

Article

Estimating the Expected Time to Enter and Leave a Common Target Area in Robotic Swarms

Yuri Tavares dos Passos ¹  and Leandro Soriano Marcolino ^{2,3,*} ¹ Centro de Ciências Exatas e Tecnológicas, Universidade Federal do Recôncavo da Bahia, Rua Rui Barbosa, 710, Centro, Cruz das Almas 44380-000, Brazil; yuri.passos@ufrb.edu.br² College of Engineering and Computer Science, VinUniversity, Vinhomes Ocean Park, Gia Lam Commune, Hanoi 131000, Vietnam³ School of Computing and Communications, Lancaster University, Bailrigg, Lancaster LA1 4WA, UK

* Correspondence: leandro.sm@vinuni.edu.vn or l.marcolino@lancaster.ac.uk

Abstract

Coordination algorithms are required to minimise congestion when every robot in a robotic swarm has a common target area to visit. Some of these algorithms use artificial potential fields to enable path planning to become distributed and local. An efficiency measure for comparing them is the time to complete a task in relation to the number of individuals in the swarm. To compare distinct solutions as the swarm grows, experiments with different numbers of robots must be simulated to form a plot of the function of the task completion time versus the number of robots or other parameters. Nevertheless, plotting it for many robots through simulation is time-consuming. Additionally, the inference of a global swarm behaviour as the task completion time from the local individual robot motion controller based on potential fields and other dynamical variables is intractable and requires experimental analysis. Based on that, equations are presented and compared with simulation data for estimating the expected task completion time of state-of-the-art algorithms, robots using only attractive and repulsive force fields and mixed teams for the common target area problem in robotic swarms with not only the number of robots as input but also environment- and algorithm-related global variables, such as the size of the common target area and the working area, average speed and average distance between the robots. This paper 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.

Keywords: robotic swarm; common target; time estimation; local-to-global analysis**MSC:** 93-10

Academic Editors: Fangfang Dong, Jiang Han and Xiaomin Zhao

Received: 2 September 2025

Revised: 23 October 2025

Accepted: 25 October 2025

Published: 5 November 2025

Citation: dos Passos, Y.T.; Marcolino, L.S. Estimating the Expected Time to Enter and Leave a Common Target Area in Robotic Swarms. *Mathematics* **2025**, *13*, 3552. <https://doi.org/10.3390/math13213552>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Motivation

Robotic swarms are systems formed by often simple robots with distributed processing and local sensing [1]. As it happens in insect swarms, no centralised coordination is performed for controlling each individual. Thus, distributed control and local sensing are assumed in robotic swarms [2,3].

As a way to reduce costs, robotic swarms use many simple and cheap robots instead of relying on a few complex and expensive ones [4]. Creating complex group behaviours based on simple rules is the goal of some robotic swarm research [1,5]. Counter-intuitively, such complex global behaviours may emerge from well-planned local rules [6,7]. Consequently,

swarm designers have to infer from local rules a global result needed to solve their problems. This inference may arise from several simulations of the local rules. However, when dealing with many robots, simulations could be time-consuming, and estimations of the resulting global outcome would be useful for swarm designers.

Having that in mind, this paper focuses on estimating a global outcome, the completion time of a task performed by some number of robots in a swarm. These estimations are based on the robot's local specification in a problem that may occur when numerous robots of that swarm have the same area: the common target problem [8]. This problem may occur, for instance, when robots are deployed to collect a leaked toxic substance. These robots need to coordinate themselves to reach 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.

Hence, solutions to the common target problem can also be applied in waste collection [9,10] in secluded areas such as the ocean, disaster response [11], construction [12], searching of cancer cells by microscopic robots [13–16], rescue in regions harmful to humans [17] and warehouse management [18,19]. In this problem, the number of robots may inevitably cause congestion as they are forced to share the same area. In addition, this number can impact the execution time, which affects other measures (e.g., heat generation and energy consumption [20]).

Thus, when devising different solutions for this problem, the typical practice is to compare the task completion time with the number of robots and environmental or algorithmic parameters (e.g., working area and target area sizes) [8,21–23]. For the common target problem, artificial potential fields are usually applied to such solutions in the same manner as in robotic swarms due to their distributed and local characteristic for path planning [24–26]. Additionally, other collision methods may not work well for this problem. For instance, Marcolino et al. [8] performed tests with the distributed collision avoidance method ORCA, which does not work for the common target problem because the robots enter a circular deadlock, not reaching the target area. The recent GBP Planning [27] requires the additional cost of using communication between robots, while the algorithms mentioned in this work do not need it.

That being said, 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. 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 or other parameters. 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.

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, environmental or algorithmic variables. Examples of dynamical variables are the average speed and the average distance between the robots. The variables about the environment in this paper are the radius of the target area and the radius of the task working area. Algorithmic variables are related to the parameters or execution of a specific algorithm being used by the robots, for example, how many possible directions the robots may go after they reach the target area. Unlike regressions, the estimations from this paper explicitly use these variables to estimate the task completion time with respect to the number of robots. As an advantage, there is no need to find by regression the best fitting function that relates how these variables behave with the task completion and the number of robots.

Moreover, another approach used in multi-agent research is the mean-field theory [28–33], which replaces the high number of interactions between the agents with a few averaged

ones. There are certain conditions to be satisfied for being able to apply this theory correctly [34], but no study has shown whether the common target problem satisfies those conditions. Also, it is unclear yet how mean-field-like trajectories from local controllers can be formally calculated to yield the global expected task completion time. Nevertheless, the usage of the mean interaction reduces the exactness of the estimation as the number of averaged objects increases. However, this does not occur in the geometric analysis performed in this paper, as it does not require a quantity of mean interactions explicitly.

Additionally, there are many situations where different swarm teams may be present in the same environment. For instance, such mixed teams (MT) can be found in the multi-agent literature [35] in problems like Pursuit-Evasion Games [36,37] and Multi-Agent Pickup and Delivery [38]. This work considers mixed teams where each group does not know the algorithm used by the other groups. This could happen, for example, in 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.

1.2. Objective

The objective of this paper is to introduce estimations of the expected task completion time according to the number of robots and environmental and algorithmic parameters in algorithms for the common target problem using potential fields. As these estimations are difficult to be directly expressed from the individual robot controller, they are based on analysing the observations made by simulating the control swarm algorithm. The equations of these estimations are mainly derived from geometric characteristics of the usual behaviour of the robots by following the control algorithm or obtained from well-known equations cited in the explanation of the estimation. No paper has presented such estimations for that problem before. Thus, this paper is a fundamental step to initiate a discussion about how the global system behaviour can be inferred from individual potential field-based local controllers in Swarm Robotics.

For doing so, extensive experiments were performed in the realistic Stage simulator [39] due to its ease of running in batch and parallel, low memory cost, fast execution for numerous robots and recent usage and review [40–42], assessing the completion time from hundreds of robots as quickly as possible to generate graphs and analyse different parameters. The proposed estimations are compared with data from experiments for a swarm with no coordination [21], state-of-the-art algorithms Single Queue Former and Touch and Run Vector Fields [23] and two groups in MT. Also, these algorithms are modelled in equation form to highlight the difficulty of directly deducing the task completion time from them. Experiments with them were made from 20 to 300 robots and different circular target area radii with holonomic and non-holonomic robots, to show the adaptability of the estimations to divergent types of robot movement. For MT, all combinations of those algorithms were tested to show the suitability of each algorithm for different ratios of the number of robots in a group to the total number of robots. The root mean squared error normalised by the y -axis range (NRMSE) values of the estimations are between 0.011 and 0.028 for a swarm without different groups and between 0.008 and 0.036 for two groups in MT.

1.3. Paper Organisation

This paper is organised as follows. The next section discusses the related work, including studies of local-to-global in science, the many-body problem and its application in Swarm Robotics and the robotic swarm common target problem and the algorithms to solve it.

Section 3 presents the task completion time estimations and how they were obtained for a swarm using only attractive and repulsive force fields, state-of-the-art algorithms and Mixed Teams.

Section 4 describes the experiments performed to corroborate the estimations derived in this work and discusses their results. These experiments aim to compare the estimations with the data obtained from simulations.

Section 5 discusses the limitations, challenges, improvements over this work and how the methodology can be applied to similar problems in Swarm Robotics.

Finally, Section 6 summarises the results and concludes this work.

2. Related Work

The effect of the task completion time of a robotic swarm based on the dynamics of one individual is an example of an analysis from local to global. Similar works can be found in other study areas. In particular, the many-body problem in Physics has elegant and strong mathematical solutions. Some works on traffic problems or Swarm Robotics have applied solutions for the many-body problem. However, the solutions presented are not adequate for the common target problem. The algorithms developed for the common target problem mostly use potential fields as a local path planner to achieve 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 [43–46] 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 [47] 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 concerning the interaction of microscopic particles are generally mentioned as the many-body problem [48,49]. For instance, in statistical mechanics, the Maxwell–Boltzmann distribution function [50,51] 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 probability frequencies to yield the desired distribution. There are different approaches to this problem. The perturbation theory [52,53] starts with a simpler version of the problem. Then, an expression, often called a perturbative expansion, using a 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 [54,55] for solving this problem numerically. Similarly, the density-functional theory [56,57] 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 [28] 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 et al. [29], Lasry and Lions [30], Ménéci [32], Lin et al. [33] and Zhang et al. [31]. The variational method [58,59] 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 [60] approximate the solution using machine learning to minimise the variational energy.

Similar to Many-Body in Swarm Robotics: Additionally, the literature on the many-body problem is mostly Physics-oriented. For instance, Mannone et al. [61] review specific quantum circuits that can be used to solve some problems of robotic swarm dynamics. An interesting approach to the intersection of Physics and Swarm Robotics is described by Hamann [62], where he applied Brownian motion, Langevin equation and the Fokker–Planck 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 [63–65], considering the drivers with intentions but random actions. However, as these approaches give results assuming that every action of every driver or particle is random as 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 [66] also presents another study on Swarm Robotics and Physics aligned with the problem in this paper, but the references provided by Elamvazhuthi and Berman [66] do not directly answer the research question shown here. 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 [39]. Additionally, these estimations are a fundamental first step to start a discussion on theoretical aspects of the relation between time and the number of robots on the common target problem.

The Robotic Swarm Common Target Area Problem and Algorithms: The common target problem is usually handled by using potential fields [8,21,23]. In fact, the algorithms discussed in Section 3 use them in the role of an individual path planner to accomplish global congestion control. Any of these algorithms modifies 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 [67], 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 the authors know, no work presents how to obtain this function from the motion equations of a robot, which is a challenge in Swarm Robotics. Thus, experimental approximations are presented in this paper 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 a minimal potential field.

Table 1 summarises the algorithms and approaches discussed in this section.

Table 1. Comparative summary of algorithms and approaches discussed in Section 2.

Approach and Reference(s)	Description	Applications/Limitations
Local-to-Global Studies		
Emergence [43–46]	Microscopic interactions give rise to unexpected macroscopic behaviours.	Biology, Chemistry, Physics; foundation for engineering and robotics.
Synergetics [47]	Explains self-organisation in non-equilibrium systems.	Cloud dissipation, lasers, slime mould, chemical reactions.

Table 1. *Cont.*

Approach and Reference(s)	Description	Applications/Limitations
Many-Body Problem Solutions		
Perturbation Theory [52,53]	Solves complex systems using successive corrections to a simpler problem.	Accurate for weak interactions; fails for strong couplings.
Coupled Cluster Method [54,55]	Numerical method from computational chemistry for many-body systems.	High accuracy; computationally expensive.
Density Functional Theory (DFT) [56,57]	Models molecular/electronic structure using electron density functionals.	Useful from atoms to solids; limited by functional choice.
Mean-Field Theory [28–33]	Replaces many interactions with averaged effects.	Reduces complexity; approximate; adapted in multi-agent systems.
Variational Method [58,59]	Uses trial functions to approximate minimal energy states.	Flexible but depends on chosen wave functions.
Swarm Robotics and Traffic Inspired Approaches		
Brownian Motion/Langevin/Fokker–Planck [62]	Estimate global behaviour from individual random motion.	Useful for theoretical modelling; limited for deterministic robot motion.
Herman–Prigogine Kinetic Equations [63–65]	Traffic flow equations with drivers as agents with intention and randomness.	Captures traffic dynamics; assumes random actions, not suited for deterministic swarms.
Common Target Problem Algorithms		
Potential Fields [8,21,23]	Local path planning via attractive and repulsive forces.	Effective for access/exit optimisation; still lacks theoretical minimum field.
Lagrangian-Inspired Methods [67]	Idea of minimising energy/time by analogy with mechanics.	Theoretical; requires a function for task time vs. number of robots, still unsolved.
This paper	Estimation of the task completion time in relation to the number of robots for different algorithms for the common target problem.	Can be applied to dynamical environments modelled by geometry; works better for about hundreds of robots.

3. Expected Task Completion Time Estimation Analysis

In this section, the estimation analysis is presented by describing the task scenario of the experiments and the individual robot controller and the challenge in the development of a global analysis from local controllers. This section also presents the common settings of the algorithms for which the expected task completion time estimations are analysed for a number of robots in the swarm.

Task Scenario: Consider the task scenario where a system with N robots in a plane must reach a common target located at $\mathbf{G} = (G_x, G_y)$. Each robot knows its position and the positions of the targets, and it can sense nearby objects (for instance, by using laser scanners covering 360 degrees). 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. Although the swarm density is not considered as an explicit parameter in this work, the number of robots N and the radius D of the working area yield the swarm density $N/(\pi D^2)$. If the robots have two or more targets, the targets are apart by at least $2D$. The initial position of the robots has a distance from the target centre randomly chosen from a uniform distribution in $[D, D + E)$ m for a fixed E . The angle between the initial position of a robot and the x -axis is selected from a uniform distribution in $[0, 2\pi)$ rad. To avoid robots starting in the same position and crashes between robots at the beginning of the simulation, if the distance between a robot and any other is less than 1 m, its initial position is selected again. This arrangement inside an annulus is only considered at the start of an experiment and before the initial target.

Once the robots have reached the first target area, they will proceed to the next one at a long distance, either to the left or right of the shared target. In other words, the next target is chosen randomly between positions $(-999999, G_y)$ and $(999999, G_y)$. Thus, approximately half of the robots will go to the left-hand side and the rest to the right, according to a uniform probability. This paper focuses on the congestion around the first target area, which is shared by all the robots in the swarm. After they leave the working area around the shared target area, they head towards the next target area by using only repulsive and attractive forces (that is, they turn off the coordination algorithm). In addition, the robots measure $0.44 \times 0.44 \times 0.44 \text{ m}^3$ in the experiments. Although some settings are precise (e.g., robot size and next target positions), the equations derived in this paper are based on general behaviours and are applicable with different parameters.

