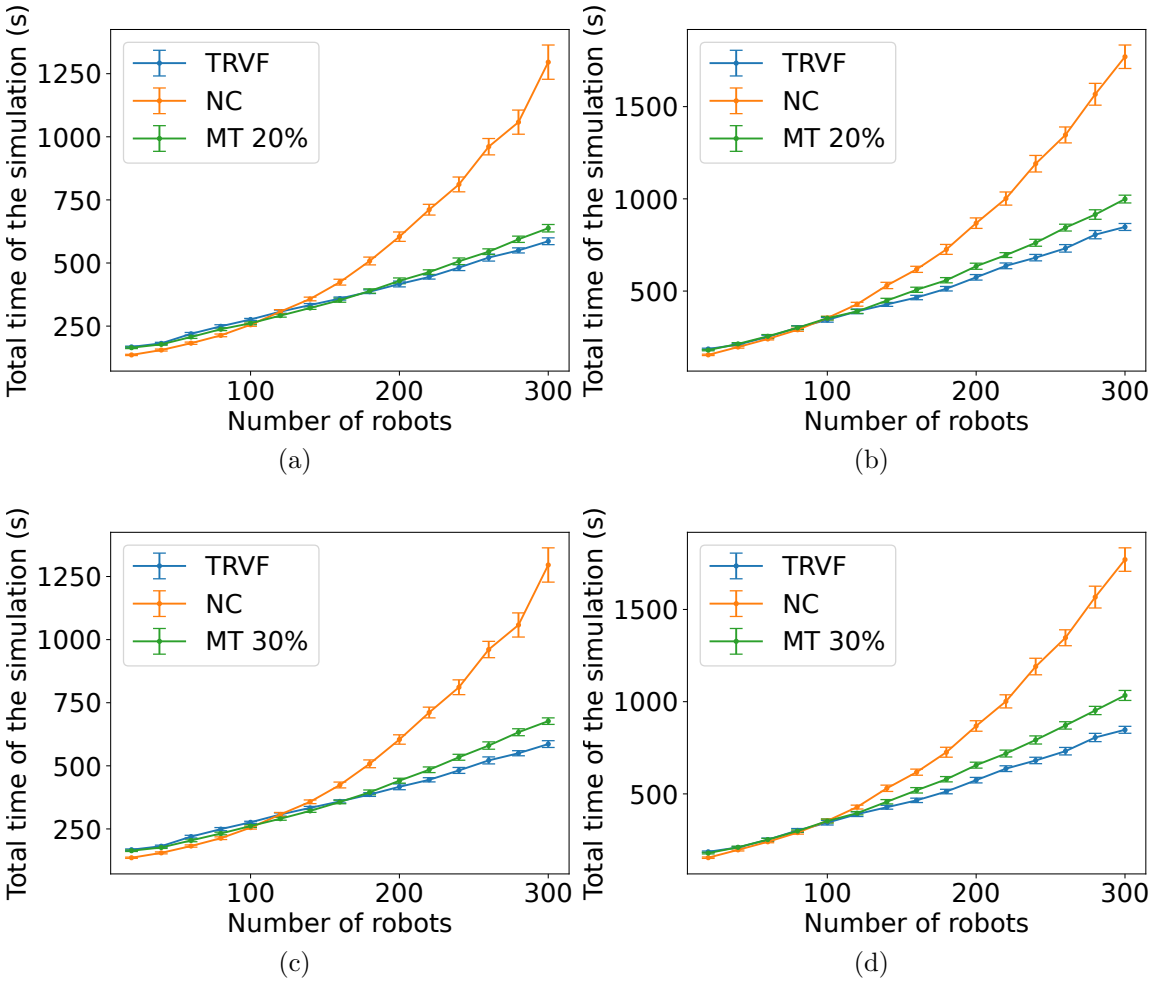


Figure D.17: Comparison of the total simulation time by the number of robots for the experiments with MT using TRVF as the swarm control algorithm and NC as the alternative algorithm by 20% of the robots for (a) holonomic and (b) non-holonomic robots and 30% for (c) holonomic and (d) non-holonomic robots.

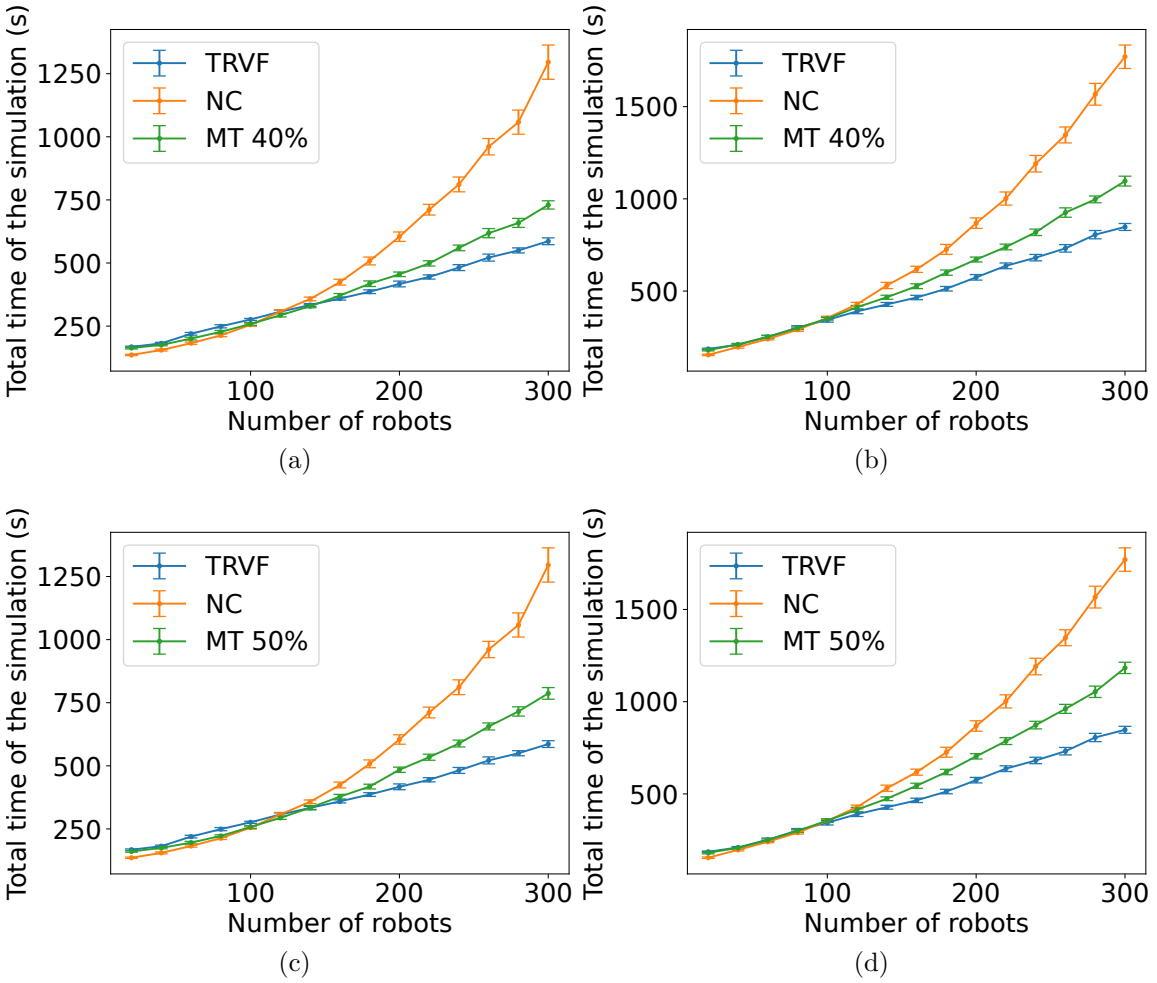


Figure D.18: Comparison of the total simulation time by the number of robots for the experiments with MT using TRVF as the swarm control algorithm and NC as the alternative algorithm by 40% of the robots for (a) holonomic and (b) non-holonomic robots and 50% for (c) holonomic and (d) non-holonomic robots.

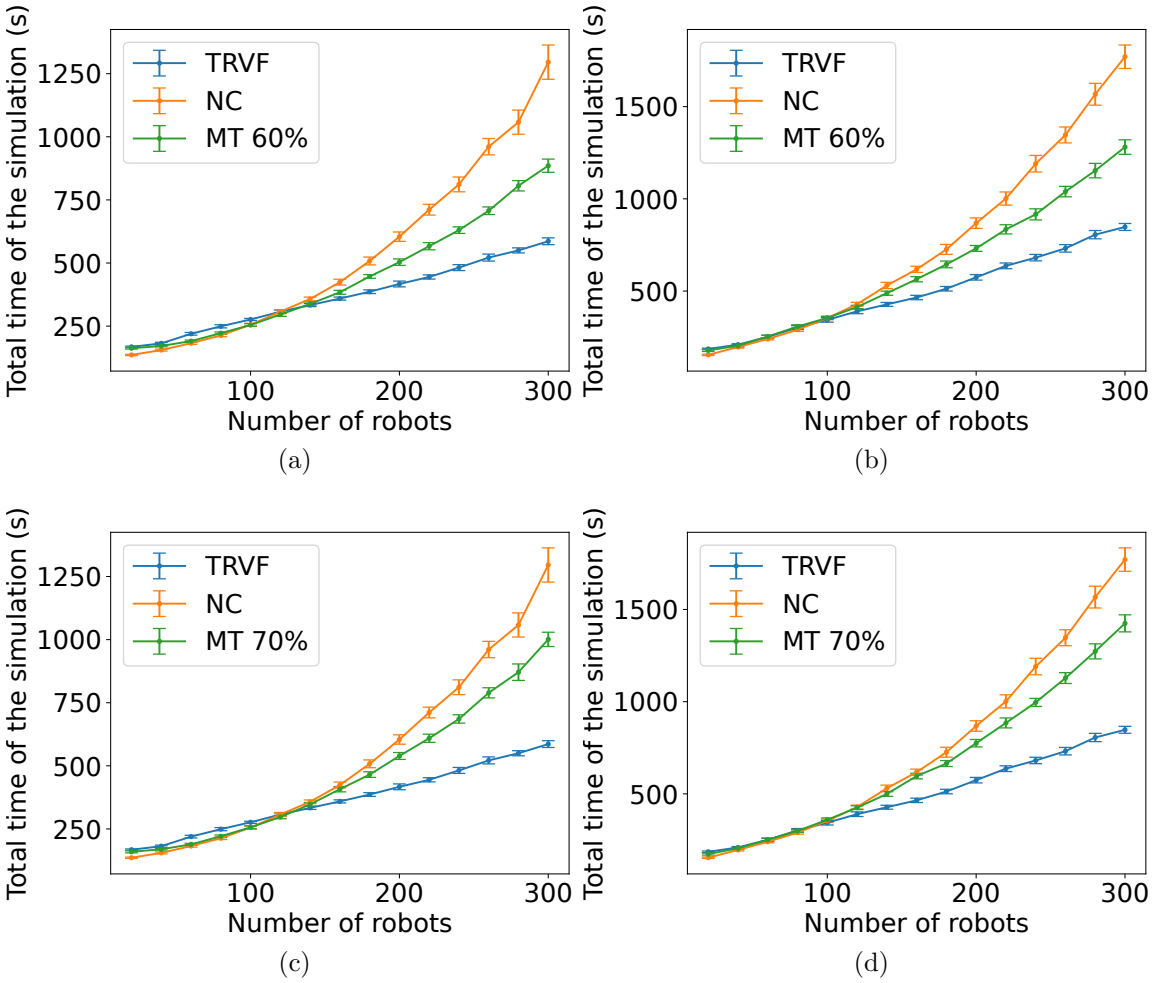


Figure D.19: Comparison of the total simulation time by the number of robots for the experiments with MT using TRVF as the swarm control algorithm and NC as the alternative algorithm by 60% of the robots for (a) holonomic and (b) non-holonomic robots and 70% for (c) holonomic and (d) non-holonomic robots.