The global metric analysed in this paper from the local individual robot controller is defined below.

Definition 1. *The task completion time is obtained from the last robot to leave the working area around the shared target area, that is, the maximum time to enter the shared target area plus the time to leave the working area for every robot in the swarm.*

Once a robot leaves the working area after it reaches the first target area, it is no longer considered in the computation of any metric presented in this paper. As the initial position of the robots is random in an experiment and influences the variation of the task completion time, it is considered the expected task completion time from various experiments, which is defined below.

Definition 2. *Let t_{Alg} be the task completion time of an algorithm Alg using N robots for an experiment with the initial position of the N robots fixed. Then, $\mathbb{E}[t_{Alg}]$ stands for the expected task completion time over all executions for N robots.*

Individual Robot Controller: Consider a robot $i \in \{1, \dots, N\}$ at position $\mathbf{r}_i(t) = (r_{ix}(t), r_{iy}(t))$ at time t . In this work, each robot uses the same movement controller, chosen to physically describe the change in the position of the robot given by any algorithm that has an input force resultant from a potential force field $U(\mathbf{r}_i(t)) = U_A(\mathbf{r}_i(t)) + U_R(\mathbf{r}_i(t))$, for an attractive potential field $U_A(\mathbf{r}_i(t))$ and a repulsive potential field $U_R(\mathbf{r}_i(t))$. The input force has to be equal among the robots for all algorithms to perform a fair comparison among different approaches. Also, the controller needs to adapt the acceleration when the robot's velocity changes in situations such as the sudden advent of another robot or the transition to a new target. To satisfy these requirements, the controller is based on the

previous works on the common target problem [8,21,23] but rewritten in continuous form: $\dot{\mathbf{r}}_i(t) = \frac{d\mathbf{r}_i(t)}{dt}$, $\ddot{\mathbf{r}}_i(t) = \frac{d\dot{\mathbf{r}}_i(t)}{dt}$, and

$$\ddot{\mathbf{r}}_i(t) = K_{res} \frac{\mathbf{F}_A(\mathbf{r}_i(t)) + \sum_{\substack{j=1 \\ i \neq j}}^N \mathbf{F}_R(\mathbf{r}_i(t), \mathbf{r}_j(t))}{\left\| \mathbf{F}_A(\mathbf{r}_i(t)) + \sum_{\substack{j=1 \\ i \neq j}}^N \mathbf{F}_R(\mathbf{r}_i(t), \mathbf{r}_j(t)) \right\|} - K_{dp} \dot{\mathbf{r}}_i(t), \quad (1)$$

where $\mathbf{F}_A(\mathbf{r}_i(t)) = -\nabla U_A(\mathbf{r}_i(t))$ is a force to attract the robot toward a target, and $\mathbf{F}_R(\mathbf{r}_i(t)) = -\nabla U_R(\mathbf{r}_i(t))$ is the repulsive force applied to avoid bumping other robots. Without the assumption that the robot knows the target position, $\mathbf{F}_A(\mathbf{r}_i(t))$ cannot be computed correctly. $\mathbf{F}_A$ depends on the algorithm used by each robot and has a fixed magnitude K_a . The sections below show different choices for $\mathbf{F}_A$. Each robot adds the attractive force with the repulsive forces and then constrains 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 from slipping due to acceleration changes.

Also, the following repulsive force is employed because it acts as a strong barrier as the robot approaches an obstacle [68]:

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

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. I_d can be set to the maximum range of the robot's local sensor. Thus, a robot deprived of local object sensing cannot compute $\mathbf{F}_R(\mathbf{r}_i(t), \mathbf{r}_j(t))$.

Challenge in the Development of a Global Analysis from Local Controllars: With that $\mathbf{F}_R$, Equation (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, obtaining a closed-form expression for the task completion time in terms of N directly from Equation (1) is difficult. In addition, observing the behaviour of an algorithm for hundreds of robots in the real world is expensive, but it is faster and more affordable by simulation.

In other words, if the individual robot control equations were simpler than Equation (1), the closed-form expression of the expected task completion time in terms of N would be obtained by the following steps: (i) integrate Equation (1) twice to obtain $\mathbf{r}_i(t)$ for $i \in 1, \dots, N$, (ii) invert it to retrieve t for each robot i , (iii) obtain the maximum t for every robot i and (iv) calculate the expected value of this maximum when the initial positions of the robots are randomly distributed inside the annulus of radii D and $D + E$ and centre at $\mathbf{G}$. As this closed-form expression cannot be derived from Equation (1) in such a manner, a theoretical macroscopic analysis inspired by simulations is performed to attain an approximation of the expected task completion time $E[t]$. For this, besides the target area radius s and the working radius D , 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 $\bar{v}$. A summary of the variables most used henceforward is in Table 2.

Moreover, the approximations stated in this section may use these variables as input and are mainly derived from geometric observations or obtained from well-known equations. These approximations are explained after their statements, based on a macroscopic analysis observed from the simulations. As these approximations are constructed from

experimentation, these explanations cannot be considered formal proofs, although they are composed of proven mathematical expressions.

Table 2. Summary of the variables commonly used in this work.

Variable	Description	Variable	Description
$\bar{d}$	The mean distance between the centre of mass of a robot and the others inside the influence radius for all robots.	I_d	Default influence radius.
D	Radius of the circular working area.	K_a	Constant magnitude of an attractive force.
E	Maximum distance from the boundary of the working area for randomly positioning robots at the start of an experiment.	K_r	Multiplicative constant for the repulsive force.
$E[t_{Alg}]$	Expected task completion time of an algorithm Alg .	N	Total number of robots in the swarm.
$\mathbf{F}_A$	Attractive force vector.	s	Radius of the common circular target area.
$\mathbf{F}_R$	Repulsive force vector.	$\bar{v}$	Mean linear speed of all robots.
$\mathbf{G}$	the common target position		

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 4.

In more detail, let n be the number of points in a plotting, $i \in \{1, \dots, n\}$, N_i , the values of the number of robots in the x -axis and $\bar{t}_i$, the values of the mean simulation time in the y -axis obtained from experiments. Consider an equation for estimating the expected simulation time of the form $C_{Alg}e(N) + g$, where C_{Alg} is a constant for an algorithm Alg to be fitted, $e(N)$ is the part of the estimation equation depending solely on the number of robots N and g is any function independent of N . In this paper, the constants are fitted by minimising the root mean squared error normalised by the y -axis range (NRMSE). For that setting, the NRMSE is given by

$$\frac{\sqrt{\sum_{i=1}^n (C_{Alg}e(N_i) + g - \bar{t}_i)^2}}{\sqrt{n}(\bar{t}_{max} - \bar{t}_{min})},$$

where $\bar{t}_{min}$ and $\bar{t}_{max}$ are the minimum and maximum values of $\bar{t}_i$ for all $i \in \{1, \dots, n\}$. The C_{Alg} that minimises the NRMSE is obtained by equalising to zero the derivative of the NRMSE with respect to C_{Alg} , that is,

$$C_{Alg} = \frac{\sum_{i=1}^n e(N_i)(\bar{t}_i - g)}{\sum_{i=1}^n e(N_i)^2}. \quad (2)$$

Common Settings of the Presented Algorithms: In this section, four algorithms are analysed in terms of the expected task completion time. In the examples in the figures that follow and in the performed experiments, the values in Table 3 are used and the robots are

holonomic. Note, however, that the theoretical results also apply for arbitrary values and holonomic or non-holonomic robots, as shown in Section 4.

Table 3. Default values for parameters used in all algorithms.

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

In the following sections, the state-of-the-art algorithms in [23] 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. This presentation begins with the no coordination (NC) [21] (that is, 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.

3.1. No Coordination Algorithm

Algorithm Description: 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 (1) is the attractive force to the target centre $\mathbf{G}$ given by

$$\mathbf{F}_A(\mathbf{r}_i(t)) = \frac{K_a(\mathbf{G} - \mathbf{r}_i(t))}{\|\mathbf{G} - \mathbf{r}_i(t)\|}. \quad (3)$$

Although this is the simplest attractive force of all algorithms presented here, Equation (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.

Samples of the Observations: As an illustration, Figures 1 and 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 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 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 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 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 2a) until the last robot leaves this area (with a circle and an arrow in Figure 2b).

Figures 3 and 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 on both sides (Figure 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 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 4a) 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 following estimation presents the result. After its statement, an explanation shows the derivations of that approximation based on the observations made by simulating the algorithm. Because the following estimation is constructed from experimentation, the explanation below cannot be considered a formal proof.

The estimation below is the sum of three values. The first value represents the expected time for the first robot to arrive at the working area from its starting position. The second value is the expected time to go from the border of the working area to the target area and vice versa, plus the expected time for robots to fill the target area. The third value is a function that depends on N . It calculates the expected time for N robots to wait for the cluttered area around the target area to clear, which is linear if N is no more than the maximum number of robots that can fit inside the target area and proportional to $N^{1.5}$, otherwise.

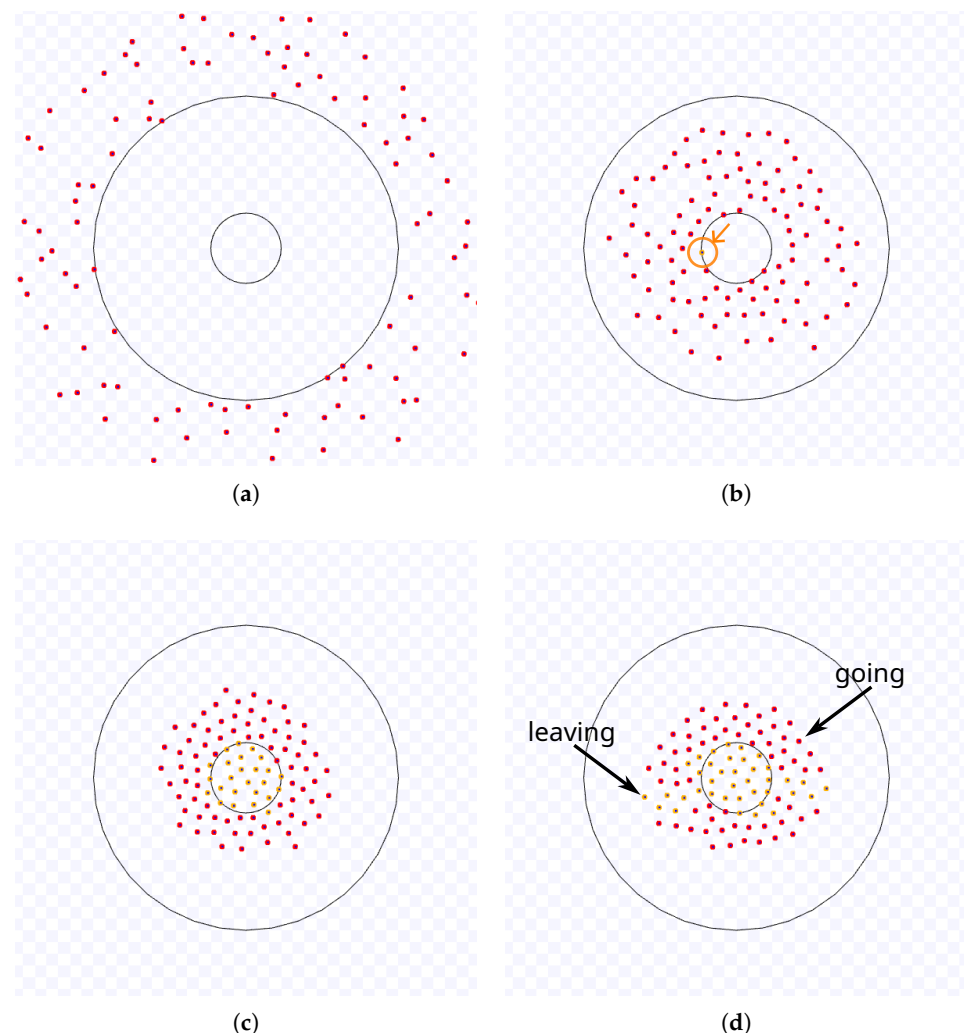


Figure 1. Example of an experiment of the NC algorithm with $N = 100$ and default values. Black arrows indicate examples of robots of the labelled state: (a) start of the experiment; (b) the first robot to reach the target area; (c) more robots reach the target region, fill this area and try to leave; (d) queues of robots formed to leave the cluttered area. Examples of robots leaving the target area (in yellow) and going to it (in red) are labelled. Full simulation available on <https://youtu.be/qgLxPk0OGyo>, accessed on 14 October 2022.

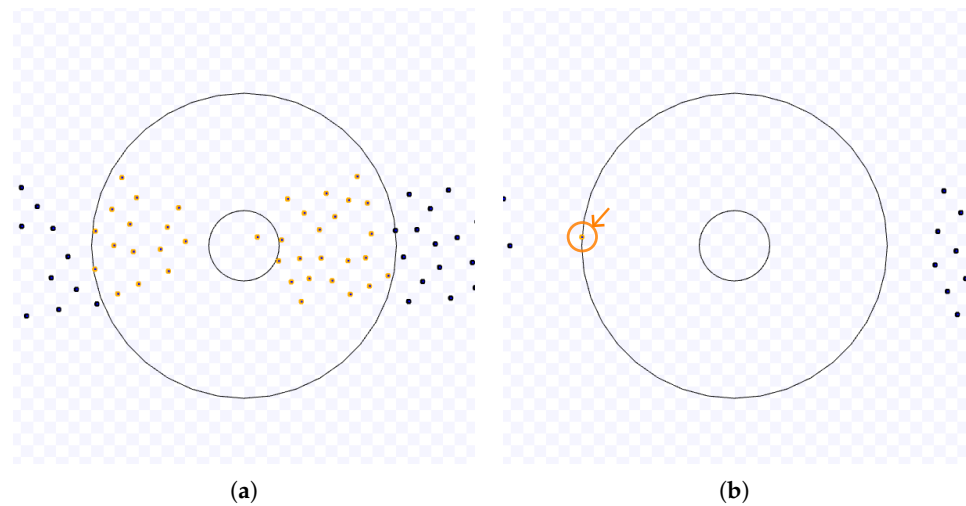


Figure 2. Continuation of Figure 1. (a) last queues near the end of the experiment; (b) the last robot to leave the working area.

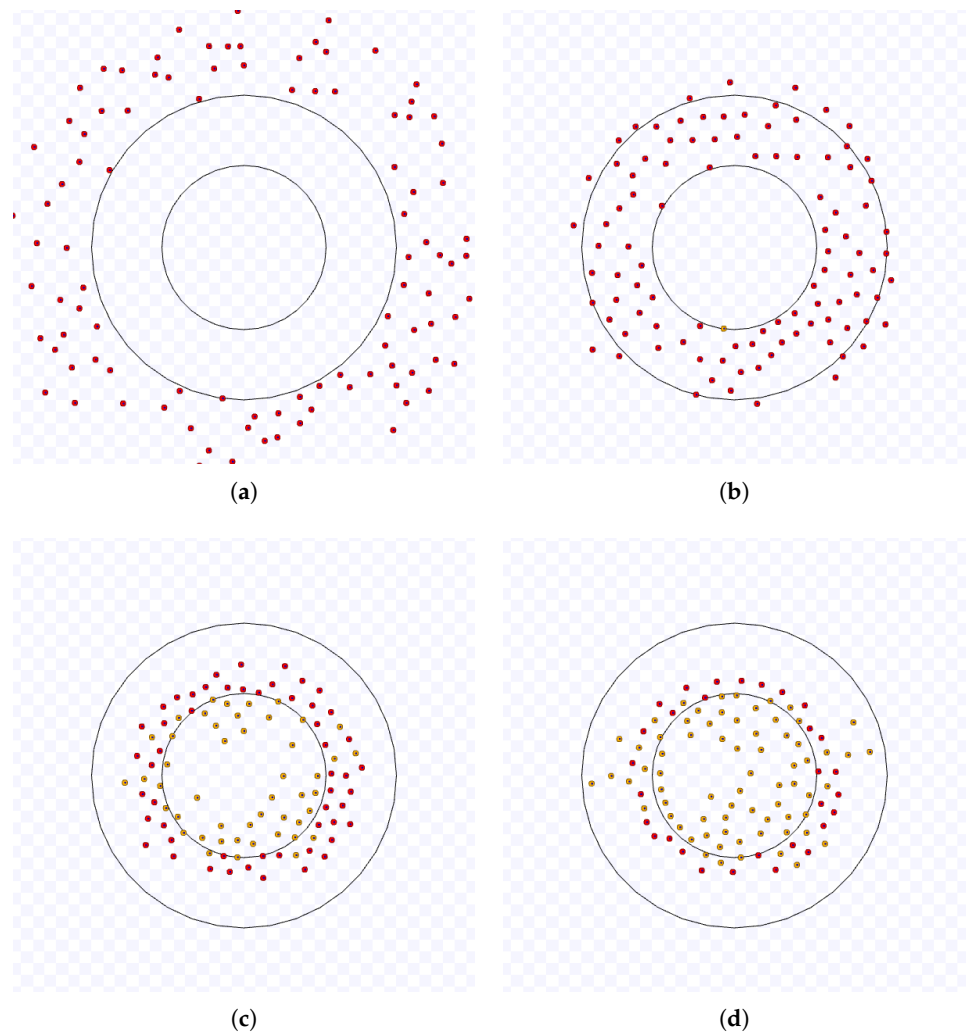


Figure 3. Example of an experiment of the NC algorithm with $N = 100$ and default values except for $s = 7$ m: (a) start of the experiment; (b) the first robot to reach the target area; (c) free space in the middle of the target area; (d) a few robots going to the target area and more space inside the target area. Full simulation available on <https://youtu.be/x0oO5DuAtwQ>, accessed on 12 May 2023.

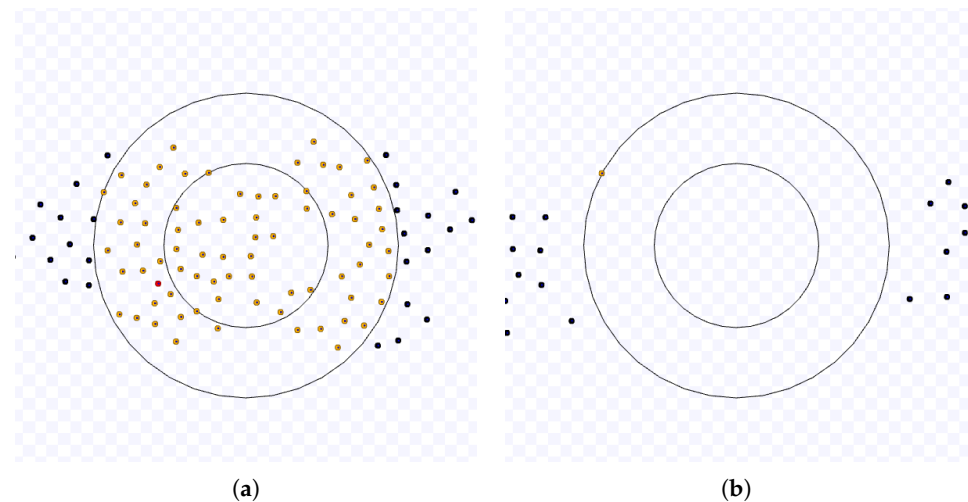


Figure 4. Continuation of Figure 3: (a) robots leave the target area with less resistance; (b) end of the experiment.

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

$$\mathbb{E}[t_{NC}] \approx \frac{E}{\bar{v}(N+1)} + \frac{2D-s}{\bar{v}} + \text{wait}_1(N), \quad (4)$$

for

$$\text{wait}_1(N) = \begin{cases} C_{NC1}N, & \text{if } N \leq \frac{2(s+\bar{d}/2)^2}{(\bar{d}/2)^2} \\ C_{NC2}\left(\frac{N^{1.5}}{1.5\sqrt{\pi}} - N\right), & \text{otherwise} \end{cases}$$

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 position, (b) the first robot that entered the working area reaching the target area (e.g., the time at Figure 1b), (c) the first robots filling the target area (e.g., the time from Figure 1b,c), (d) the cluttered area disappearing by the queues of robots leaving the target (e.g., the time of Figures 1c,d, 2a, 3c,d and 4a) and (e) the last robot that reached the target leaving the working area (e.g., the time from Figure 2a,b).

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 [69] (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. Let the capacity of the target area be the maximum number of robots that can fit inside the target area. When the capacity of the target area is greater than or equal to the number of robots outside the target area, the robots that arrive at the target area have less resistance to leave through the space

between the robots that have not yet left. The capacity of the target 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, $\frac{(s+\bar{d}/2)^2}{(\bar{d}/2)^2}$. Thus, the number of robots exceeding the capacity of the target area is less than or equal to its capacity if $N - \frac{(s+\bar{d}/2)^2}{(\bar{d}/2)^2} \leq \frac{(s+\bar{d}/2)^2}{(\bar{d}/2)^2} \Leftrightarrow N \leq \frac{2(s+\bar{d}/2)^2}{(\bar{d}/2)^2}$. From experiments, the waiting time can be approximated by $C_{NC1}N$ for a constant C_{NC1} that abstracts the influence of other factors at 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 on the overall movement by the type of the robot—holonomic or non-holonomic).

If the maximum number of robots that fit inside the target area is less than the number of robots outside the target area, 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 the 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 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 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 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 $r(t)$.

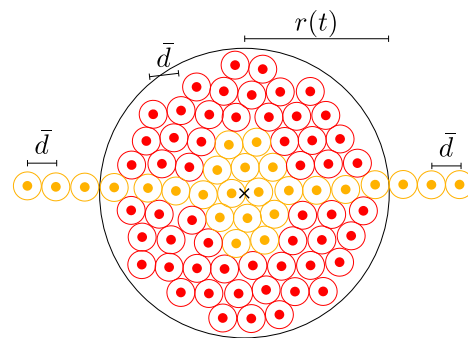


Figure 5. Number of robots at a given time, $N(t)$, occupying a circle of radius r with robots distant by each other by $\bar{d}$. Red and yellow dots and circles represent robots going to and leaving the target area, respectively.

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 $\bar{d}^2 \text{ m}^2$. 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

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

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) and simplifying the final result. $\square$

3.2. Single Queue Former Algorithm

Algorithm Description: The Single Queue Former (SQF) algorithm [23] 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 (Figure 6). The robots are permitted to enter the target region only through this queue. Potential fields are deployed to form this queue (Figure 6a) and to guide the robots to the target area exit (Figure 6b) 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 $(G_x + s, G_y)$.

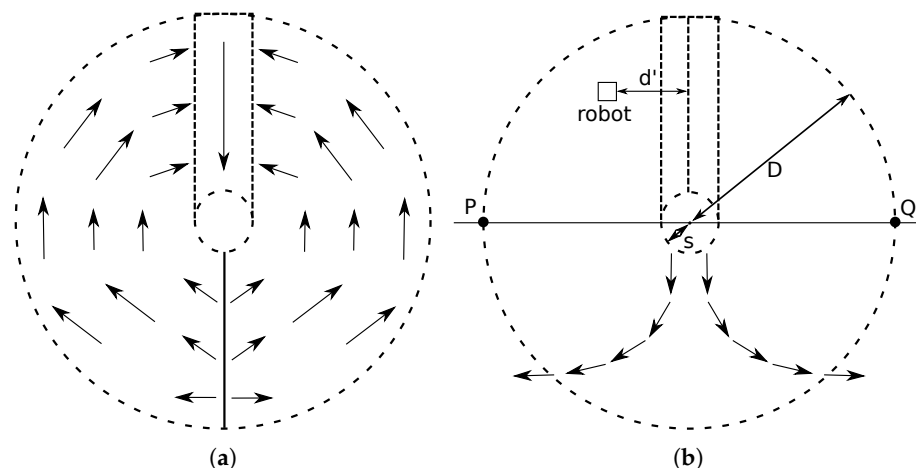


Figure 6. (a) Rotational force field to reach the target with the Single Queue Former algorithm [23]; (b) rotational force field to leave the target in the SQF algorithm, and the distance d' from the central line of the corridor to the robot for varying its influence radius [23].

A rotational force field is applied to robots in the working area 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:

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

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 does. Thus, the states of the robot can be represented as formulas using these conditions, which are mutually exclusive:

$$\begin{aligned} GT &= \neg W \vee \neg O \wedge \neg A, \\ GC &= W \wedge O \wedge \neg A, \\ LT &= W \wedge A. \end{aligned}$$

By using these conditions, the attractive force is represented as follows. Robots outside the corridor follow a force according to Passos et al. [23]

$$\mathbf{F}_A(\mathbf{r}_i(t)) = \begin{cases} K_a \frac{(r_{iy}(t) - G_y, -r_{ix}(t) + G_x)}{\|\mathbf{r}_i(t) - \mathbf{G}\|}, & \text{if } GC \wedge r_{ix}(t) - G_x \leq 0, \\ K_a \frac{(-r_{iy}(t) + G_y, r_{ix}(t) - G_x)}{\|\mathbf{r}_i(t) - \mathbf{G}\|}, & \text{if } GC \wedge r_{ix}(t) - G_x > 0, \end{cases} \quad (5)$$

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 (3), that is,

$$\mathbf{F}_A(\mathbf{r}_i(t)) = \frac{K_a(\mathbf{G} - \mathbf{r}_i(t))}{\|\mathbf{G} - \mathbf{r}_i(t)\|}, \text{ if } GT. \quad (6)$$

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

$$\mathbf{F}_A(\mathbf{r}_i(t)) = \begin{cases} K_a \frac{(r_{iy}(t) - P_y, -r_{ix}(t) + P_x)}{\|\mathbf{r}_i(t) - \mathbf{P}\|}, & \text{if } LT \wedge G'_x \leq G_x, \\ K_a \frac{(-r_{iy}(t) + Q_y, r_{ix}(t) - Q_x)}{\|\mathbf{r}_i(t) - \mathbf{Q}\|}, & \text{if } LT \wedge G'_x > G_x, \end{cases} \quad (7)$$

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

(Figure 6b). 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

$$I(\mathbf{r}_i(t)) = \begin{cases} I_{min}, & \text{if } W \wedge (GT \vee LT), \\ I_{min} + |r_{ix}(t) - G_x|, & \text{if } GC \wedge r_{iy}(t) > G_y \wedge |r_{ix}(t) - G_x| < I_d - I_{min}, \\ I_d, & \text{otherwise,} \end{cases} \quad (8)$$

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

Observe that plugging those attractive and repulsive forces in Equations (5)–(8) into the movement controller (Equation (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 (8) 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.

Samples of the Observations: As an example, Figures 7 and 8 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 7a. The robots go in the direction of the target and change to state *going to the corridor* as they reach the working area (Figure 7b). 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 7c). 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 7d). After the last robot reaches the target area (marked with a circle and an arrow in Figure 8a), all robots follow the exit force field and are on different sides depending on their next target (Figure 8b).

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 9 and 10 present an example of such a situation. Figure 9a illustrates the initial configuration. Figure 9b presents the first robot to reach the target area through the corridor (marked with a circle and an arrow). Figure 9c 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 9d), 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 from going to the corridor, thus increasing their time to leave (Figure 10a). Figure 10b 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 10b were the last to overcome the robots exiting through the leaving area.