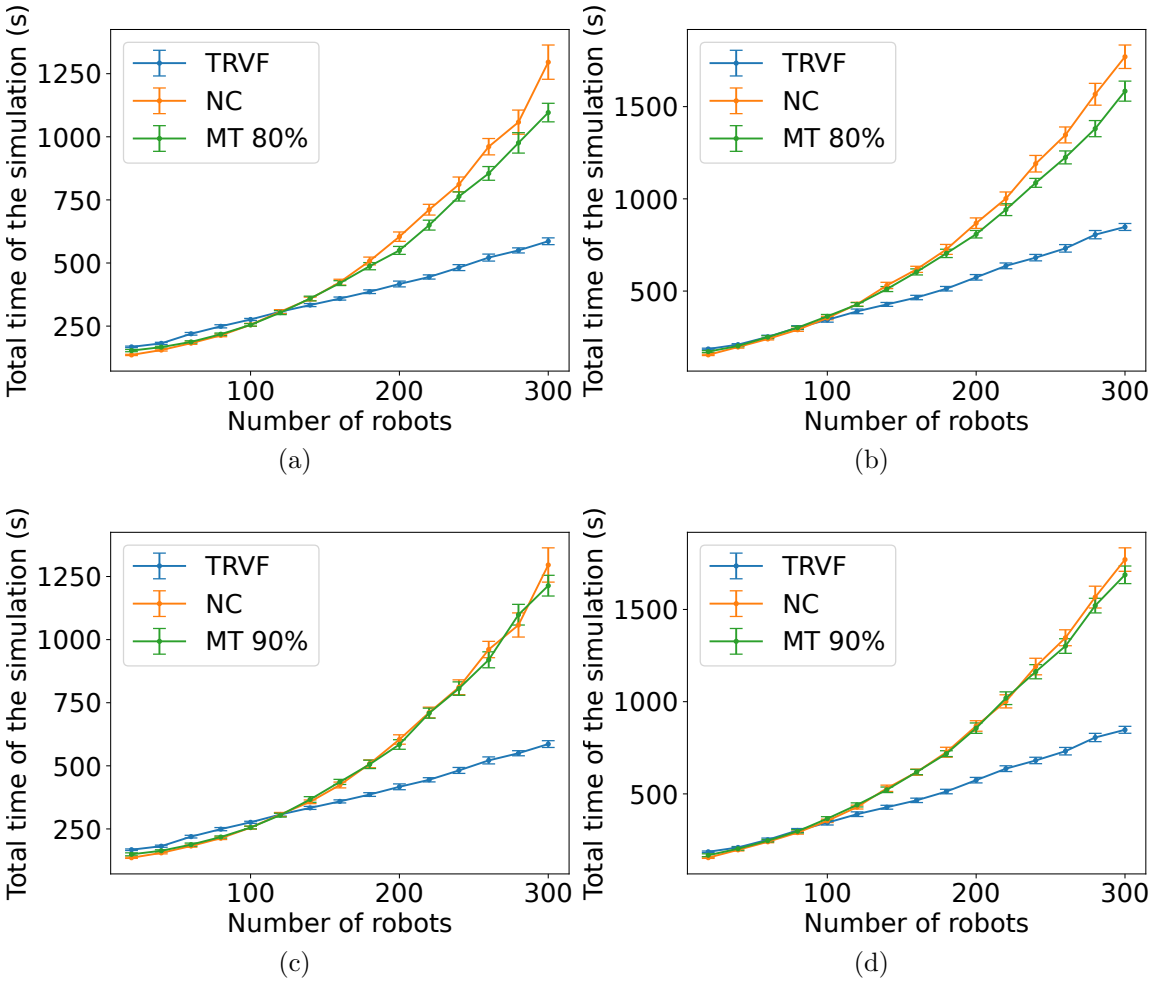


Figure D.20: Comparison of the total simulation time by the number of robots for the experiments with MT using TRVF as the swarm control algorithm and NC as the alternative algorithm by 80% of the robots for (a) holonomic and (b) non-holonomic robots and 90% for (c) holonomic and (d) non-holonomic robots.

Appendix E

Additional graphs for Chapter 6

E.1 Estimations for MT

Figure E.1–E.4 present estimations for MT with the proportion p of ad hoc robots from 20% to 90% in increments of 10% and SQF as the control algorithm of the aware robots.

Figures E.5–E.8 show the estimations for MT with TRVF as control algorithm of the aware robots with p from 20% to 90% in steps of 10%.

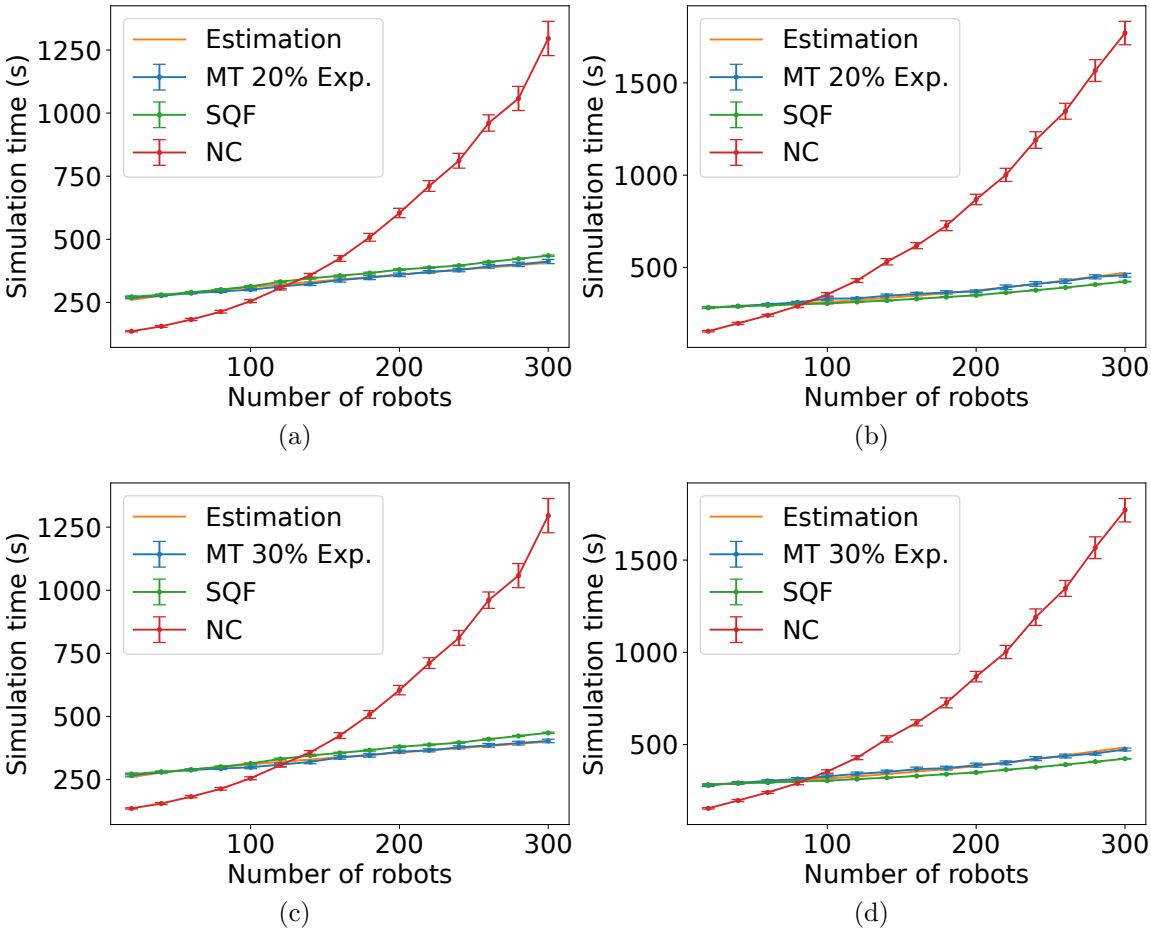


Figure E.1: Comparison of the estimated expected time and the experimental data for MT with NC as the alternative algorithm, the swarm control algorithm is the SQF algorithm, $p = 20\%$ for (a) holonomic robots ($C_{MTNC} = -0.0091$ and $C_{MTAw} = 0.9709$) and (b) non-holonomic robots ($C_{MTNC} = 0.0187$ and $C_{MTAw} = 1.0246$) and $p = 30\%$ for (c) holonomic robots ($C_{MTNC} = -0.0169$ and $C_{MTAw} = 0.9757$) and (d) non-holonomic robots ($C_{MTNC} = 0.026$ and $C_{MTAw} = 1.029$).

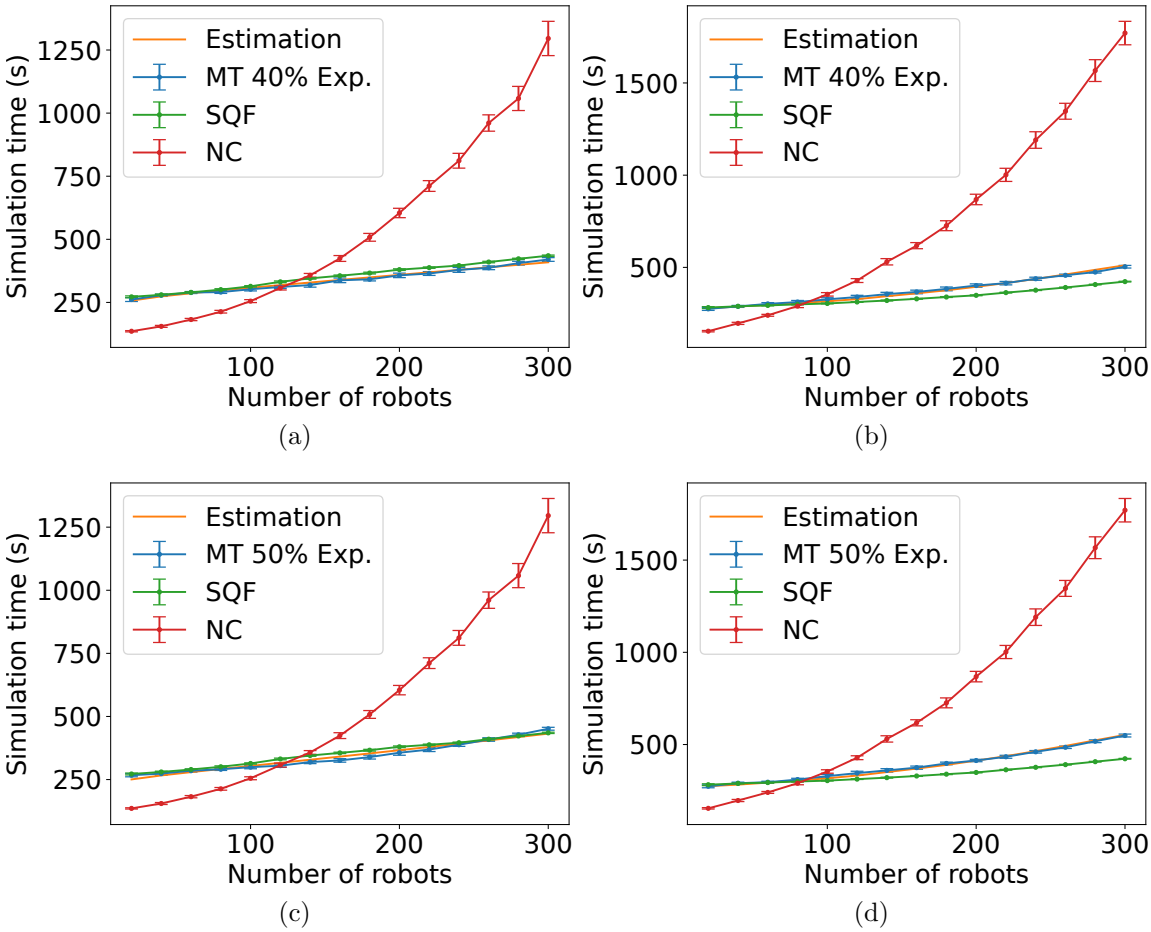


Figure E.2: Comparison of the estimated expected time and the experimental data for MT with NC as the alternative algorithm, the swarm control algorithm is the SQF algorithm, $p = 40\%$ for (a) holonomic robots ($C_{MTNC} = -0.0023$ and $C_{MTAw} = 0.9569$) and (b) non-holonomic robots ($C_{MTNC} = 0.0486$ and $C_{MTAw} = 1.0023$) and $p = 50\%$ for (c) holonomic robots ($C_{MTNC} = 0.0328$ and $C_{MTAw} = 0.9148$) and (d) non-holonomic robots ($C_{MTNC} = 0.0813$ and $C_{MTAw} = 0.9682$).