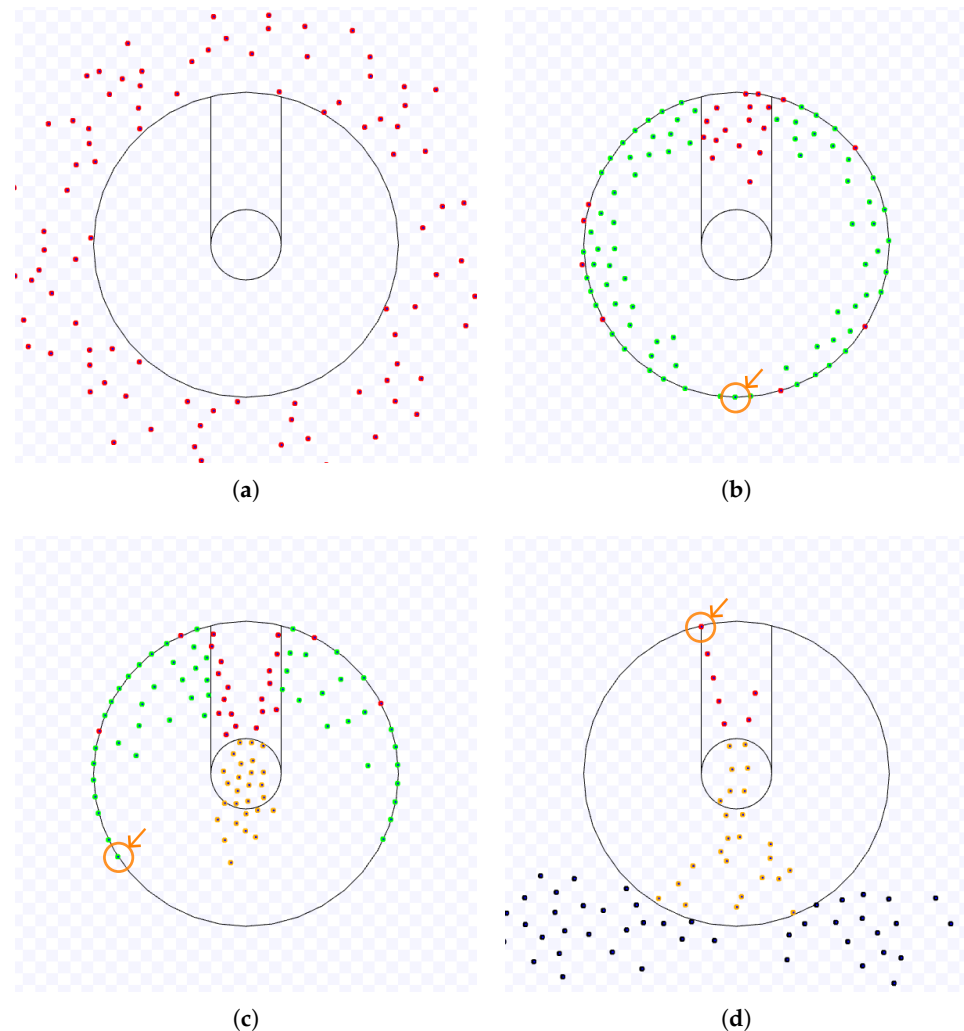


Figure 7. Example of an experiment of the SQF algorithm with default values and $N = 100$. Black arrows indicate examples of robots of the labelled state: (a) start of the experiment; (b) bottom-most robot to reach the working area; (c) robots waiting on both sides of the corridor; (d) the last robot to reach the corridor. Full simulation available on <https://youtu.be/EdkCnUVw4Bg>, accessed on 24 March 2023.

As before, an approximation of the expected task completion time for the SQF and its explanation are presented as follows. The estimation below is the sum of five values. The first value is the expected time for the last robot to leave the working area from the target area. The second value is the expected time for the last robot to go from the topmost edge of the corridor to the target area. The third value is the expected time for the bottom-most robot travelling from where it first entered the working area border to the corridor. The fourth value is the expected time for the bottom-most robot to go from its starting position to the working area. The fifth value is a function that depends on N . It calculates the expected time for N robots would wait to access the corridor. It is linear if N is no more than the approximated maximum number of robots waiting to enter the target area and quadratic, otherwise.

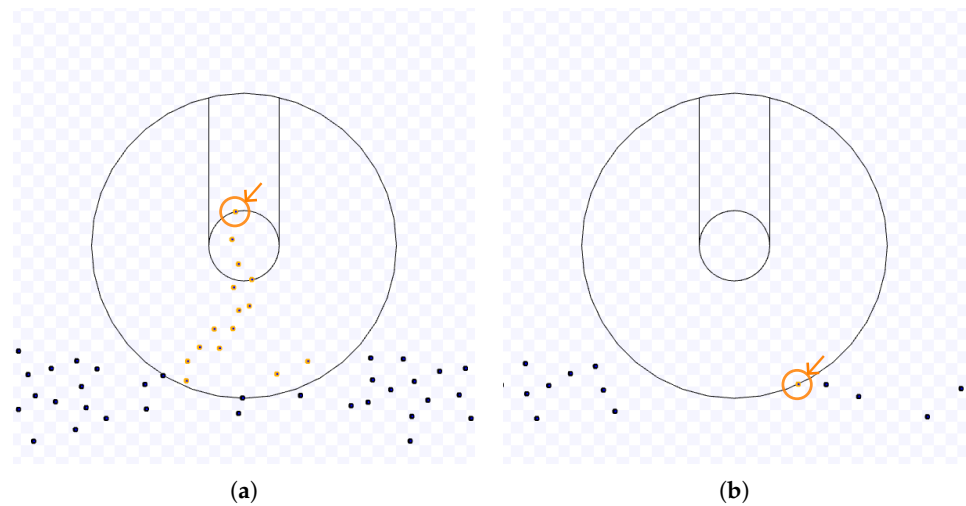


Figure 8. Continuation of Figure 7: (a) the last robot to reach the target area; (b) the last robot to leave the working area.

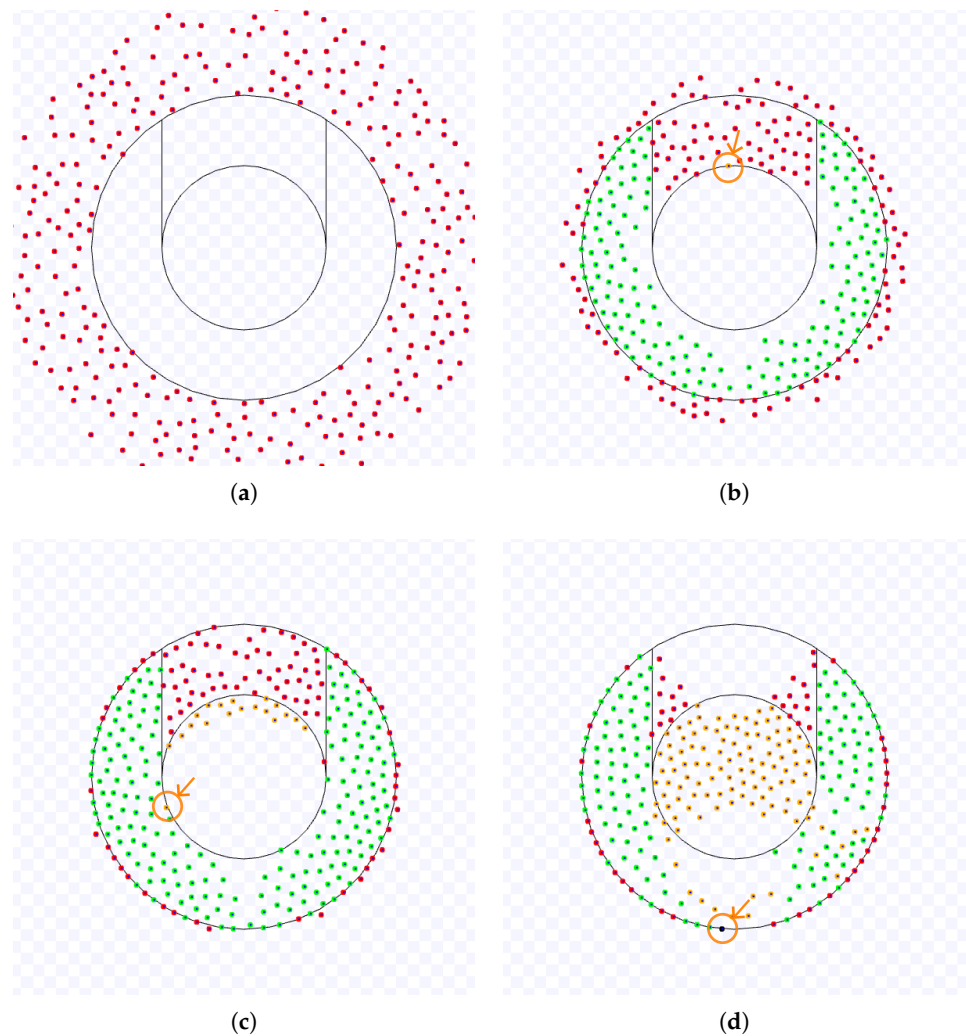


Figure 9. Example of an experiment of the SQF algorithm with default values except for $s = 7$ m and $N = 300$: (a) start of the experiment; (b) the first robot to enter the target area by going to the corridor; (c) the first robot to leave the target area without passing through the corridor; (d) the first robot to leave the working area. Full simulation available on <https://youtu.be/pEQzg0JwPbQ>, accessed on 19 April 2023.

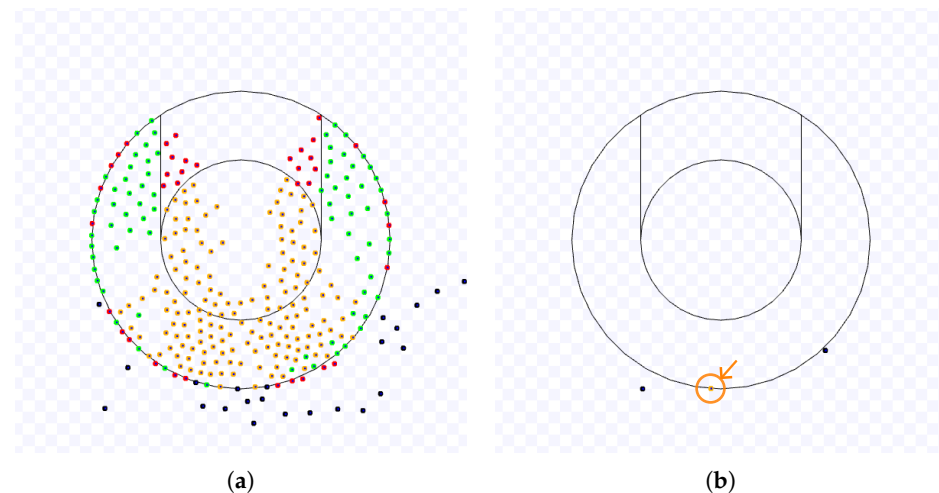


Figure 10. Continuation of Figure 9: (a) robots create a barrier obstructing others from leaving the working area; (b) the last robot to leave the working area.

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

$$\mathbb{E}[t_{SQF}] \approx \frac{\sqrt{D^2 - s^2}}{\bar{v}} \left(\arcsin\left(\frac{s}{D}\right) + \arccos\left(\frac{\sqrt{D^2 - s^2}}{2D}\right) \right) + \frac{D - s}{\bar{v}} + \frac{N(\pi - \arcsin(s/D))D}{(N + 2)\bar{v}} + \frac{NE}{\bar{v}(N + 1)} + wait_2(N), \quad (9)$$

for

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

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 8a,b), (b) the last robot going from the topmost edge of the corridor to the target area (e.g., from Figures 7d and 8a), (c) robots waiting to access the corridor (e.g., Figures 7c, 9d and 10a), (d) the bottom-most robot travelling from where it first entered the working area border to the corridor (e.g., the time from Figure 7b–d but excluding the waiting time (c)), (e) the bottom-most robot going from its starting position to the working area (e.g., Figure 7a,b).

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 11a, which shows this arc in a 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 11b), the target centre located at G and the point at P (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.) 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 $\mathbf{GPA}_2$. 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 11b) 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 Figures 9d and 10a,b). The entry area capacity is calculated from the part of the working 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 9d and 10a). 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 11c. 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.

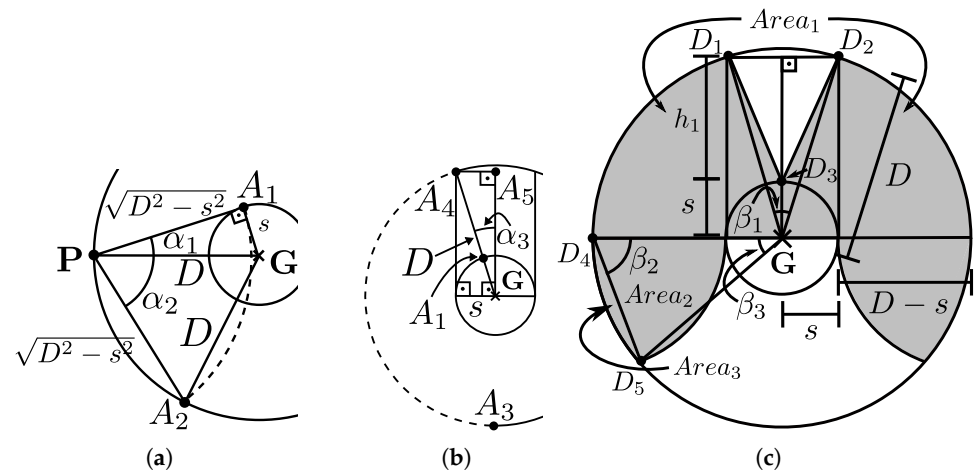


Figure 11. (a) Dashed arc length from the expected target arriving location at A_1 to the working area border at A_2 ; (b) dashed arc length from the expected position of the bottom-most robot A_3 to the corridor corner A_4 ; (c) area considered for entry capacity for the SQF algorithm.

$Area_1$ 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 $\mathbf{GD}_1D_2$ —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 $\mathbf{GD}_1D_2$ (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 $\triangle \mathbf{GD}_1D_2$ 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, $Area_1 = \frac{\pi}{2}D^2 - s(\sqrt{D^2 - s^2} - s) - D^2\arcsin(s/D) + s\sqrt{D^2 - s^2} = D^2(\frac{\pi}{2} - \arcsin(\frac{s}{D})) + s^2 = D^2\arccos(\frac{s}{D}) + s^2$.

$Area_2$ 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 $\mathbf{GD}_4D_5$ 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 GD_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 $\bar{d}^2$ m² inside the entry area: $\frac{1}{\bar{d}^2}(Area_1 + 2(Area_2 + Area_3)) = \frac{1}{2\bar{d}^2}(2D^2 \arccos\left(\frac{s}{D}\right) + 2s^2 + 2(D-s)^2 \arccos\left(\frac{D-s}{2D}\right) + 4D^2 \arcsin\left(\frac{D-s}{2D}\right) - (D-s)\sqrt{4D^2 - (D-s)^2})$.

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 11b). In Figure 11b, 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., $\frac{\frac{N}{2}(\pi - \alpha_3)}{\frac{N}{2} + 1} = \frac{N(\pi - \alpha_3)}{N + 2}$ [69] (p. 229). From the right triangle formed by A_4, A_5 and G in Figure 11b, $\alpha_3 = \arcsin(s/D)$. 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). $\square$

3.3. Touch and Run Vector Fields Algorithm

Algorithm Description: The Touch and Run Vector Fields (TRVF) algorithm [23] 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* (Figure 12). 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 a sector called the central angle region.

Consider a central angle region $z = \lfloor \frac{\eta}{\alpha} \rfloor + 1$, 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 (Figure 12).

Thus [23],

$$\begin{aligned}\mathbf{w}_1 &= \mathbf{G} + D(\cos(z\alpha), \sin(z\alpha)) + \frac{d}{2}(\sin(z\alpha), -\cos(z\alpha)); \\ \mathbf{w}_2 &= \mathbf{G} + \sqrt{(r+s)^2 - (r+d/2)^2}(\cos(z\alpha), \sin(z\alpha)) + \\ &\quad \frac{d}{2}(\sin(z\alpha), -\cos(z\alpha));\end{aligned}$$

$$\begin{aligned} \mathbf{w}_3 &= \mathbf{G} + \sqrt{(r+s)^2 - (r+d/2)^2}(\cos((z-1)\alpha), \\ &\sin((z-1)\alpha)) + \frac{d}{2}(\sin((z-1)\alpha), -\cos((z-1)\alpha)); \text{ and} \\ \mathbf{w}_4 &= \mathbf{G} + D(\cos((z-1)\alpha), \sin((z-1)\alpha)) + \\ &\frac{d}{2}(\sin((z-1)\alpha), -\cos((z-1)\alpha)) \end{aligned}$$

for the radius r of the circular path from $\mathbf{w}_2$ to $\mathbf{w}_3$. Due to Passos et al. [22] (Lemma 3), $r = \frac{s \sin(\alpha/2) - d/2}{1 - \sin(\alpha/2)}$. This path has centre located at $\mathbf{c} = \mathbf{G} + (r+s)(\cos((z-\frac{1}{2})\alpha), \sin((z-\frac{1}{2})\alpha))$ [23].

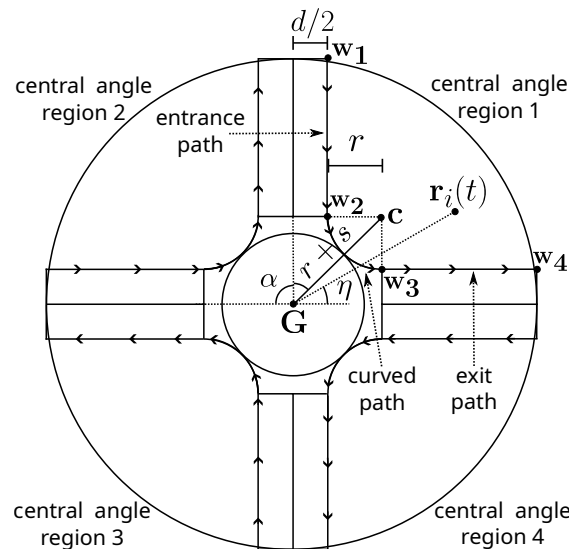


Figure 12. Example of the lanes in TRVF algorithm with $K = 4$. The robots follow the waypoints $\mathbf{w}_1$, $\mathbf{w}_2$, $\mathbf{w}_3$ and $\mathbf{w}_4$ depending on the central angle region where they are located. The centre $\mathbf{c}$ for the curved trajectory between waypoints $\mathbf{w}_2$ and $\mathbf{w}_3$ is distant by $r + s$ from the target centre $\mathbf{G}$.

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. [70])

$$\begin{aligned} \zeta_l &= \begin{cases} \zeta_f - \frac{\pi}{2}\rho, & \text{if } \epsilon > \frac{d}{5}, \\ \zeta_f - \left(\frac{k_s \frac{\pi}{2} v}{k_\omega \left(\frac{d}{5} \right)^{k_s}} \right) e^{k_s - 1} \sin(\zeta) - \frac{\pi}{2} \left(\frac{\epsilon}{\frac{d}{5}} \right)^{k_s}, & \text{otherwise,} \end{cases} \\ \mathbf{F}_L(\mathbf{r}_i(t), \mathbf{w}_i, \mathbf{w}_f) &= \begin{cases} K_a(\cos(\zeta_l), \sin(\zeta_l)), & \text{if } \frac{(\mathbf{r}_i(t) - \mathbf{w}_i) \cdot \mathbf{w}_f}{\|\mathbf{w}_f\|^2} \geq 1, \\ 0, & \text{otherwise,} \end{cases} \end{aligned} \quad (10)$$

for $\mathbf{w}_f = \mathbf{w}_f - \mathbf{w}_i$, $\zeta_f = \text{atan2}(w_{fi,y}, w_{fi,x})$, $\epsilon = \left\| \mathbf{r}_i(t) - \mathbf{w}_i - \frac{(\mathbf{r}_i(t) - \mathbf{w}_i) \cdot \mathbf{w}_f}{\|\mathbf{w}_f\|^2} \mathbf{w}_f \right\|$, $\rho = \text{sign}(\mathbf{w}_f \times (\mathbf{r}_i(t) - \mathbf{w}_i))$, a constant for setting the force magnitude K_a , the current orientation of the robot ζ , 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. [70])

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

$$\mathbf{F}_O(\mathbf{r}_i(t), \mathbf{C}, R, \mathbf{w}_f) = \begin{cases} K_a (\cos(\frac{\pi}{2} - \zeta_c), \sin(\frac{\pi}{2} - \zeta_c)), & \text{if } \mathbf{q} \times (\mathbf{w}_f - \mathbf{C}) \leq 0, \\ 0, & \text{otherwise,} \end{cases} \quad (11)$$

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 curved path (ECP)*, *leaving through the curved 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:

$$\begin{aligned} W &= \|\mathbf{r}_i(t) - \mathbf{G}\| \leq D, \\ A &= \exists t' \leq t : \|\mathbf{r}_i(t') - \mathbf{G}\| \leq s, \\ B &= \exists t' \leq t : (\mathbf{r}_i(t') - \mathbf{G}) \times (\mathbf{w}_1 - \mathbf{G}) \leq 0, \\ EN &= \exists t' \leq t : \frac{(\mathbf{r}_i(t') - \mathbf{w}_1) \cdot (\mathbf{w}_2 - \mathbf{w}_1)}{\|\mathbf{w}_2 - \mathbf{w}_1\|^2} \geq 1, \\ TA &= \exists t' \geq t_A : (\mathbf{r}_i(t') - \mathbf{c}) \times (\mathbf{w}_3 - \mathbf{c}) \leq 0. \end{aligned}$$

W and A mean the same as in SQF in the previous section. B , EN and TA , respectively, mean 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 the 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:

$$\begin{aligned} GT &= \neg W \\ GEP &= W \wedge \neg A \wedge \neg B \wedge \neg EN \wedge \neg TA, \\ OEN &= W \wedge \neg A \wedge B \wedge \neg EN \wedge \neg TA, \\ ECP &= W \wedge \neg A \wedge B \wedge EN \wedge \neg TA, \\ LCP &= W \wedge A \wedge B \wedge EN \wedge \neg TA, \\ OEX &= W \wedge A \wedge B \wedge EN \wedge TA. \end{aligned}$$

Using them and Equations (10) and (11) 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 :

$$\begin{aligned} \mathbf{F}_{A1}(\mathbf{r}_i(t)) &= \frac{K_a(\mathbf{G} - \mathbf{r}_i(t))}{\|\mathbf{G} - \mathbf{r}_i(t)\|}, \\ \mathbf{F}_{TR}(\mathbf{r}_i(t)) &= \begin{cases} K_r \left(\frac{1}{\|\mathbf{G}' - \mathbf{r}_i(t)\|} - \frac{1}{D} \right) \frac{\mathbf{G}' - \mathbf{r}_i(t)}{\|\mathbf{G}' - \mathbf{r}_i(t)\|^3}, & \text{if } \|\mathbf{G}' - \mathbf{r}_i(t)\| - D \leq D, \\ 0, & \text{otherwise,} \end{cases} \end{aligned}$$

$$\mathbf{F}_A(\mathbf{r}_i(t)) = K_a \frac{\mathbf{F}_{A1}(\mathbf{r}_i(t)) + \mathbf{F}_{TR}(\mathbf{r}_i(t))}{\|\mathbf{F}_{A1}(\mathbf{r}_i(t)) + \mathbf{F}_{TR}(\mathbf{r}_i(t))\|}, \text{ if } GT. \quad (12)$$

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

$$\mathbf{F}_A(\mathbf{r}_i(t)) = \begin{cases} \mathbf{F}_L(\mathbf{r}_i(t), \mathbf{w}_1, \mathbf{w}_2), & \text{if } OEN, \\ \mathbf{F}_L(\mathbf{r}_i(t), \mathbf{w}_3, \mathbf{w}_4), & \text{if } OEX. \end{cases} \quad (13)$$

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

$$\mathbf{F}_A(\mathbf{r}_i(t)) = \mathbf{F}_O(\mathbf{r}_i(t), \mathbf{G}, D, \mathbf{w}_1), \text{ if } GEP. \quad (14)$$

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 :

$$\begin{aligned} \mathbf{F}_{A2}(\mathbf{r}_i(t)) &= 1.5 \frac{K_a(\mathbf{G} - \mathbf{r}_i(t))}{\|\mathbf{G} - \mathbf{r}_i(t)\|}, \\ \mathbf{F}_{A3}(\mathbf{r}_i(t)) &= 1.5 \frac{K_a(\mathbf{w}_3 - \mathbf{r}_i(t))}{\|\mathbf{w}_3 - \mathbf{r}_i(t)\|}, \\ \mathbf{F}_A(\mathbf{r}_i(t)) &= \begin{cases} K_a \frac{\mathbf{F}_{A2}(\mathbf{r}_i(t)) + \mathbf{F}_O(\mathbf{r}_i(t), \mathbf{c}, r, \mathbf{w}_3)}{\|\mathbf{F}_{A2}(\mathbf{r}_i(t)) + \mathbf{F}_O(\mathbf{r}_i(t), \mathbf{c}, r, \mathbf{w}_3)\|}, & \text{if } ECP, \\ K_a \frac{\mathbf{F}_{A3}(\mathbf{r}_i(t)) + \mathbf{F}_O(\mathbf{r}_i(t), \mathbf{c}, r, \mathbf{w}_3)}{\|\mathbf{F}_{A3}(\mathbf{r}_i(t)) + \mathbf{F}_O(\mathbf{r}_i(t), \mathbf{c}, r, \mathbf{w}_3)\|}, & \text{if } LCP. \end{cases} \end{aligned} \quad (15)$$

Again, the controller in Equation (1) with the attractive force described by Equations (12)–(15) 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.

Sample of the Observations: As an example, Figures 13 and 14 present an execution of the TRVF algorithm with its default values (Table 4). 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 13a illustrates the 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 13b). Figure 13c 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 at the bottom of Figure 13d). After that wait, the last robot to access the working area proceeds through the entrance path in the lane, arrives in the target area (with a circle and an arrow in Figure 14a) and leaves by the exit path, facing a few robots going in different directions (Figure 14b).

Table 4. Default values for the parameters used for TRVF algorithm.

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

From such examples, an approximation of the expected task completion time for the TRVF and its explanation are shown below. The estimation below is the sum of three values. The first value is the expected time for the first robot, starting from its initial position, to reach an entrance path without moving across the border of the working area circular sector. The second value is the expected time for the first robot to move to the target area and then leave it until reaching the working area border by one of the lanes. The third value is the expected time to wait and travel through one of the lanes, which is linear in N .

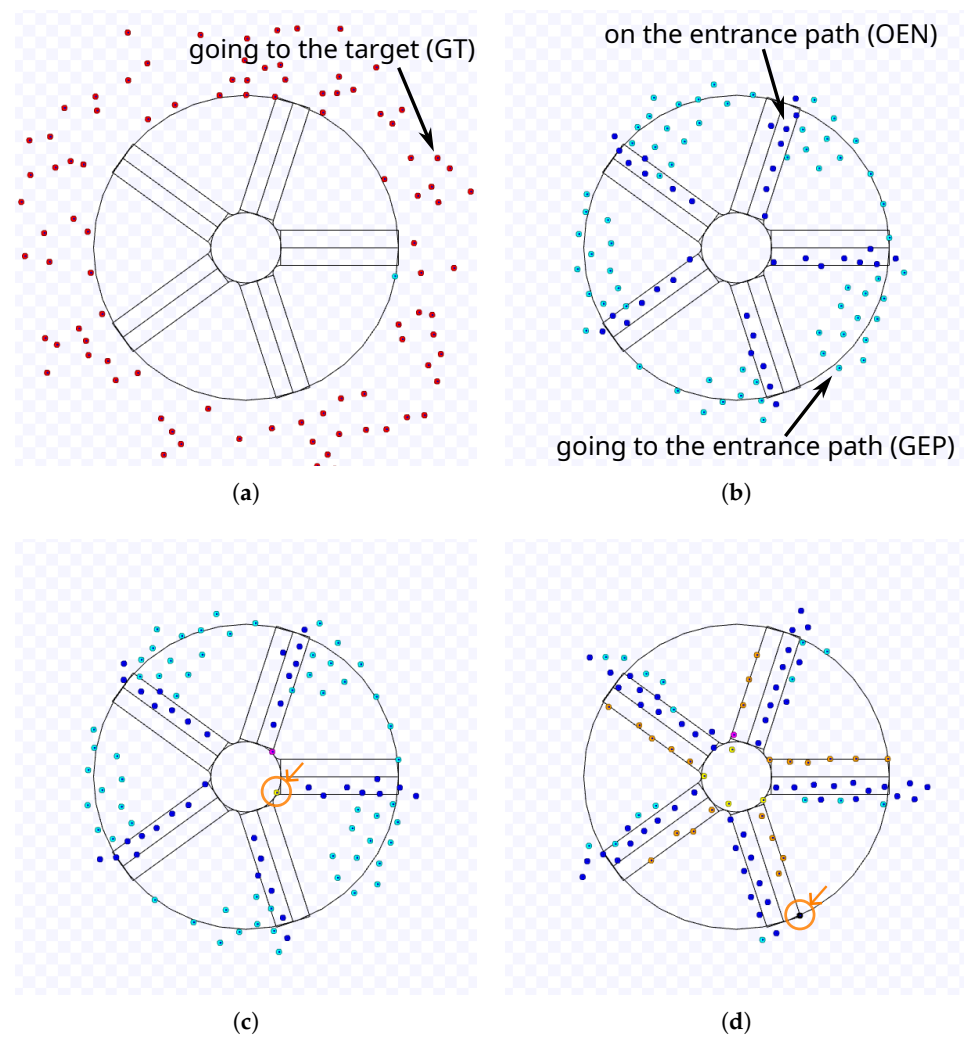


Figure 13. Example of an experiment of the TRVF algorithm with default values and $N = 100$. Black arrows indicate examples of robots of the labelled state: (a) start of the experiment; (b) robots go through the entrance path; (c) the first robot to reach the target area; (d) the first robot to leave the working area. Full simulation available on https://youtu.be/bNlgvgpl_J4, accessed on 14 October 2022.