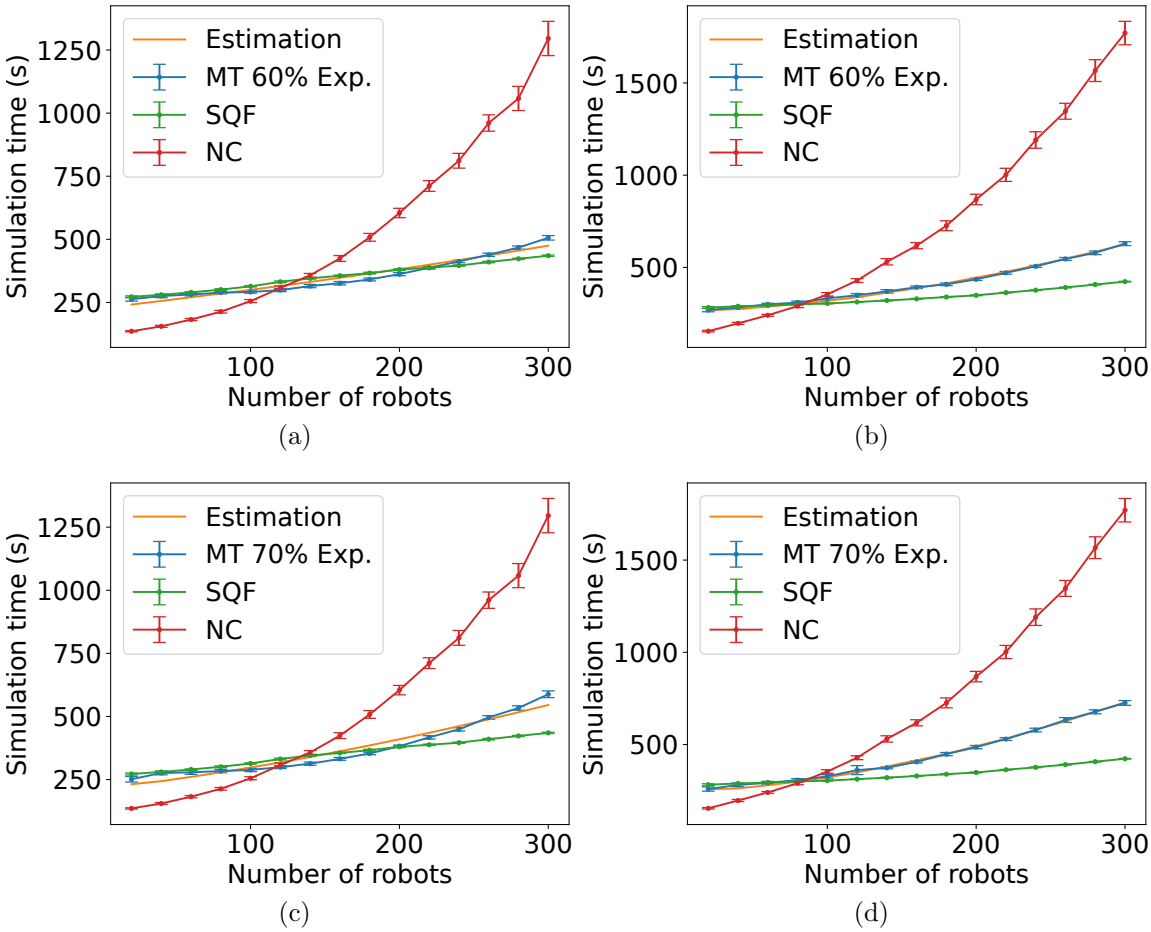


Figure E.3: Comparison of the estimated expected time and the experimental data for MT with NC as the alternative algorithm, the swarm control algorithm is the SQF algorithm, $p = 60\%$ for (a) holonomic robots ($C_{MTNC} = 0.093$ and $C_{MTAw} = 0.8514$) and (b) non-holonomic robots ($C_{MTNC} = 0.1383$ and $C_{MTAw} = 0.9088$) and $p = 70\%$ for (c) holonomic robots ($C_{MTNC} = 0.1842$ and $C_{MTAw} = 0.7678$) and (d) non-holonomic robots ($C_{MTNC} = 0.2217$ and $C_{MTAw} = 0.8241$).

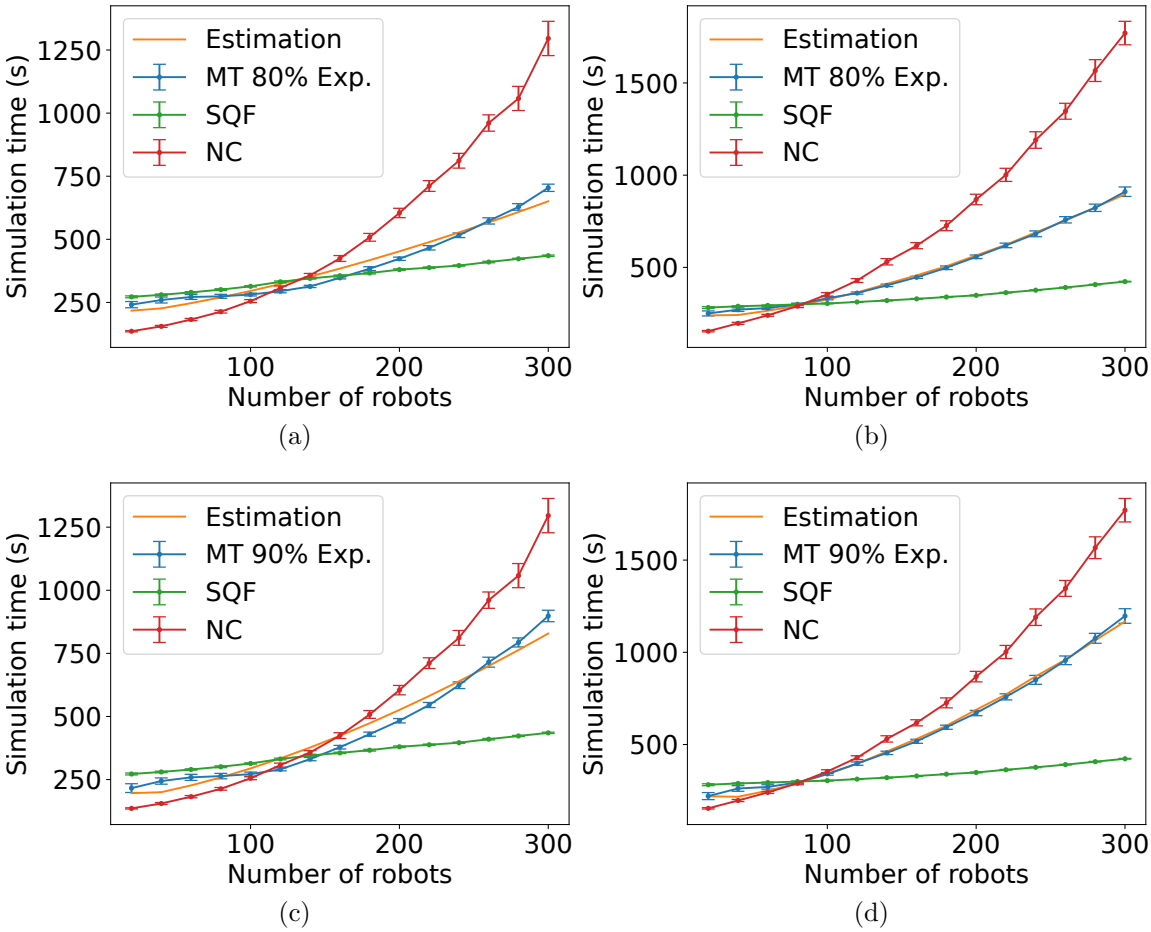


Figure E.4: Comparison of the estimated expected time and the experimental data for MT with NC as the alternative algorithm, the swarm control algorithm is the SQF algorithm, $p = 80\%$ for (a) holonomic robots ($C_{MTNC} = 0.3188$ and $C_{MTAw} = 0.6471$) and (b) non-holonomic robots ($C_{MTNC} = 0.3557$ and $C_{MTAw} = 0.6847$) and $p = 90\%$ for (c) holonomic robots ($C_{MTNC} = 0.5419$ and $C_{MTAw} = 0.4552$) and (d) non-holonomic robots ($C_{MTNC} = 0.5644$ and $C_{MTAw} = 0.4906$).