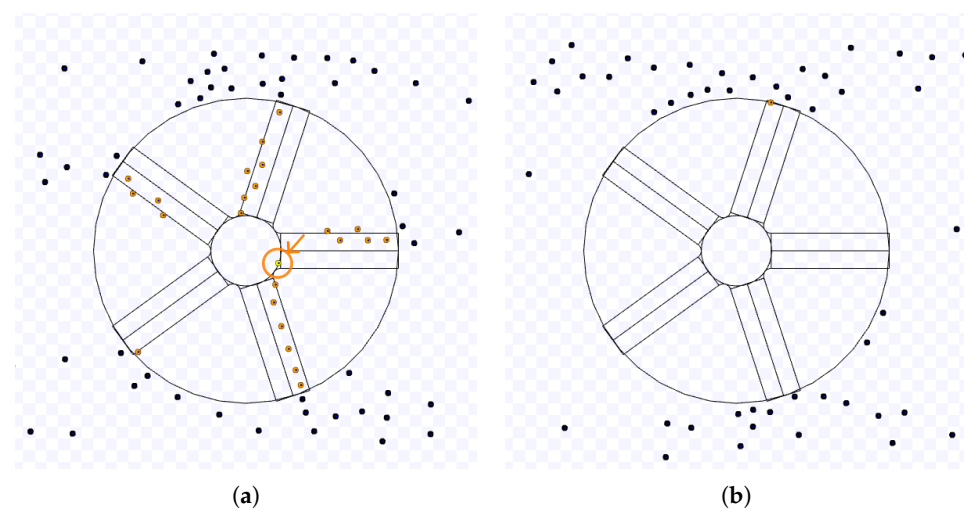


Figure 14. Continuation of Figure 13: (a) the last robot to reach the target area; (b) end of the experiment.

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

$$\mathbb{E}[t_{TRVF}] \approx \frac{E}{\left(\frac{\arctan(\frac{d}{2D})}{2\pi}N + 1\right)\bar{v}} + \frac{2d_s + r(\pi - \alpha)}{\bar{v}} + C_{TRVF} \frac{N}{K}, \quad (16)$$

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. 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., Figure 13b–d), (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 working area border until it arrives at the entrance path, which is more distant, as the cyan robots in Figure 13b–d. From Figure 15a, 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 [69] (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 15b 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 15b is calculated from $\overline{C_1C}$ and $e_s = |\overline{A_1C_1}|$ in Figure 15c. The triangle ABC is the same as depicted in the proof in [22] (Lemma 2), where the distance to the target centre for the robot to start turning was calculated using this 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 $\frac{2d_s + r(\pi - \alpha)}{v}$.

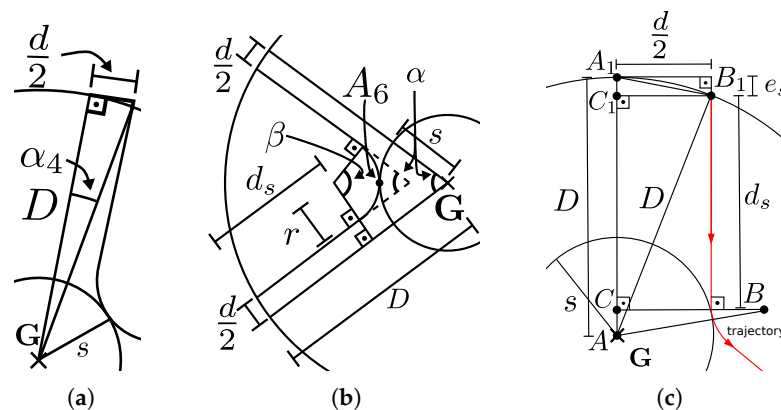


Figure 15. (a) Sector angle α_4 where the first robot on that lane arrives; (b) estimated location A_6 where the first robot will arrive at the target area; (c) right triangles $A_1B_1C_1$ and ABC for calculating the lengths d_s and e_s .

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 is obtained by adding parts (a)–(c). $\square$

3.4. Mixed Teams

Algorithm Description: Mixed Teams (MT) occurs when multiple groups of swarms act in the same environment, but they do not know the algorithm used by other groups. Assume two groups of robots, group A of $M < N$ robots and group B of $N - M$ robots. As explained below, the estimation for MT shows that the estimations for the algorithms can be used as a basis for further derivations. A robot of group A at position $\mathbf{r}_i(t)$ executes an algorithm Alg which returns a vector $\mathbf{F}_{Alg}$. Thus, their attractive force is simply $\mathbf{F}_A(\mathbf{r}_i(t)) = \mathbf{F}_{Alg}(\mathbf{r}_i(t))$. 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 Appendix A, using NC has the best results among the experimented algorithms for the group A when the percentage of robots in group A to the total number of robots ($\frac{N}{M} \cdot 100\%$) is lower than 30% and 60% for non-holonomic and holonomic robots, respectively. Thus, in the analysis below $\mathbf{F}_{Alg}$ is the same as in (3), and $Alg = NC$ was used in the following illustrations.

Samples of the Observations: Figures 16–19 show the execution of MT by 10% and 50% of robots in group A (in grey) to the total number of robots, respectively, when the robots in group B are executing the SQF algorithm with its default values. In these figures, the total number of robots is $N = 100$. In Figure 16b, the usual behaviour of the 10% of the robots in group A is to go directly to the target. In Figure 16c, 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 16d (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 17a). After that, only robots using SQF are in the experiment (Figure 17b). When a few robots use NC, this often happens.

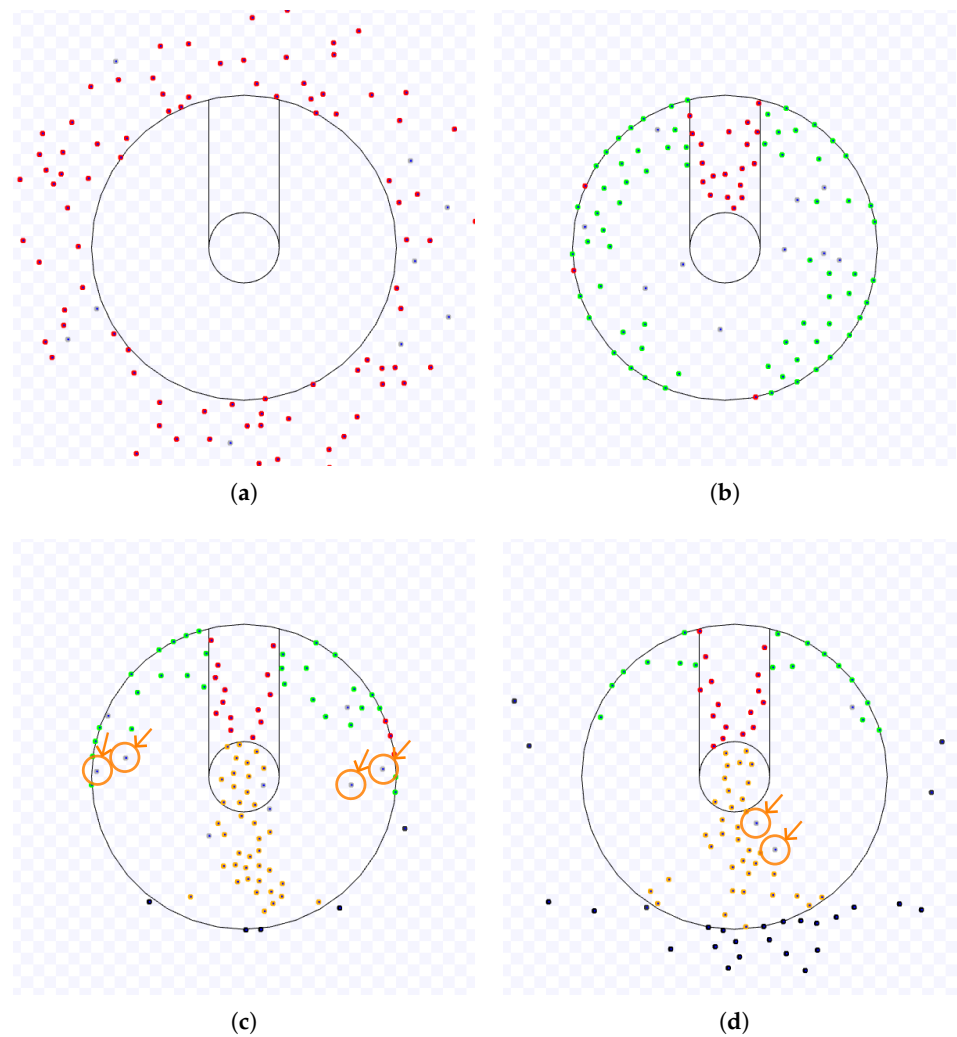


Figure 16. Example of an experiment of MT with 10% of robots in group A ($M = 10$ and $N = 100$) and the SQF algorithm executed by group B: (a) start of the experiment; (b) robots using NC go directly to the target; (c) some grey robots go to the next target; (d) the last robots using NC leave the target, but SQF robots are on their way to the next target. Full simulation available on <https://youtu.be/uBetIxGOI20>, accessed on 24 April 2023.

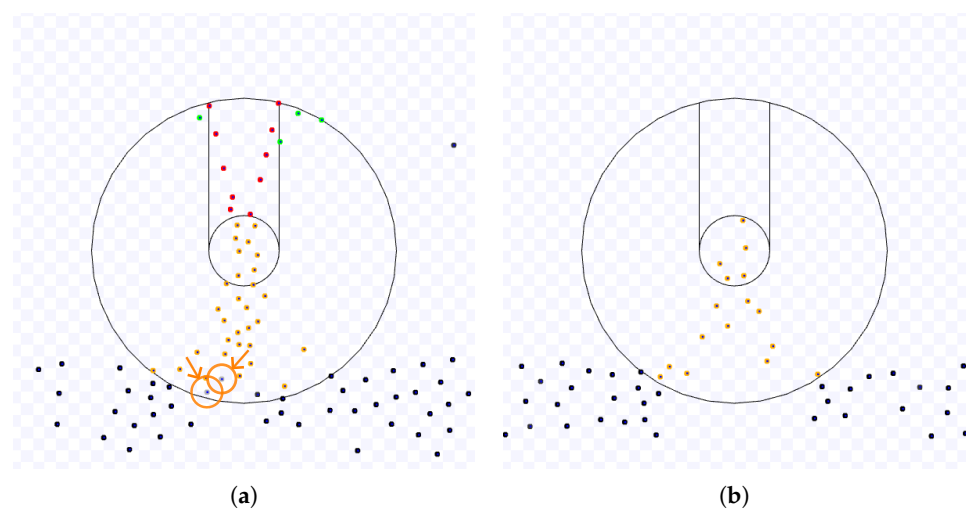


Figure 17. Continuation of Figure 16: (a) the last grey robots leave the working area through the SQF-induced leaving route; (b) the remaining robots use the SQF algorithm near the end.

Figures 18 and 19 have more robots using NC. From the bottom of Figure 18b, 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 18c). 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 18d). This tendency to clutter occurs proportionally to the number of robots using NC as the algorithm of group A. 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 19a). 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 19b).

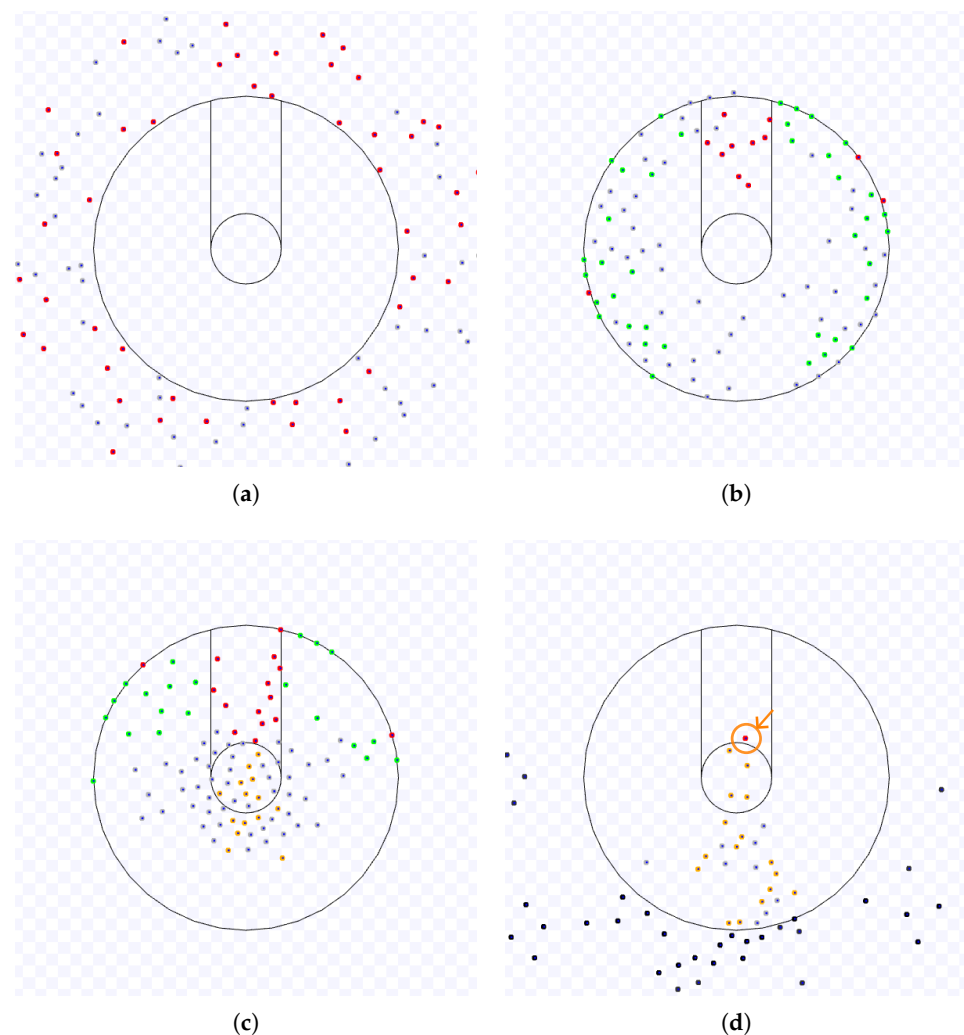


Figure 18. Example of an experiment of MT with 50% of robots in group A ($M = 50$ and $N = 100$) and the SQF algorithm executed by group B: (a) start of the experiment; (b) robots using NC go directly to the target; (c) robots using NC go to the target after being pushed by SQF robots leaving the target; (d) the last robot using SQF reaches the target area. Full simulation available on <https://youtu.be/a-05slaNud8>, accessed on 24 April 2023.