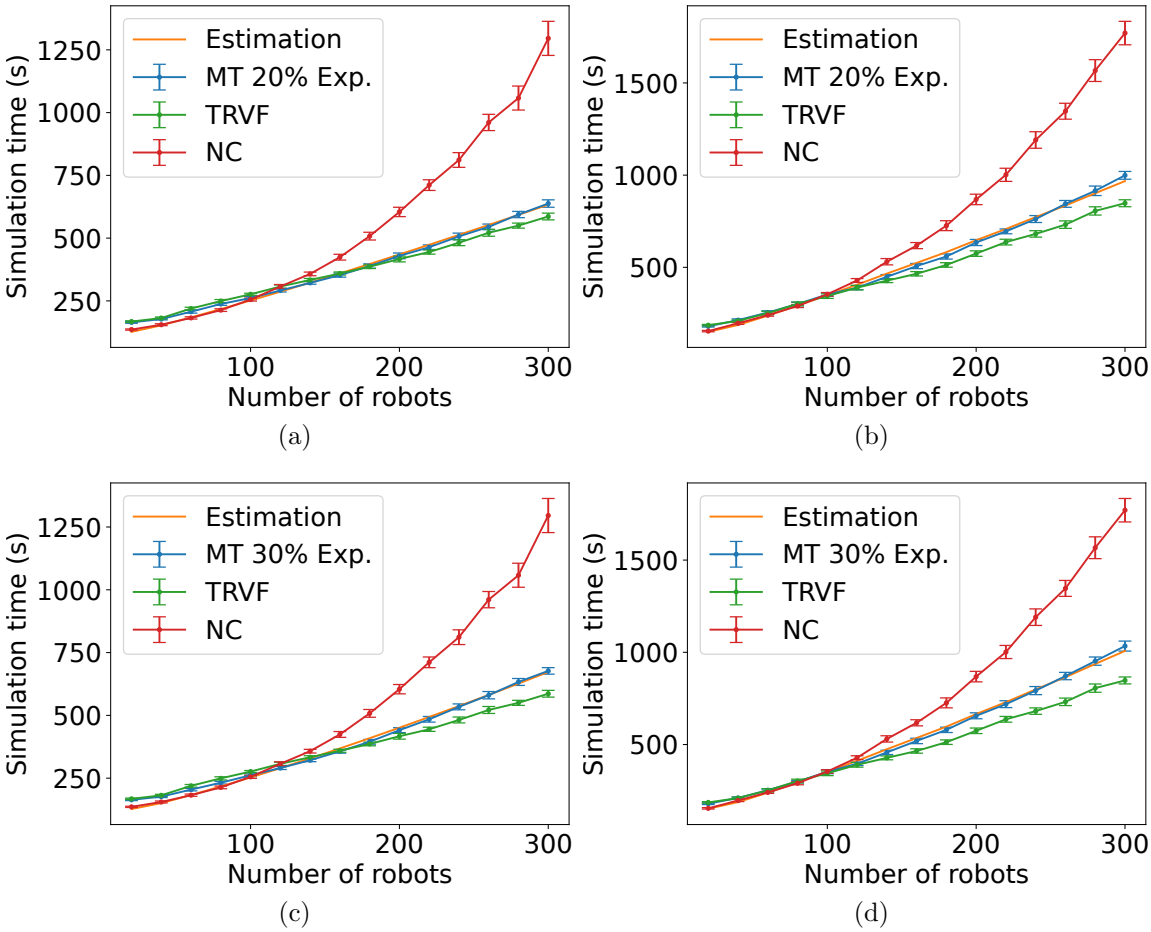


Figure E.5: Comparison of the estimated expected time and the experimental data for MT with NC as the alternative algorithm, the swarm control algorithm is the TRVF algorithm, $p = 20\%$ for (a) holonomic robots ($C_{MTNC} = 0.1062$ and $C_{MTAw} = 0.863$) and (b) non-holonomic robots ($C_{MTNC} = 0.1445$ and $C_{MTAw} = 0.8724$) and $p = 30\%$ for (c) holonomic robots ($C_{MTNC} = 0.1729$ and $C_{MTAw} = 0.7976$) and (d) non-holonomic robots ($C_{MTNC} = 0.1791$ and $C_{MTAw} = 0.8497$).

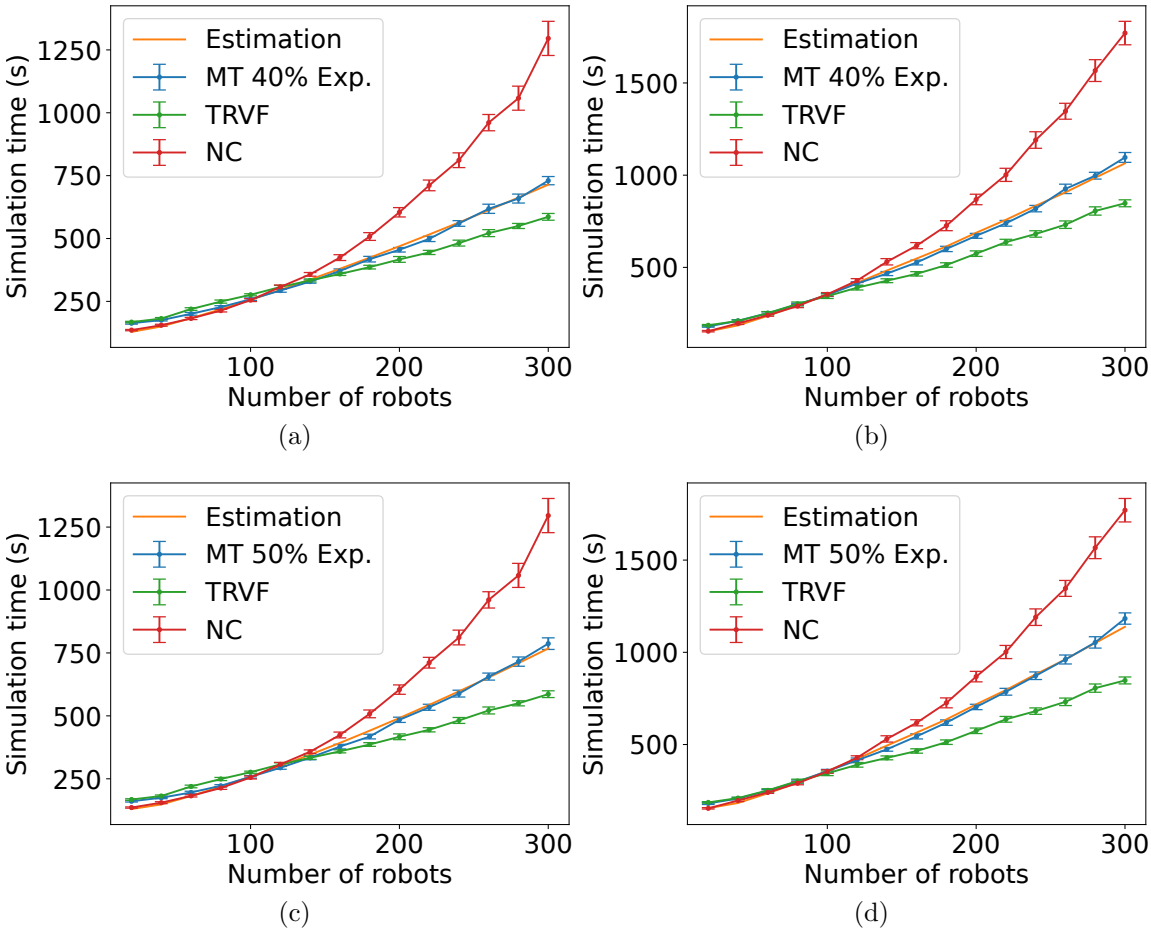


Figure E.6: Comparison of the estimated expected time and the experimental data for MT with NC as the alternative algorithm, the swarm control algorithm is the TRVF algorithm, $p = 40\%$ for (a) holonomic robots ($C_{MTNC} = 0.2346$ and $C_{MTAw} = 0.7458$) and (b) non-holonomic robots ($C_{MTNC} = 0.2395$ and $C_{MTAw} = 0.7943$) and $p = 50\%$ for (c) holonomic robots ($C_{MTNC} = 0.3221$ and $C_{MTAw} = 0.6633$) and (d) non-holonomic robots ($C_{MTNC} = 0.3178$ and $C_{MTAw} = 0.7238$).

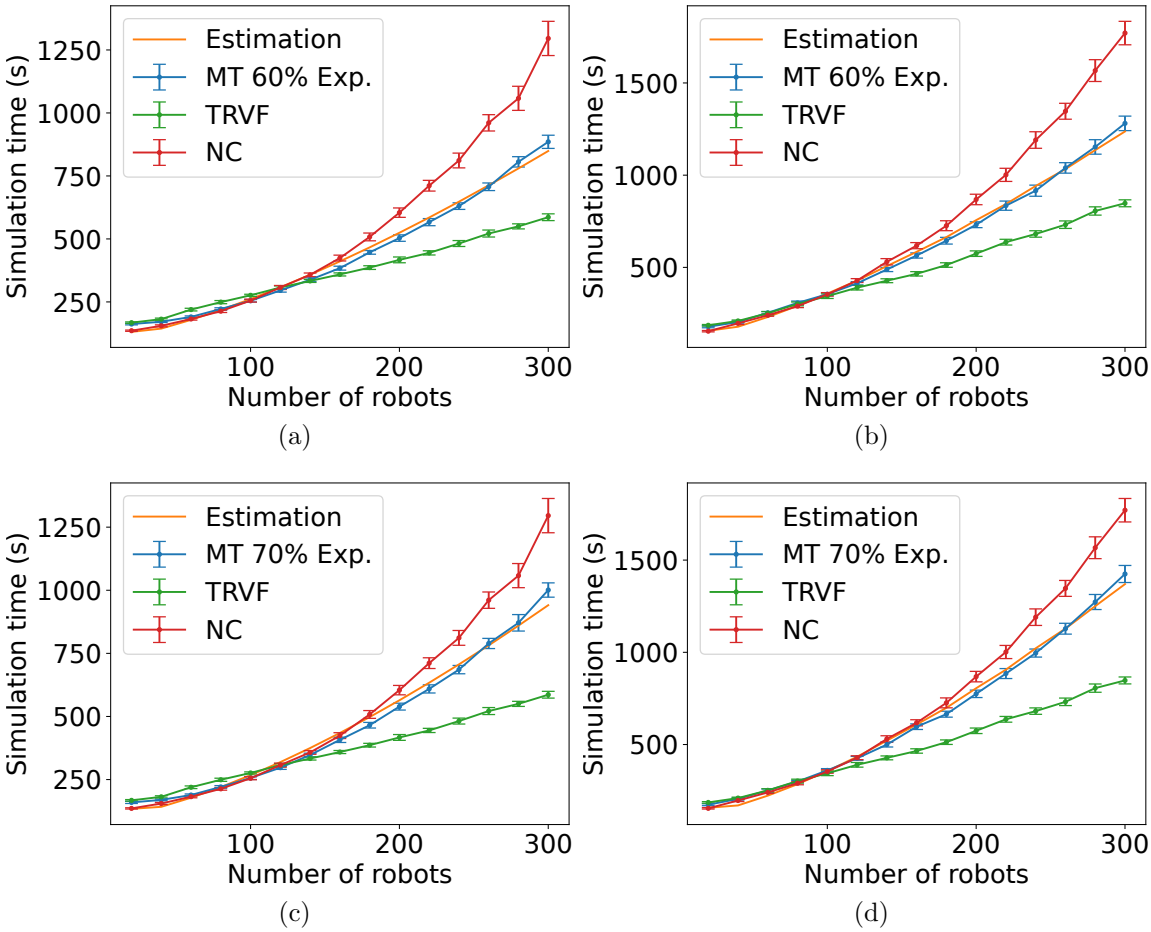


Figure E.7: Comparison of the estimated expected time and the experimental data for MT with NC as the alternative algorithm, the swarm control algorithm is the TRVF algorithm, $p = 60\%$ for (a) holonomic robots ($C_{MTNC} = 0.46$ and $C_{MTAw} = 0.5282$) and (b) non-holonomic robots ($C_{MTNC} = 0.4279$ and $C_{MTAw} = 0.6173$) and $p = 70\%$ for (c) holonomic robots ($C_{MTNC} = 0.603$ and $C_{MTAw} = 0.4011$) and (d) non-holonomic robots ($C_{MTNC} = 0.5836$ and $C_{MTAw} = 0.4588$).

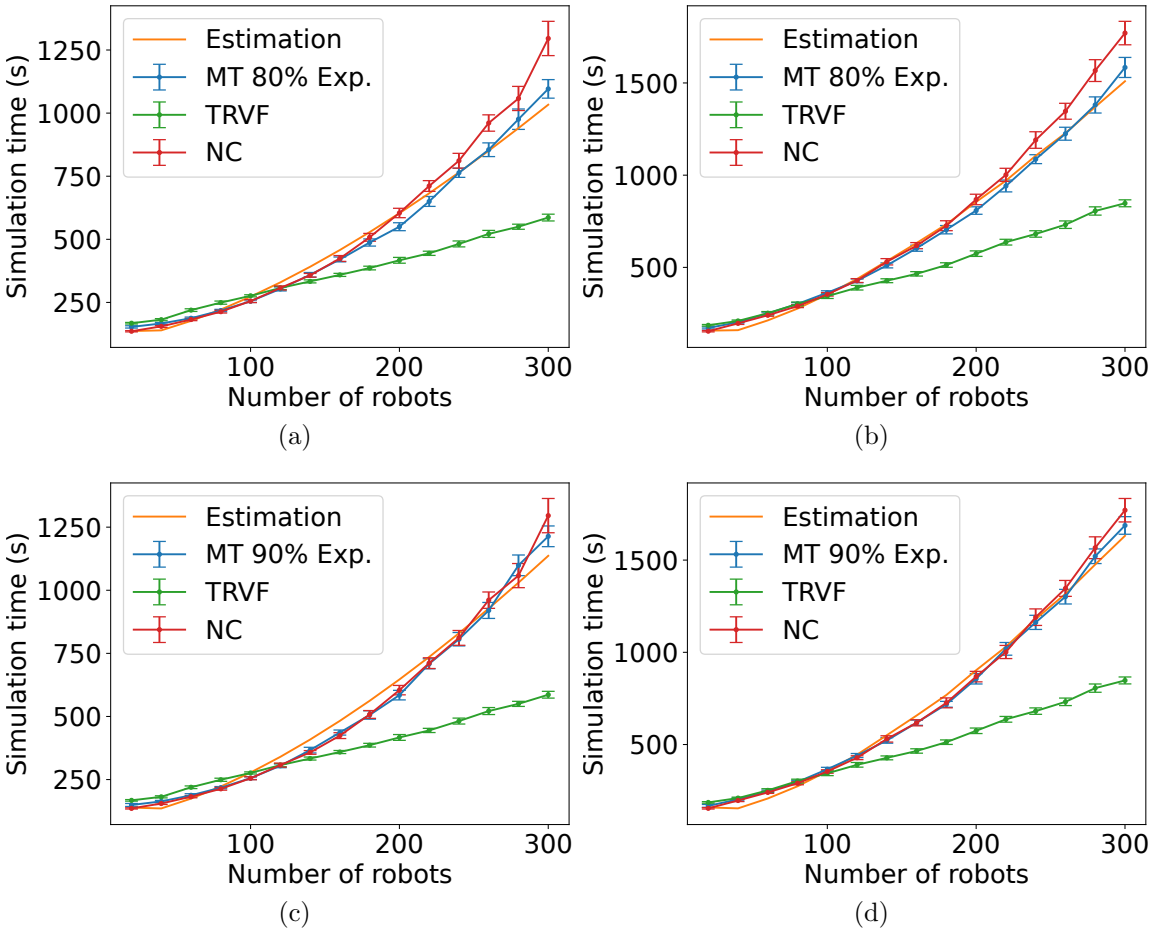


Figure E.8: Comparison of the estimated expected time and the experimental data for MT with NC as the alternative algorithm, the swarm control algorithm is the TRVF algorithm, $p = 80\%$ for (a) holonomic robots ($C_{MTNC} = 0.7498$ and $C_{MTAw} = 0.2673$) and (b) non-holonomic robots ($C_{MTNC} = 0.753$ and $C_{MTAw} = 0.2806$) and $p = 90\%$ for (c) holonomic robots ($C_{MTNC} = 0.9173$ and $C_{MTAw} = 0.1106$) and (d) non-holonomic robots ($C_{MTNC} = 0.8884$ and $C_{MTAw} = 0.1511$).

References

Abdullah, Al, Areej Malibari, and Mona Alkhozae (2014). "Students' performance prediction system using multi agent data mining technique". In: *International Journal of Data Mining & Knowledge Management Process* 4.5, p. 1. DOI: 10.5121/ijdkp.2014.4501.

Aeronautics, National and Space Administration NASA (2015). *NASA Swarmathon*. https://swarmathon.cs.unm.edu/. Accessed on 14 October 2023.

— (2018). *Swarmathon Improves Student Skills in Robotics, Computer Science*. https://www.nasa.gov/learning-resources/stem-engagement-at-nasa/swarmathon-improves-student-skills-in-robotics-computer-science/. Accessed on 14 October 2023.

Albrecht, Stefano V. and Peter Stone (2018). "Autonomous agents modelling other agents: A comprehensive survey and open problems". In: *Artificial Intelligence* 258, pp. 66–95. ISSN: 0004-3702. DOI: 10.1016/j.artint.2018.01.002.

Arnold, Ross D., Hiroyuki Yamaguchi, and Toshiyuki Tanaka (Dec. 2018). "Search and rescue with autonomous flying robots through behavior-based cooperative intelligence". In: *Journal of International Humanitarian Action* 3.1, p. 18. ISSN: 2364-3404. DOI: 10.1186/s41018-018-0045-4.

Aubourg, Lucie (2022). *Drone swarms can now fly autonomously through thick forest*. https://techxplore.com/news/2022-05-drone-swarms-autonomously-thick-forest.html. Accessed on 30 January 2023.

Avrahami, Sapir and Noa Agmon (2019). "The Robotic Swarm Contamination Problem". In: *International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, pp. 119–125. DOI: 10.1109/MRS.2019.8901090.

Barca, Jan Carlo and Y. Ahmet Sekercioglu (May 2013). "Swarm robotics reviewed". In: *Robotica* 31, pp. 345–359. DOI: 10.1017/S026357471200032X.

Barnes, Laura, MaryAnne Fields, and Kimon Valavanis (2007). "Unmanned ground vehicle swarm formation control using potential fields". In: *Mediterranean Conference on Control and Automation (MED)*, pp. 1–8. DOI: 10.1109/MED.2007.4433724.

Barrett, Samuel (2015). *Making Friends on the Fly: Advances in Ad Hoc Teamwork*. Studies in Computational Intelligence. USA: Springer International Publishing. ISBN: 9783319180694.

Barrett, Samuel and Peter Stone (2015). "Cooperating with Unknown Teammates in Complex Domains: A Robot Soccer Case Study of Ad Hoc Teamwork". In: *29th Conference on Artificial Intelligence (AAAI)*. AAAI'15. Austin, Texas: AAAI Press, pp. 2010–2016. ISBN: 0262511290. DOI: 10.1609/aaai.v29i1.9428.

Barrett, Samuel Rubin (2014). "Making Friends on the Fly: Advances in Ad Hoc Teamwork". PhD thesis. University of Texas at Austin.

Bartlett, Rodney J. and Monika Musiał (Feb. 2007). "Coupled-cluster theory in quantum chemistry". In: *Reviews of Modern Physics* 79 (1), pp. 291–352. DOI: 10.1103/RevModPhys.79.291.

Batra, Sumeet et al. (Nov. 2022). "Decentralized Control of Quadrotor Swarms with End-to-end Deep Reinforcement Learning". In: *5th Conference on Robot Learning (CoRL)*. Ed. by Aleksandra Faust, David Hsu, and Gerhard Neumann. Vol. 164. Proceedings of Machine Learning Research. PMLR, pp. 576–586.

Bayındır, Levent (2016). "A review of swarm robotics tasks". In: *Neurocomputing* 172, pp. 292–321. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2015.05.116.

Beavers, G. and H. Hexmoor (2001). "Teams of agents". In: *2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace*. Vol. 1, pp. 574–582. DOI: 10.1109/ICSMC.2001.969875.

Beni, Gerardo (2005). "From Swarm Intelligence to Swarm Robotics". In: *Swarm Robotics*. Ed. by Erol Şahin and William M. Spears. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1–9. ISBN: 978-3-540-30552-1.

Beni, Gerardo and Jing Wang (1993). "Swarm Intelligence in Cellular Robotic Systems". In: *NATO Advanced Workshop on Robots and Biological Systems*. Ed. by Paolo Dario, Giulio Sandini, and Patrick Aebischer. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 703–712. ISBN: 978-3-642-58069-7. DOI: 10.1007/978-3-642-58069-7.

Berg, Jur van den et al. (2011). "Reciprocal n-Body Collision Avoidance". In: *International Symposium in Robotics Research (ISRR)*. Ed. by Cédric Pradalier, Roland Siegwart, and Gerhard Hirzinger. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 3–19. ISBN: 978-3-642-19457-3. DOI: 10.1007/978-3-642-19457-3_1.

Bogdan, Paul, Guopeng Wei, and Radu Marculescu (2012). "Modeling populations of micro-robots for biological applications". In: *IEEE International Conference on Communications (ICC)*, pp. 6188–6192. DOI: 10.1109/ICC.2012.6364979.

Boogerd, F. C. et al. (May 2005). "Emergence and Its Place in Nature: A Case Study of Biochemical Networks". In: *Synthese* 145.1, pp. 131–164. ISSN: 1573-0964. DOI: 10.1007/s11229-004-4421-9.

Borrmann, Urs et al. (2015). "Control Barrier Certificates for Safe Swarm Behavior". In: *IFAC-PapersOnLine* 48.27. Analysis and Design of Hybrid Systems ADHS, pp. 68–73. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2015.11.154.

Bowling, Michael and Peter McCracken (July 2005). "Coordination and Adaptation in Impromptu Teams". In: *20th national conference on Artificial intelligence (AAAI)*. Vol. 1. AAAI'05. Pittsburgh, Pennsylvania: AAAI Press, pp. 53–58. ISBN: 157735236x.

Brambilla, Manuele et al. (2013). "Swarm robotics: a review from the swarm engineering perspective". In: *Swarm Intelligence* 7.1, pp. 1–41. DOI: 10.1007/s11721-012-0075-2.

Bristol Robotics Laboratory (2017). *Symbiotic evolutionary robot organisms (SYMBRION)*. http://www.brl.ac.uk/research/researchthemes/swarmrobotics/symbrion.aspx. Accessed on 30 January 2023.

Brown, Robert and Julie Adams (2023). "Congestion Analysis for the DARPA OFFSET CCAST Swarm". In: *Field Robotics* 3.1, pp. 190–221. DOI: 10.55417/fr.2023005.

Bullard, Kalesha et al. (2020). "Exploring Zero-Shot Emergent Communication in Embodied Multi-Agent Populations". In: *4th NeurIPS Workshop on Emergent Communication*. Virtual: JMLR.org.

Cai, Yahui et al. (2023). "A Self-organized Circular Formation Algorithm for Planar Robot Swarm". In: *5th Chinese Conference on Swarm Intelligence and Cooperative Control (CCSICC)*. Ed. by Zhang Ren, Mengyi Wang, and Yongzhao Hua. Singapore: Springer Nature Singapore, pp. 702–710. ISBN: 978-981-19-3998-3. DOI: 10.1007/978-981-19-3998-3_67.

Caloud, P. et al. (July 1990). "Indoor automation with many mobile robots". In: *IEEE International Workshop on Intelligent Robots and Systems (IROS)*. IROS, pp. 67–72. DOI: 10.1109/IROS.1990.262370.

Caprari, G. et al. (1998). "The autonomous micro robot "Alice": a platform for scientific and commercial applications". In: *1998 International Symposium on Micromechatronics and Human Science (MHA'98)*, pp. 231–235. DOI: 10.1109/MHS.1998.745787.

Caridi, Maria and Sergio Cavalieri (2004). "Multi-agent systems in production planning and control: an overview". In: *Production Planning & Control* 15.2, pp. 106–118. DOI: 10.1080/09537280410001662556.

Carleo, Giuseppe and Matthias Troyer (2017). "Solving the quantum many-body problem with artificial neural networks". In: *Science* 355.6325, pp. 602–606. DOI: 10.1126/science.aag2302.

Carlino, D., S. D. Boyles, and P. Stone (Oct. 2013). "Auction-Based Autonomous Intersection Management". In: *16th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 529–534. DOI: 10.1109/ITSC.2013.6728285.