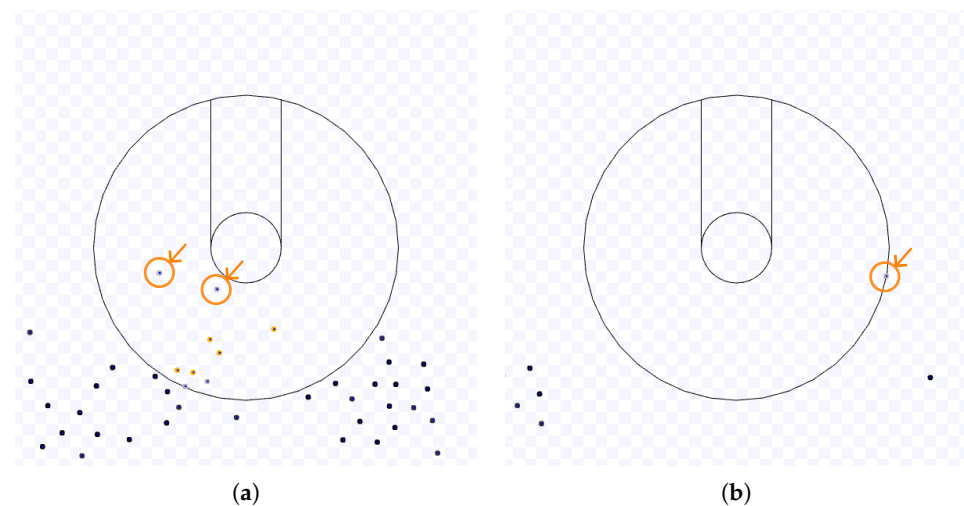


Figure 19. Continuation of Figure 18: (a) the robots on the SQF-induced leaving route are the last to arrive at the target region; (b) end of the experiment.

Figures 20–23 illustrate the execution of MT with 10% and 50% of robots in group A to the total number of robots for an experiment with the TRVF, with the default parameter values as the algorithm of the robots in group B. 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 20b shows the robots using NC arriving 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 20c). 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 20d). In Figure 21a, 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 21b.

Figures 22 and 23 exemplify the usage of 50% of robots in group A, and group B executes 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 22b) and cause congestion inside the target area (Figure 22c). If a lane has more robots using NC, they hamper robots using TRVF from going through it, as the exit lane on the top left-hand side in Figure 22d. As for the 10% case, the last robot using NC leaves before the TRVF robots (marked with a circle and an arrow in Figure 23a). However, fewer TRVF robots than in the 10% case are inside the working area until the end of the experiment (Figure 23b).

These illustrations hint that the proportion of robots in group A induces a cluttering similar to that which occurs by using only NC. Thus, the task completion time tends to be near NC time as the number of robots in group A M increases and the algorithm followed by the robots in group B as M decreases. To better show this, Figures 24 and 25 show the result from the experiments with MT for scenarios with the robots executing SQF and TRVF with different values of M . The bars represent the 99% confidence interval of the average for 40 runs for each value in the horizontal axis obtained from the Student's t -distribution. (The examples above are shown for convenience for the explanation below. More detailed results are shown in Section 4.)

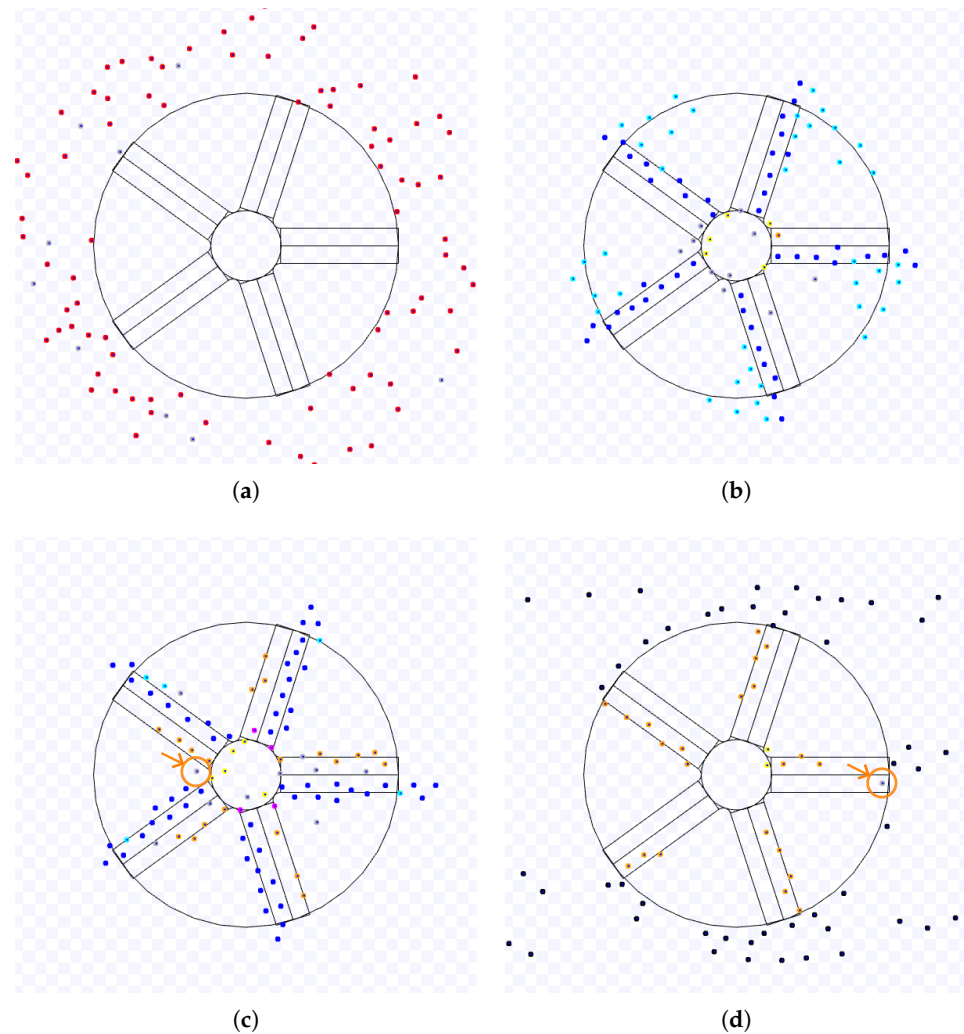


Figure 20. Example of an experiment of MT with 10% of robots in group A ($M = 10$ and $N = 100$) and the TRVF algorithm executed by group B: (a) start of the experiment; (b) robots using NC arrive at the target area through the free space; (c) a grey robot using NC waits for robots in a lane; (d) the last grey robot is leaving the working area. Full simulation available on <https://youtu.be/VlhXKM77q90>, accessed on 24 April 2023.

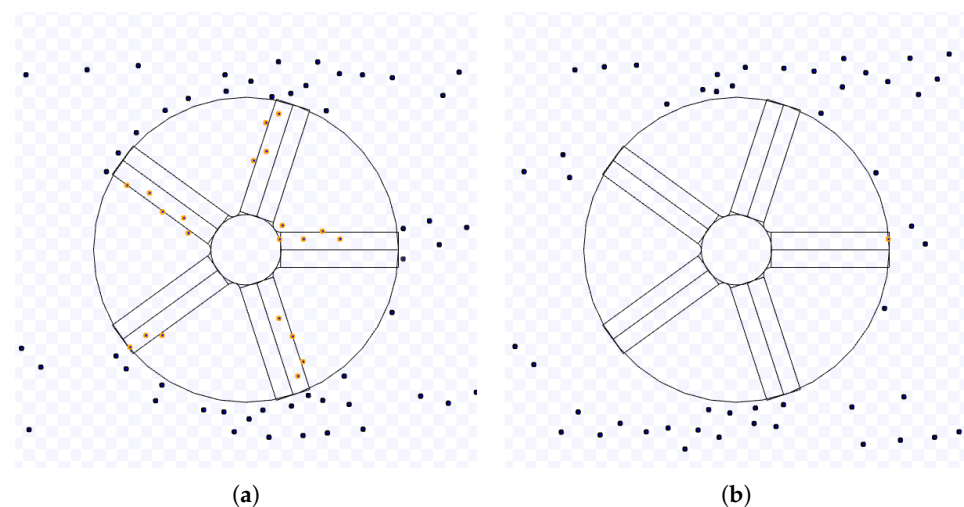


Figure 21. Continuation of Figure 20: (a) only robots using TRVF are inside the working area; (b) end of the experiment.

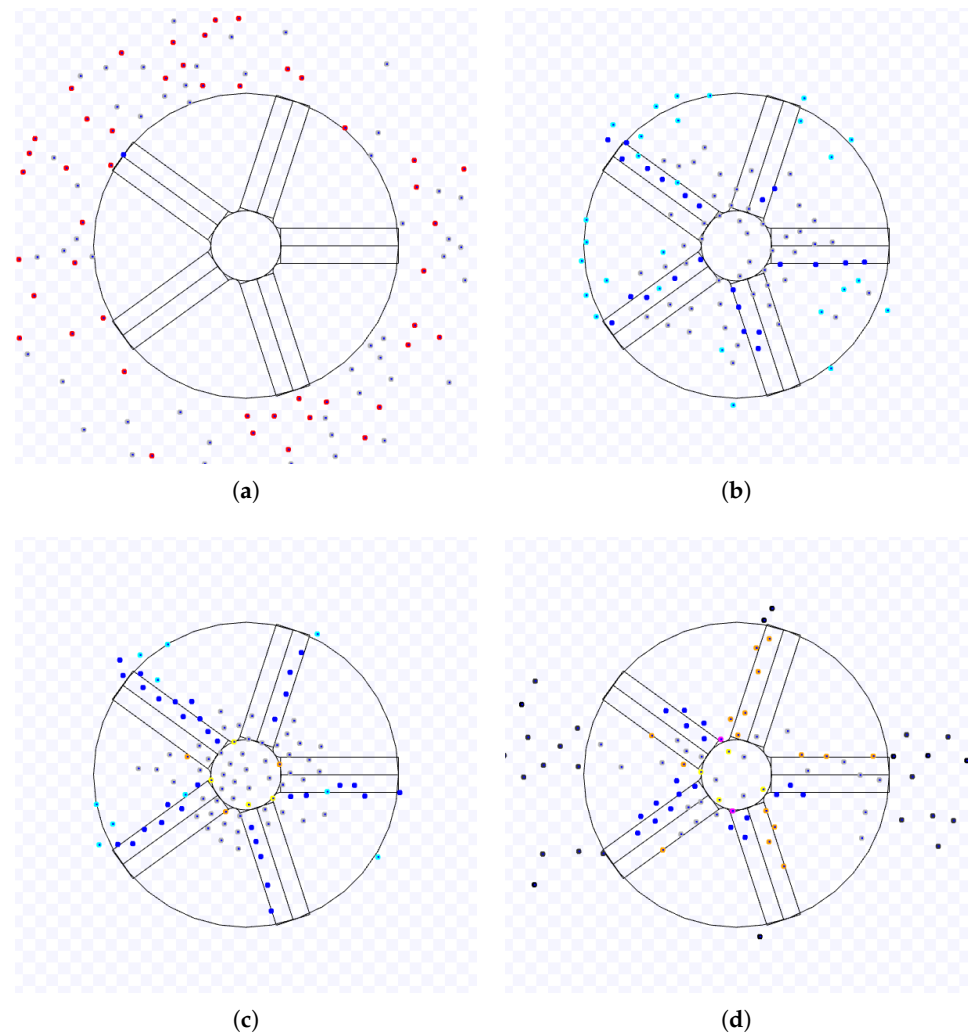


Figure 22. Example of an experiment of MT with 50% of robots in group A ($M = 50$ and $N = 100$) and the TRVF algorithm executed by group B: (a) start of the experiment; (b) robots using NC go directly to the target; (c) NC robots (in grey) cluster in the target area; (d) NC robots interfere with the access of robots using TRVF. Full simulation available on <https://youtu.be/1nxSEsre5mk>, accessed on 24 April 2023.

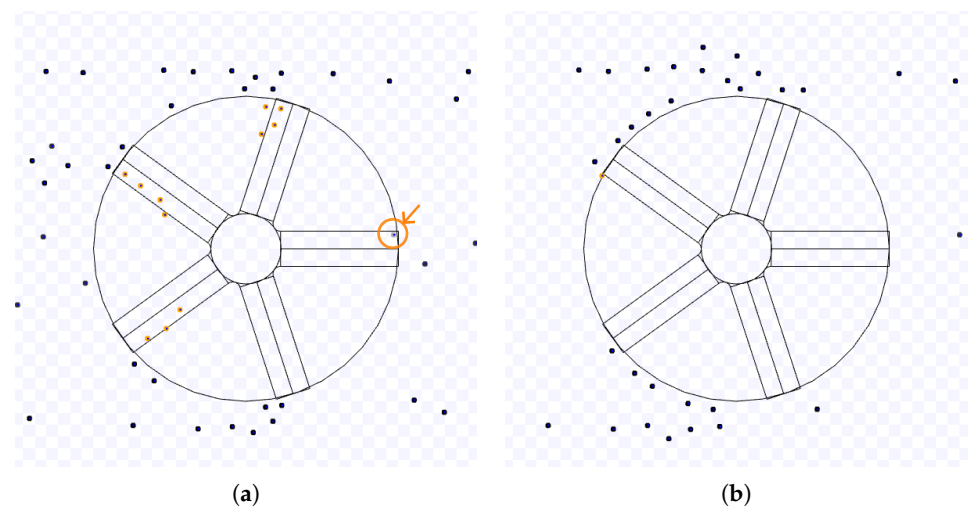


Figure 23. Continuation of Figure 22: (a) the last robot using NC leaves the working area; (b) end of the experiment.