Carlucho, Ignacio et al. (2022). "Cooperative Marine Operations via Ad Hoc Teams". In: *31st International Joint Conference on Artificial Intelligence (IJCAI). 31st International Joint Conference on Artificial Intelligence (IJCAI)*. Vienna.

Casella, George and Roger Berger (2002). *Statistical Inference*. 2nd ed. USA: Duxbury – Thomson Learning. ISBN: 0534243126.

Cenedese, A., C. Favaretto, and G. Occioni (2016). "Multi-agent swarm control through kuramoto modeling". In: *IEEE 55th Conference on Decision and Control (CDC)*, pp. 1820–1825. DOI: 10.1109/CDC.2016.7798529.

Chang, Hai-Chau and Lih-Chung Wang (2010). *A Simple Proof of Thue's Theorem on Circle Packing*. Preprint arXiv:1009.4322v1. arXiv: 1009.4322 [math.MG].

Chaudhari, Megha S., Bharti Patil, and Vaishali Raut (2019). "IoT based Waste Collection Management System for Smart Cities: An Overview". In: *3rd International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 802–805. DOI: 10.1109/ICCMC.2019.8819776.

Chen, Yongxin, Tryphon T. Georgiou, and Michele Pavon (Oct. 2018). "Steering the Distribution of Agents in Mean-Field Games System". In: *Journal of Optimization Theory and Applications* 179.1, pp. 332–357. ISSN: 1573-2878. DOI: 10.1007/s10957-018-1365-7.

Choudhury, Shushman et al. (2021). "Efficient Large-Scale Multi-Drone Delivery using Transit Networks". In: *Journal of Artificial Intelligence Research* 70, pp. 757–788. DOI: 10.1613/jair.1.12450.

Chung, S. et al. (2018). "A Survey on Aerial Swarm Robotics". In: *IEEE Transactions on Robotics* 34.4, pp. 837–855. DOI: 10.1109/TRO.2018.2857475.

Cicerone, Serafino et al. (2021). "MOBLOT: Molecular Oblivious Robots". In: *20th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*. AAMAS '21. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, pp. 350–358. ISBN: 9781450383073.

Cohen, Saar and Noa Agmon (2021). "Spatial Consensus-Prevention in Robotic Swarms". In: *20th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*. AAMAS '21. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, pp. 359–367. ISBN: 9781450383073.

Cortés, José C. Romero and Leonid B. Sheremetov (2002). "Model of Cooperation in Multi-agent Systems with Fuzzy Coalitions". In: *International Workshop of Central and Eastern Europe on Multi-Agent Systems (CEEMAS)*. Ed. by Barbara Dunin-Keplicz and Edward Nawarecki. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 263–272. DOI: 10.1007/3-540-45941-3_28.

Costa, Leandro R. et al. (2022). "OptiMaP: swarm-powered Optimized 3D Mapping Pipeline for emergency response operations". In: *18th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 269–276. DOI: 10.1109/DCOSS54816.2022.00052.

Couzin, Iain (2023). *Geometric Principles of Individual and Collective Decision-Making*. Keynote talk presented at the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2023). https://www.southampton.ac.uk/~eg/AAMAS2023/pdfs/p3.pdf. Accessed on 11 July 2023. London, UK.

Cui, Jiaxun et al. (2021). "Scalable Multiagent Driving Policies for Reducing Traffic Congestion". In: *20th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*. AAMAS '21. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, pp. 386–394. ISBN: 9781450383073.

Daduna, Hans, Victor Pestien, and S. Ramakrishnan (2003). "Asymptotic Throughput in Discrete-Time Cyclic Networks with Queue-Length-Dependent Service Rates". In: *Stochastic Models* 19.4, pp. 483–506. DOI: 10.1081/STM-120025401.

Demiris, Yiannis (Sept. 2007). "Prediction of intent in robotics and multi-agent systems". In: *Cognitive Processing* 8.3, pp. 151–158. ISSN: 1612-4790. DOI: 10.1007/s10339-007-0168-9.

Dias, Pollyanna G. Faria et al. (2021). "Swarm Robotics: A Perspective on the Latest Reviewed Concepts and Applications". In: *Sensors* 21.6. ISSN: 1424-8220. DOI: 10.3390/s21062062.

Dorigo, Marco, Guy Theraulaz, and Vito Trianni (2020). "Reflections on the future of swarm robotics". In: *Science Robotics* 5.49. DOI: 10.1126/scirobotics.abe4385.

dos Santos, Renato F. et al. (2023). "Parallel multi-speed Pursuit-Evasion Game algorithms". In: *Robotics and Autonomous Systems* 163, p. 104382. ISSN: 0921-8890. DOI: 10.1016/j.robot.2023.104382.

Dreizler, Reiner M. and Eberhard K. U. Gross (1995). *Density functional theory*. NATO ASI series. Series B, Physics ; v. 337. New York: Plenum Press. ISBN: 0306449056.

Duarte, Miguel et al. (Mar. 2016). "Evolution of Collective Behaviors for a Real Swarm of Aquatic Surface Robots". In: *PLOS ONE* 11.3, pp. 1–25. DOI: 10.1371/journal.pone.0151834.

Ducatelle, F. et al. (Sept. 2011). "Communication Assisted Navigation in Robotic Swarms: Self-Organization and Cooperation". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4981–4988. DOI: 10.1109/IROS.2011.6094454.

Dudek, G. et al. (1993). "A taxonomy for swarm robots". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vol. 1, pp. 441–447. DOI: 10.1109/IROS.1993.583135.

Elamvazhuthi, Karthik and Spring Berman (Nov. 2019). "Mean-field models in swarm robotics: a survey". In: *Bioinspiration & Biomimetics* 15.1, p. 015001. DOI: 10.1088/1748-3190/ab49a4.

Feltz, B., M. Crommelinck, and P. Goujon (2006). *Self-organization and Emergence in Life Sciences*. Synthese Library. Netherlands: Springer Netherlands. ISBN: 9781402039164.

Ferrera, Eduardo et al. (May 2017). "Decentralized Safe Conflict Resolution for Multiple Robots in Dense Scenarios". en. In: *Robotics and Autonomous Systems* 91, pp. 179–193. ISSN: 09218890. DOI: 10.1016/j.robot.2017.01.008.

Flammini, Benedetta, Davide Azzalini, and Francesco Amigoni (2023). "Multi-Agent Pickup and Delivery in Presence of Another Team of Robots". In: *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pp. 2562–2564.

Fujisawa, Ryusuke, Genki Ichinose, and Shigeto Dobata (Jan. 2019). "Regulatory mechanism predates the evolution of self-organizing capacity in simulated ant-like robots". In: *Communications Biology* 2.1, p. 25. ISSN: 2399-3642. DOI: 10.1038/s42003-018-0276-3.

Galstyan, A., T. Hogg, and K. Lerman (2005). "Modeling and mathematical analysis of swarms of microscopic robots". In: *IEEE Swarm Intelligence Symposium (SIS)*, pp. 201–208. DOI: 10.1109/SIS.2005.1501623.

Galvez, Reagan L. et al. (2017). "Obstacle avoidance algorithm for swarm of quadrotor unmanned aerial vehicle using artificial potential fields". In: *IEEE Region 10 Conference (TENCON)*, pp. 2307–2312. DOI: 10.1109/TENCON.2017.8228246.

Garnier, Simon, Jacques Gautrais, and Guy Theraulaz (June 2007). "The biological principles of swarm intelligence". In: *Swarm Intelligence* 1.1, pp. 3–31. ISSN: 1935-3820. DOI: 10.1007/s11721-007-0004-y.

Genter, Katie, Noa Agmon, and Peter Stone (May 2013). "Ad Hoc Teamwork for Leading a Flock". In: *12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. Saint Paul, MN, USA.

Genter, Katie and Peter Stone (2014). "Influencing a Flock via Ad Hoc Teamwork". In: *International Conference on Swarm Intelligence (ANTS)*. Ed. by Marco Dorigo et al. Cham: Springer International Publishing, pp. 110–121. DOI: 10.1007/978-3-319-09952-1_10.

Gerkey, Brian P., Richard T. Vaughan, and Andrew Howard (2003). "The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems". In: *11th International Conference on Advanced Robotics (ICAR)*. ICAR, pp. 317–323.

Giavitto, Jean-Louis et al. (May 2013). *Spatial Computing Workshop (SCW 2013) colocated with AAMAS (W09)*. https://hal.archives-ouvertes.fr/hal-00821901.

Giordano, Joseph P. et al. (2021). "Comparison of Desynchronization Methods for a Decentralized Swarm on a Logistical Resupply Problem". In: *20th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*. AAMAS

'21. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, pp. 1510–1511. ISBN: 9781450383073.

Glossman-Mitnik, Daniel (2019). *Density Functional Theory*. Digital: IntechOpen. ISBN: 978-1-78985-168-7.

Golden (2021). *Swarm Robotics*. https://golden.com/wiki/Swarm_robotics-AD9D39. Accessed on 30 January 2023.

Grabe, Thiago M. et al. (2021). "Stand by me: Learning to keep cohesion in the navigation of heterogeneous swarms". In: *The 4th International Symposium on Swarm Behavior and Bio-inspired Robotics (DARS-SWARM)*. Online.

Graciano Santos, Vinicius and Luiz Chaimowicz (2011). "Hierarchical congestion control for robotic swarms". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4372–4377. DOI: 10.1109/IROS.2011.6094640.

Graham, Ronald L., Donald E. Knuth, and Oren Patashnik (1994). *Concrete Mathematics: A Foundation for Computer Science*. 2nd. Reading, MA, USA: Addison-Wesley Professional. ISBN: 0201558025.

Grossman, D.D. (Oct. 1988). "Traffic control of multiple robot vehicles". In: *IEEE Journal of Robotics and Automation* 4.5, pp. 491–497. DOI: 10.1109/56.20433.

Haken, H. (1983). *Synergetics: An Introduction : Nonequilibrium Phase Transitions and Self-organization in Physics, Chemistry, and Biology*. Germany: Springer. ISBN: 9783540088660.

Hamann, Heiko (Jan. 2018). *Swarm Robotics: A Formal Approach*. Germany: Springer. ISBN: 978-3-319-74526-8.

Hanga, Khadijah M. and Yevgeniya Kovalchuk (2019). "Machine learning and multiagent systems in oil and gas industry applications: A survey". In: *Computer Science Review* 34, p. 100191. ISSN: 1574-0137. DOI: 10.1016/j.cosrev.2019.08.002.

Hewawasam, H. S., M. Yousef Ibrahim, and Gayan Kahandawa Appuhamillage (2022). "Past, Present and Future of Path-Planning Algorithms for Mobile Robot Navigation in Dynamic Environments". In: *IEEE Open Journal of the Industrial Electronics Society* 3, pp. 353–365. DOI: 10.1109/OJIES.2022.3179617.

Hockney, Roger W. (1994). "The communication challenge for MPP: Intel Paragon and Meiko CS-2". In: *Parallel Computing* 20.3, pp. 389–398. ISSN: 0167-8191. DOI: 10.1016/S0167-8191(06)80021-9.

Holland, J.H. (2000). *Emergence: From Chaos to Order*. Popular science / Oxford University Press. Oxford University Press. ISBN: 9780192862112.

Holmes, Mark H. (2015). *Introduction to Perturbation Methods*. Texts in Applied Mathematics. New York: Springer New York. ISBN: 978-1-4899-9613-8.

Hoshino, Satoshi (2011). "Multi-robot coordination methodology in congested systems with bottlenecks". In: *IEEE/RSJ International Conference on Intelligent Robots*

and Systems (IROS). IROS. IEEE, pp. 2810–2816. DOI: 10.1109/IROS.2011.6094438.

Hoshino, Satoshi and Hiroya Seki (Aug. 2013). "Multi-Robot Coordination for Jams in Congested Systems". en. In: *Robotics and Autonomous Systems* 61.8, pp. 808–820. ISSN: 09218890. DOI: 10.1016/j.robot.2013.04.011.

Hoy, Michael, Alexey S. Matveev, and Andrey V. Savkin (Mar. 2015). "Algorithms for Collision-Free Navigation of Mobile Robots in Complex Cluttered Environments: A Survey". en. In: *Robotica* 33.3, pp. 463–497. ISSN: 0263-5747, 1469-8668. DOI: 10.1017/S0263574714000289.

Hu, Hengyuan et al. (July 2020). ""Other-Play" for Zero-Shot Coordination". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. Virtual: PMLR, pp. 4399–4410.

Huang, Kerson (2008). *Statistical Mechanics*. Second. India: Wiley India Pvt. Limited. ISBN: 9788126518494.

Iannini, M. L. L. and Ronald Dickman (2016). "Kinetic theory of vehicular traffic". In: *American Journal of Physics* 84.2, pp. 135–145. DOI: 10.1119/1.4935895.

Jain, Preeti (2019). *Swarm Robotics*. https://www.engineersgarage.com/swarm-robotics/. Accessed on 30 January 2023.

Jiao, Chen et al. (Aug. 2022). "Bose—Einstein condensates with tunable spin—orbit coupling in the two-dimensional harmonic potential: The ground-state phases, stability phase diagram and collapse dynamics". In: *Frontiers of Physics* 17.6, p. 61503. ISSN: 2095-0470. DOI: 10.1007/s11467-022-1180-3.

Jones, E.G. et al. (2006). "Dynamically Formed Heterogeneous Robot Teams Performing Tightly-Coordinated Tasks". In: *IEEE International Conference on Robotics and Automation (ICRA)*. Orlando, Florida, USA. DOI: 10.1109/ROBOT.2006.1641771.

Kato, S., S. Nishiyama, and J. Takeno (July 1992). "Coordinating Mobile Robots By Applying Traffic Rules". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IROS, pp. 1535–1541. DOI: 10.1109/IROS.1992.594218.

Kato, T. (2012). *A Short Introduction to Perturbation Theory for Linear Operators*. New York: Springer New York. ISBN: 9781461257004.

Khaluf, Yara and Marco Dorigo (June 2016). "Modeling Robot Swarms Using Integrals of Birth-Death Processes". In: *ACM Transactions on Autonomous and Adaptive Systems* 11.2. ISSN: 1556-4665. DOI: 10.1145/2870637.

Khatib, O. (1985). "Real-time obstacle avoidance for manipulators and mobile robots". In: *IEEE International Conference on Robotics and Automation (ICRA)*. Vol. 2, pp. 500–505. DOI: 10.1109/ROBOT.1985.1087247.

Kobayashi, Kazuho, Takehiro Higuchi, and Seiya Ueno (2023). "Hierarchical Control and Subgroup Formation for Robotic Swarms in Patrol Missions". In: *International Journal of Mechanical Engineering and Robotics Research* 12.5. DOI: 10.18178/ijmerr.12.5.258-263.

Kube, C. Ronald and Hong Zhang (1993). "Collective Robotics: From Social Insects to Robots". In: *Adaptive Behavior* 2.2, pp. 189–218. DOI: 10.1177/105971239300200204.

Kuzemsky, A.L. (2017). *Statistical Mechanics And The Physics Of Many-particle Model Systems*. Digital: World Scientific Publishing Company. ISBN: 9789813145658.

Lasry, Jean-Michel and Pierre-Louis Lions (Mar. 2007). "Mean field games". In: *Japanese Journal of Mathematics* 2.1, pp. 229–260. ISSN: 1861-3624. DOI: 10.1007/s11537-007-0657-8.

Li, Yangmin and Xin Chen (May 2006). "Modeling and Simulation of Swarms for Collecting Objects". In: *Robotica* 24.3, pp. 315–324. ISSN: 0263-5747. DOI: 10.1017/S0263574705002195.

Lima, Danielli A. and Gina M. B. Oliveira (2017). "A cellular automata ant memory model of foraging in a swarm of robots". In: *Applied Mathematical Modelling* 47, pp. 551–572. ISSN: 0307-904X. DOI: 10.1016/j.apm.2017.03.021.

Liu, Jun et al. (2015). "Robotic Adherent Cell Injection for Characterizing Cell-Cell Communication". In: *IEEE Transactions on Biomedical Engineering* 62.1, pp. 119–125. DOI: 10.1109/TBME.2014.2342036.

Luca, Alessandro De and Giuseppe Oriolo (1994). "Local Incremental Planning for Nonholonomic Mobile Robots". In: *IEEE International Conference on Robotics and Automation (ICRA)*. ICRA, pp. 104–110. DOI: 10.1109/ROBOT.1994.351003.

Ma, Aaron, Michael Ouimet, and Jorge Cortés (Mar. 2020). "Hierarchical reinforcement learning via dynamic subspace search for multi-agent planning". In: *Autonomous Robots* 44.3, pp. 485–503. ISSN: 1573-7527. DOI: 10.1007/s10514-019-09871-2.

Mabrouk, M. H. and C. R. McInnes (2008). "Solving the potential field local minimum problem using internal agent states". In: *Robotics and Autonomous Systems* 56.12. Towards Autonomous Robotic Systems 2008: Mobile Robotics in the UK, pp. 1050–1060. ISSN: 0921-8890. DOI: 10.1016/j.robot.2008.09.006.

MacAlpine, Patrick and Peter Stone (2017). "Evaluating Ad Hoc Teamwork Performance in Drop-In Player Challenges". In: *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. Ed. by Gita Sukthankar and Juan A. Rodriguez-Aguilar. Cham: Springer International Publishing, pp. 168–186. ISBN: 978-3-319-71682-4. DOI: 10.1007/978-3-319-71682-4_11.

MacAlpine, Patrick et al. (2014). "The RoboCup 2013 drop-in player challenges: Experiments in ad hoc teamwork". In: *IEEE/RSJ International Conference on*

Intelligent Robots and Systems (IROS), pp. 382–387. DOI: 10.1109/IROS.2014.6942588.

Majcherczyk, Nathalie et al. (2018). "Decentralized Connectivity-Preserving Deployment of Large-Scale Robot Swarms". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4295–4302. DOI: 10.1109/IROS.2018.8594422.

Mandl, Franz (2013). *Statistical Physics*. Manchester Physics Series. United Kingdom: Wiley. ISBN: 9781118723432.

Mannone, Maria, Valeria Seidita, and Antonio Chella (2022). "Categories, Quantum Computing, and Swarm Robotics: A Case Study". In: *Mathematics* 10.3, p. 372. ISSN: 2227-7390. DOI: 10.3390/math10030372.

March, N.H., W.H. Young, and S. Sampanthar (1995). *The Many-Body Problem in Quantum Mechanics*. Dover books on physics. United Kingdom: Dover Publications. ISBN: 9780486687544.

Marcolino, Leandro Soriano and Luiz Chaimowicz (May 2008). "No Robot Left behind: Coordination to Overcome Local Minima in Swarm Navigation". en. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Pasadena, CA, USA: IEEE, pp. 1904–1909. ISBN: 978-1-4244-1646-2. DOI: 10.1109/ROBOT.2008.4543485.

— (2009). "Traffic control for a swarm of robots: Avoiding target congestion". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1955–1961. DOI: 10.1109/IROS.2009.5354407.

Marcolino, Leandro Soriano et al. (Aug. 2017). "Avoiding Target Congestion on the Navigation of Robotic Swarms". en. In: *Autonomous Robots* 41.6, pp. 1297–1320. ISSN: 1573-7527. DOI: 10.1007/s10514-016-9577-x.

Masud, U. et al. (2020). "Traffic Congestion Avoidance System Using Foreground Estimation and Cascade Classifier". In: *IEEE Access* 8, pp. 178859–178869. DOI: 10.1109/ACCESS.2020.3027715.

Mataric, Maja J. (1995). "Designing and Understanding Adaptive Group Behavior". In: *Adaptive Behavior* 4.1, pp. 51–80. DOI: 10.1177/105971239500400104.

Matoui, Fethi, Boumedyen Boussaid, and Mohamed Naceur Abdelkrim (2015). "Local minimum solution for the potential field method in multiple robot motion planning task". In: *16th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, pp. 452–457. DOI: 10.1109/STA.2015.7505223.

Melo, Francisco S. and Alberto Sardinha (Mar. 2016). "Ad Hoc Teamwork by Learning Teammates' Task". In: *Autonomous Agents and Multi-Agent Systems* 30.2, pp. 175–219. ISSN: 1387-2532. DOI: 10.1007/s10458-015-9280-x.

Menashe, Nofar and Noa Agmon (2022). "Leading a Swarm with Signals". In: *Distributed Autonomous Robotic Systems (DARS)*. Ed. by Fumitoshi Matsuno, Shun-

ichi Azuma, and Masahito Yamamoto. Cham: Springer International Publishing, pp. 16–30. ISBN: 978-3-030-92790-5. DOI: 10.1007/978-3-030-92790-5_2.

Miley, Jessica (Mar. 2018). *This Start-Up Uses Human Swarm Intelligence to Develop AI That Can Predict the Future*. https://interestingengineering.com/this-start-up-uses-human-swarm-intelligence-to-develop-ai-that-can-predict-the-future. Accessed on 02 October 2022.

Mirsky, Reuth et al. (2022). "A Survey of Ad Hoc Teamwork Research". In: *19th European Conference on Multi-Agent Systems (EUMAS)*. Ed. by Dorothea Baumeister and Jörg Rothe. Cham: Springer International Publishing, pp. 275–293. ISBN: 978-3-031-20614-6. DOI: 10.1007/978-3-031-20614-6_16.

Morin, David (2007). *Introduction to classical mechanics : with problems and solutions*. Cambridge, United Kingdom: Cambridge University Press, p. 719. ISBN: 978-0521876223.

Mărgăritescu, Mihai et al. (2020). "Control of an Autonomous Mobile Waste Collection Robot". In: *International Conference of Mechatronics and Cyber - MixMechatronics*. Ed. by Gheorghe Ion Gheorghe. Cham: Springer International Publishing, pp. 51–63. ISBN: 978-3-030-26991-3.

Navarro, Iñaki and Fernando Matía (2013). "An introduction to swarm robotics". In: *International Scholarly Research Notices* 2013. DOI: 10.5402/2013/608164.

Nelson, Derek et al. (2006). "Vector field path following for small unmanned air vehicles". In: *American Control Conference (ACC)*, pp. 5788–5794. DOI: 10.1109/ACC.2006.1657648.

Oliveira, Gustavo P. M., Fabrício R. Inácio, and Douglas G. Macharet (2017). "Robotic swarms coordination strategies for minimizing group target congestion". In: *2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR)*, pp. 1–6. DOI: 10.1109/SBR-LARS-R.2017.8215286.

Opper, M. and D. Saad (2001). *Advanced Mean Field Methods: Theory and Practice*. Neural information processing series. USA: MIT Press. ISBN: 9780262150545.

Palanca, Javier et al. (2020). "SPADE 3: Supporting the New Generation of Multi-Agent Systems". In: *IEEE Access* 8, pp. 182537–182549. DOI: 10.1109/ACCESS.2020.3027357.

Panait, Liviu and Sean Luke (2003). "Evolving Foraging Behaviors". In: *Second International Workshop on the Mathematics and Algorithms of Social Insects*, pp. 131–138.

Passos, Yuri Tavares dos (2012). *Algoritmos de controle de tráfego para exames de robôs com alvos em comum*. Dissertation in Portuguese. Brazil.

— (2022a). *Swarm Ad Hoc Follower*. https://github.com/yuri-tavares/swarm-ad-hoc-follower. Accessed on 28 February 2023.

Passos, Yuri Tavares dos (2022b). *Swarm Common Target Area Congestion*. https://github.com/yuri-tavares/swarm-common-target-area-congestion. Accessed on 10 October 2022.

— (2022c). *Swarm Strategies*. https://github.com/yuri-tavares/swarm-strategies. Accessed on 10 October 2022.

Passos, Yuri Tavares dos, Xavier Duquesne, and Leandro Soriano Marcolino (2022). "On the Throughput of the Common Target Area for Robotic Swarm Strategies". In: *Mathematics* 10.14, p. 2482. ISSN: 2227-7390. DOI: 10.3390/math10142482.

— (2023). "Congestion control algorithms for robotic swarms with a common target based on the throughput of the target area". In: *Robotics and Autonomous Systems* 159, p. 104284. ISSN: 0921-8890. DOI: 10.1016/j.robot.2022.104284.

Pérez-Torres, Jhon Fredy (Apr. 2019). "Dilemma of the "Best Wavefunction": Comparing Results of the STO-NG Procedure versus the Linear Variational Method". In: *Journal of Chemical Education* 96.4, pp. 704–707. ISSN: 0021-9584. DOI: 10.1021/acs.jchemed.8b00959.

Pourpanah, Farhad et al. (2017). "A Q-learning-based multi-agent system for data classification". In: *Applied Soft Computing* 52, pp. 519–531. ISSN: 1568-4946. DOI: 10.1016/j.asoc.2016.10.016.

Prigogine, I. and F. C. Andrews (1960). "A Boltzmann-Like Approach for Traffic Flow". In: *Operations Research* 8.6, pp. 789–797. DOI: 10.1287/opre.8.6.789.

Prigogine, I. and R. Herman (1971). *Kinetic Theory of Vehicular Traffic*. New York: Elsevier. ISBN: 9780444000828.

Red Blob Games (2021). *Hexagonal Grids*. https://www.redblobgames.com/grids/hexagons/. Accessed on 16 November 2021.

Rubenstein, Michael, Alejandro Cornejo, and Radhika Nagpal (2014). "Programmable self-assembly in a thousand-robot swarm". In: *Science* 345.6198, pp. 795–799. DOI: 10.1126/science.1254295.

Rudd, Lael (2017). *OFFensive Swarm-Enabled Tactics (OFFSET)*. https://www.darpa.mil/program/offensive-swarm-enabled-tactics. Accessed on 14 October 2023.

Russell, Stuart and Peter Norvig (2020). *Artificial Intelligence: A Modern Approach*. Fourth. Pearson. ISBN: 9780134610993.

Sahin, Erol (2005). "Swarm Robotics: From Sources of Inspiration to Domains of Application". In: *International Workshop in Swarm Robotics (SAB)*. Ed. by Erol Sahin and William M. Spears. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 10–20. ISBN: 978-3-540-30552-1. DOI: 10.1007/978-3-540-30552-1_2.

Sahin, Erol et al. (2008). "Swarm Robotics". In: *Swarm Intelligence: Introduction and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 87–100. ISBN: 978-3-540-74089-6. DOI: 10.1007/978-3-540-74089-6_3.

Salimi, Mehdi et al. (Sept. 2016). "On a Fixed Duration Pursuit Differential Game with Geometric and Integral Constraints". In: *Dynamic Games and Applications* 6.3, pp. 409–425. ISSN: 2153-0793. DOI: 10.1007/s13235-015-0161-3.

Saska, Martin et al. (July 2020). "Formation control of unmanned micro aerial vehicles for straitened environments". In: *Autonomous Robots* 44.6, pp. 991–1008. ISSN: 1573-7527. DOI: 10.1007/s10514-020-09913-0.

Shahar, Tomer et al. (May 2021). "Safe Multi-Agent Pathfinding with Time Uncertainty". In: *Journal of Artificial Intelligence Research* 70, pp. 923–954. ISSN: 1076-9757. DOI: 10.1613/jair.1.12397.

Sharon, Guni and Peter Stone (2017). "A Protocol for Mixed Autonomous and Human-Operated Vehicles at Intersections". In: *Autonomous Agents and Multiagent Systems (AAMAS)*. Ed. by Gita Sukthankar and Juan A. Rodriguez-Aguilar. Vol. 10642. Lecture Notes in Artificial Intelligence. New York: Springer International Publishing, pp. 151–167. DOI: 10.1007/978-3-319-71682-4_10.

Sharon, Guni et al. (2017). "Network-wide adaptive tolling for connected and automated vehicles". In: *Transportation Research Part C: Emerging Technologies* 84, pp. 142–157. ISSN: 0968-090X. DOI: 10.1016/j.trc.2017.08.019.

Sharon, Guni et al. (2018). "Traffic Optimization For a Mixture of Self-interested and Compliant Agents". In: *32nd Conference on Artificial Intelligence (AAAI)*. AAAI, pp. 1202–1209. DOI: 10.1609/aaai.v32i1.11444.

Shavitt, I. and R.J. Bartlett (2009). *Many-Body Methods in Chemistry and Physics: MBPT and Coupled-Cluster Theory*. Cambridge Molecular Science. United Kingdom: Cambridge University Press. ISBN: 9780521818322.

Shen, Wei-Min et al. (July 2004). "Hormone-Inspired Self-Organization and Distributed Control of Robotic Swarms". In: *Autonomous Robots* 17.1, pp. 93–105. ISSN: 1573-7527. DOI: 10.1023/B:AURO.0000032940.08116.f1.

Siegwart, Roland, Illah R. Nourbakhsh, and Davide Scaramuzza (2011). *Introduction to Autonomous Mobile Robots, second edition*. Intelligent Robotics and Autonomous Agents series. MIT Press. ISBN: 9780262015356.

Soria, Enrica (2022). "Swarms of flying robots in unknown environments". In: *Science Robotics* 7.66, eabq2215. DOI: 10.1126/scirobotics.abq2215.

Stewart, Robert L. and R. Andrew Russell (2006). "A Distributed Feedback Mechanism to Regulate Wall Construction by a Robotic Swarm". In: *Adaptive Behavior* 14.1, pp. 21–51. DOI: 10.1177/105971230601400104.

Stone, Peter et al. (July 2010). "Ad Hoc Autonomous Agent Teams: Collaboration without Pre-Coordination". In: *Twenty-Fourth Conference on Artificial Intelligence (AAAI)*. DOI: 10.1609/aaai.v24i1.7529.

Tarapore, Danesh, Roderich Groß, and Klaus-Peter Zauner (2020). "Sparse Robot Swarms: Moving Swarms to Real-World Applications". In: *Frontiers in robotics and AI* 7. DOI: 10.3389/frobt.2020.00083.

Tatari, Farzaneh, Mohammad-R. Akbarzadeh-T, and Ahmad Sabahi (2012). "Fuzzy-probabilistic multi agent system for breast cancer risk assessment and insurance premium assignment". In: *Journal of Biomedical Informatics* 45.6, pp. 1021–1034. ISSN: 1532-0464. DOI: 10.1016/j.jbi.2012.05.004.

Taylor-King, Jake P. et al. (Mar. 2015). "Mathematical modelling of turning delays in swarm robotics". In: *IMA Journal of Applied Mathematics* 80.5, pp. 1454–1474. ISSN: 0272-4960. DOI: 10.1093/imamat/hxv001.

Treuille, Adrien, Seth Cooper, and Zoran Popović (July 2006). "Continuum Crowds". In: *ACM Transactions on Graphics* 25.3, pp. 1160–1168. ISSN: 0730-0301. DOI: 10.1145/1141911.1142008.

Varghese, Blesson and Gerard McKee (2010). "A mathematical model, implementation and study of a swarm system". In: *Robotics and Autonomous Systems* 58.3. Towards Autonomous Robotic Systems 2009: Intelligent, Autonomous Robotics in the UK, pp. 287–294. ISSN: 0921-8890. DOI: 10.1016/j.robot.2009.08.006.

Viswanath, D. V. Karthikeya and K. Madhava Krishna (Apr. 2009). "Towards Load-Balanced de-Congested Multi-Robotic Agent Traffic Control by Coordinated Control at Intersections". en. In: *Intelligent Service Robotics* 2.2, pp. 81–93. ISSN: 1861-2776, 1861-2784. DOI: 10.1007/s11370-009-0035-x.

Wang, Bin et al. (2022). "Spatiotemporally Actuated Hydrogel by Magnetic Swarm Nanorobotics". In: *ACS Nano* 16.12. PMID: 36469837, pp. 20985–21001. DOI: 10.1021/acsnano.2c08626.

Wen, Jinming, Li He, and Fumin Zhu (2018). "Swarm Robotics Control and Communications: Imminent Challenges for Next Generation Smart Logistics". In: *IEEE Communications Magazine* 56.7, pp. 102–107. DOI: 10.1109/MCOM.2018.1700544.

Wu, Mei-Min, Yu-Xiang Xiao, and Qian Bi (2020). "Software design of monitoring and flight simulation for UAV swarms based on OSGEarth". In: *International Journal of Computational Science and Engineering* 21.3, pp. 346–354. DOI: 10.1504/IJCSE.2020.106059.

Xia, Guoqing, Xianxin Sun, and Xiaoming Xia (2021). "Distributed Swarm Control Algorithm of Multiple Unmanned Surface Vehicles Based on Grouping Method". In: *Journal of Marine Science and Engineering* 9.12, p. 1324. ISSN: 2077-1312. DOI: 10.3390/jmse9121324.

Yan, Zhi, Nicolas Jouandeau, and Arab Ali Cherif (Dec. 2013). "A Survey and Analysis of Multi-Robot Coordination". In: *International Journal of Advanced Robotic Systems* 10. DOI: 10.5772/57313.

Yang, Bo, Chaofan Ma, and Xiaofang Xia (2021). "Drone Formation Control via Belief-Correlated Imitation Learning". In: *20th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*. AAMAS '21. Richland,

SC: International Foundation for Autonomous Agents and Multiagent Systems, pp. 1407–1415. ISBN: 9781450383073.

Yogi, Gondi Bala et al. (May 2020). "Swarm Robots for Warehouse Management". In: *International Journal of Research in Engineering, Science and Management* 3.5, pp. 706–709. ISSN: 2581-5792.

Yong, Ed (Mar. 2013). *How the Science of Swarms Can Help Us Fight Cancer and Predict the Future*. https://www.wired.com/2013/03/powers-of-swarms/. Accessed on 02 October 2022.

— (2014). *A Swarm of a Thousand Cooperative, Self-Organising Robots*. https://www.nationalgeographic.com/science/article/a-swarm-of-a-thousand-cooperative-self-organising-robots. Accessed on 30 January 2023.

Yoshimoto, Masahiro et al. (Dec. 2018). "Decentralized navigation method for a robotic swarm with nonhomogeneous abilities". In: *Autonomous Robots* 42.8, pp. 1583–1599. ISSN: 1573-7527. DOI: 10.1007/s10514-018-9774-x.

Zhou, Yuan et al. (July 2017). "Collision and Deadlock Avoidance in Multirobot Systems: A Distributed Approach". en. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47.7, pp. 1712–1726. ISSN: 2168-2216, 2168-2232. DOI: 10.1109/TSMC.2017.2670643.