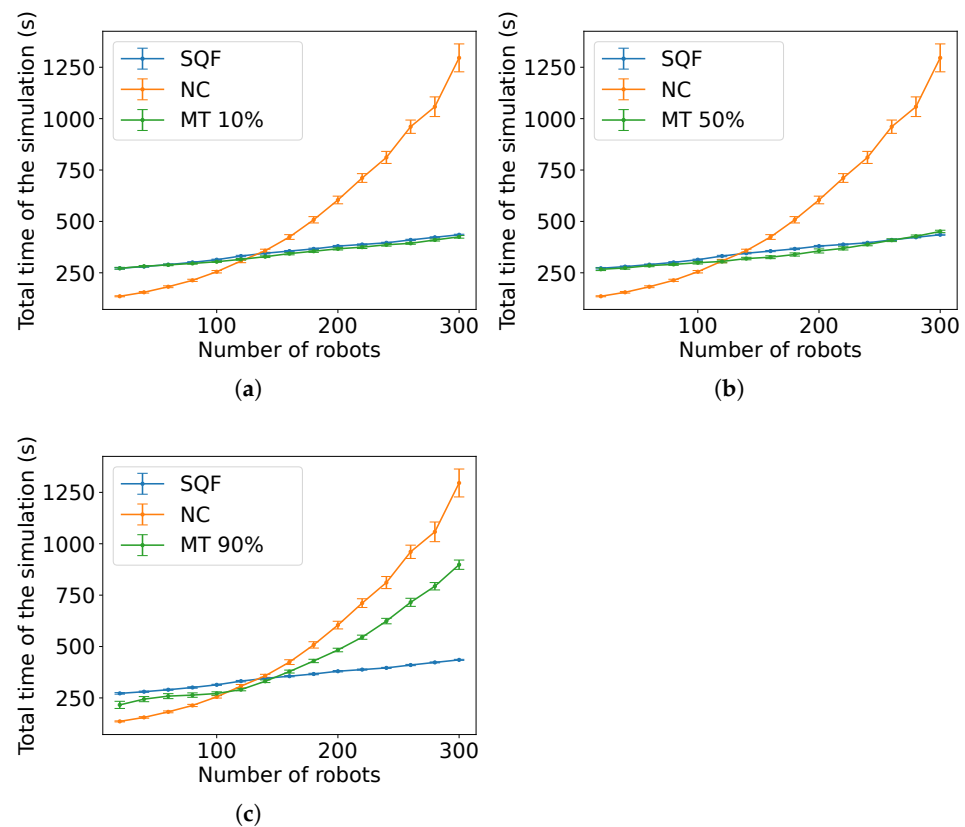


Figure 24. Task completion time versus the number of robots for NC, SQF and an MT with NC as the group A algorithm and robots in group B with SQF for holonomic robots and (a) $M = 0.1 N$, (b) $M = 0.5 N$ and (c) $M = 0.9 N$.

Based on these observations, the next estimation shows an approximation for an MT, followed by its explanation. The estimation below is a linear combination of the estimated times of the algorithm NC, followed by one group, and of the algorithm executed by the other group.

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 group A algorithm depends on the ratio $p = M/N$ and is expressed by

$$\mathbb{E}[t_{MT}](p) \approx C_{MTNC}\mathbb{E}[t_{NC}] + C_{MTB}\mathbb{E}[t_{AlgB}] \quad (17)$$

for constants C_{MTNC} and C_{MTB} , $AlgB$ being the algorithm followed by the $N - M$ robots in group B and its estimated expected time $\mathbb{E}[t_{AlgB}]$.

Explanation. Let C_{MTNC} be a given constant abstracting the environment and the dynamics of the robots in group A for the ratio p and C_{MTB} a similar constant for the robots in group B. Thus, as observed in the experiments (as shown later in Section 4, Figures 24, 25, 35 and 36 and in Appendix B, Figures A9–A14), the estimated expected time is the sum of the estimated expected time of NC and the control algorithm followed by the robots in group B each multiplied by the constants C_{MTNC} and C_{MTB} , respectively. $\square$

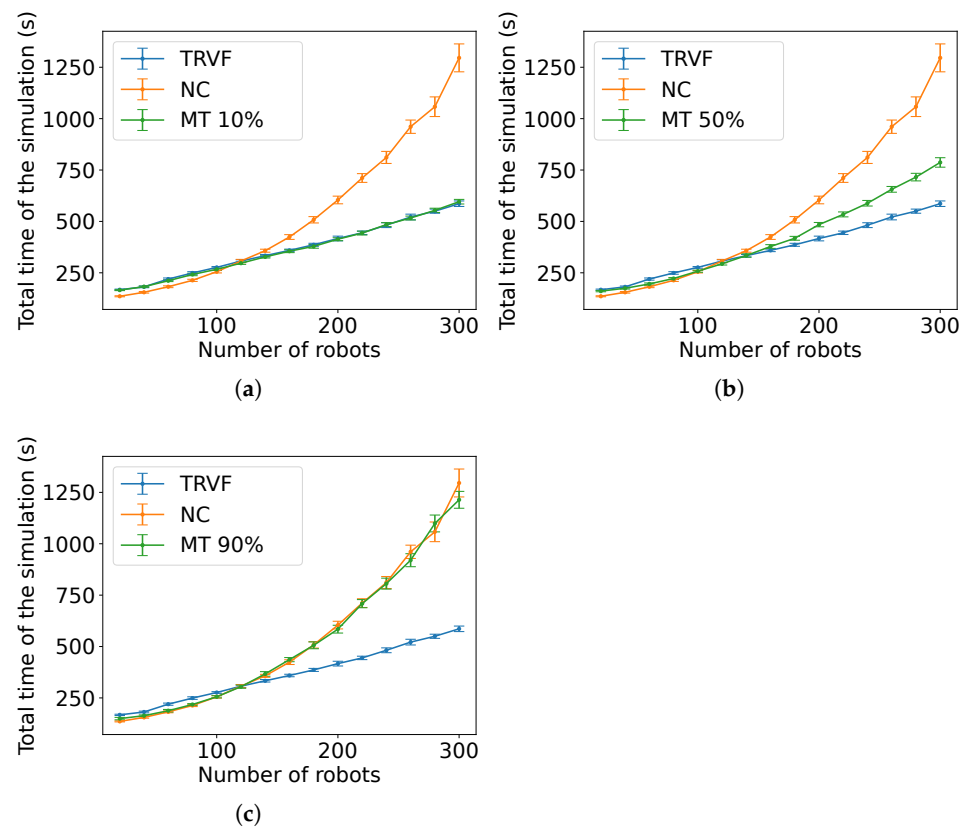


Figure 25. Task completion time versus the number of robots for NC, TRVF and MT with NC as the group A algorithm and robots in group B with TRVF for holonomic robots and (a) $M = 0.1 N$, (b) $M = 0.5 N$ and (c) $M = 0.9 N$.

4. Experiments and Results

Experiments were performed in the Stage simulator [39] 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 the estimations). For each N , 40 executions were run. In the figures of this section, the error bars represent the 99% confidence interval of the mean value of the considered variable obtained from the Student's t -distribution. “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, as described by Equation (2). It is important to note 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 constants obtained from the least squares 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 experimental data than the presented equation. The source codes of the experimented algorithms and estimations are in <https://github.com/yuri-tavares/swarm-ad-hoc-follower> (accessed on 3 November 2025).

Figure 26 and 27 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.

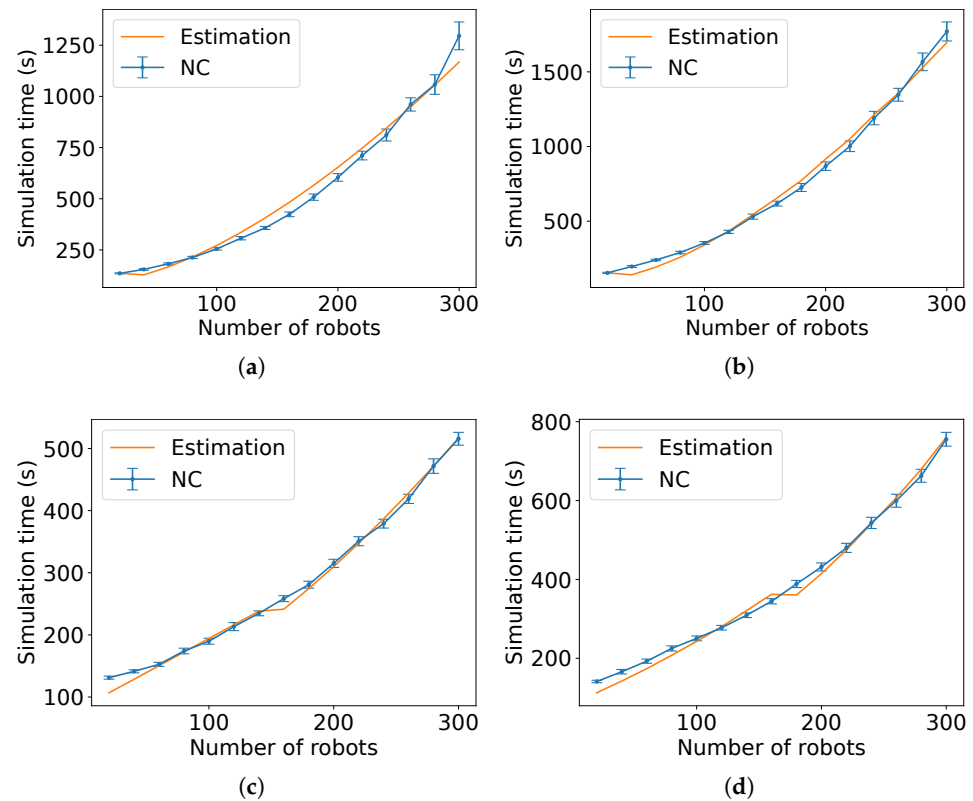


Figure 26. Comparison of the estimated expected time and the experimental data for NC, $s = 3$ m with (a) holonomic robots ($(C_{NC1}, C_{NC2}) = (2.1431, 0.6502)$) and (b) non-holonomic robots ($(C_{NC1}, C_{NC2}) = (2.7784, 0.5694)$) and $s = 5$ m with (c) holonomic robots ($(C_{NC1}, C_{NC2}) = (1.0994, 0.2614)$) and (d) non-holonomic robots ($(C_{NC1}, C_{NC2}) = (1.2867, 0.2539)$).

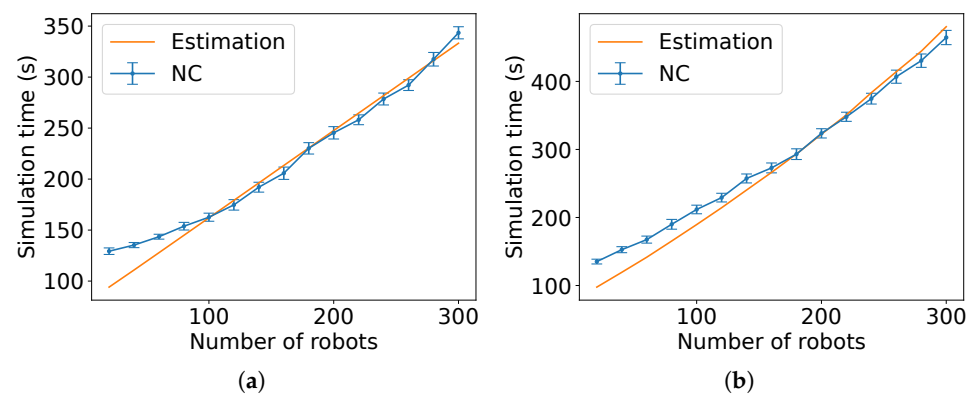


Figure 27. Comparison of the estimated expected time and the experimental data for NC, $s = 7$ m with (a) holonomic robots ($(C_{NC1}, C_{NC2}) = (0.857, 0)$) and (b) non-holonomic robots ($(C_{NC1}, C_{NC2}) = (0.9795, 0)$).

Figures 28 and 29 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 28b) 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 types 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.

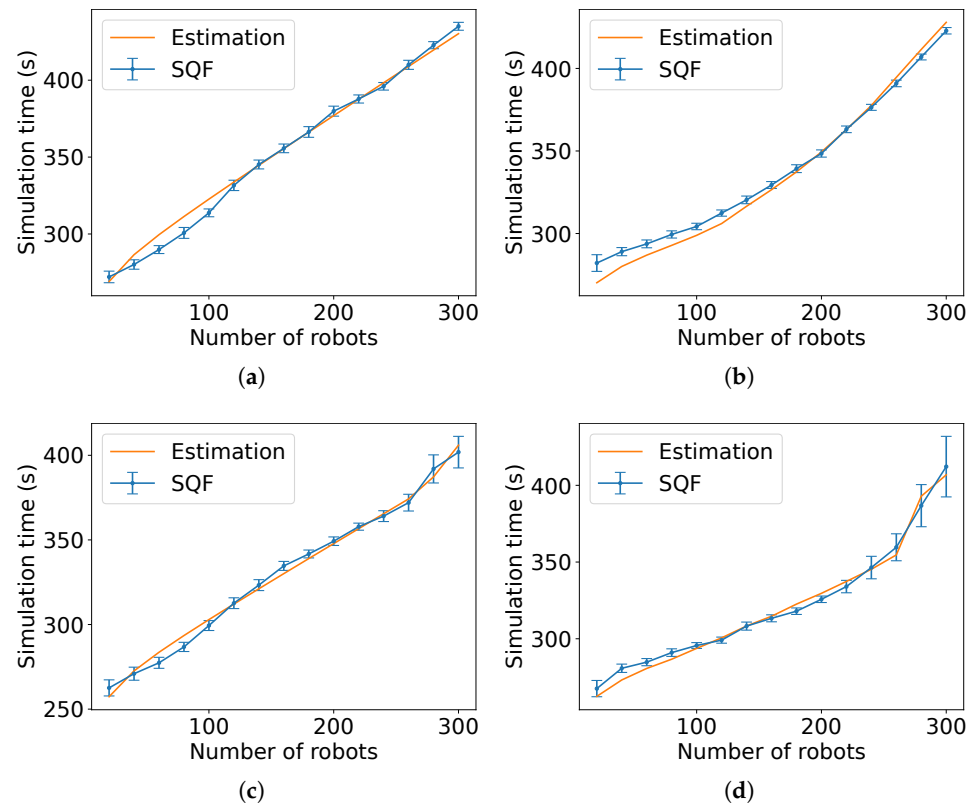


Figure 28. Comparison of the estimated expected time and the experimental data for the SQF algorithm, $s = 3$ m with (a) holonomic robots ($(C_{SQF1}, C_{SQF2}) = (0.5277, 0)$) and (b) non-holonomic robots ($(C_{SQF1}, C_{SQF2}) = (-0.1202, 0)$) and $s = 5$ m with (c) holonomic robots ($(C_{SQF1}, C_{SQF2}) = (0.4346, 0.0016)$) and (d) non-holonomic robots ($(C_{SQF1}, C_{SQF2}) = (0.0037, 0.0004)$).

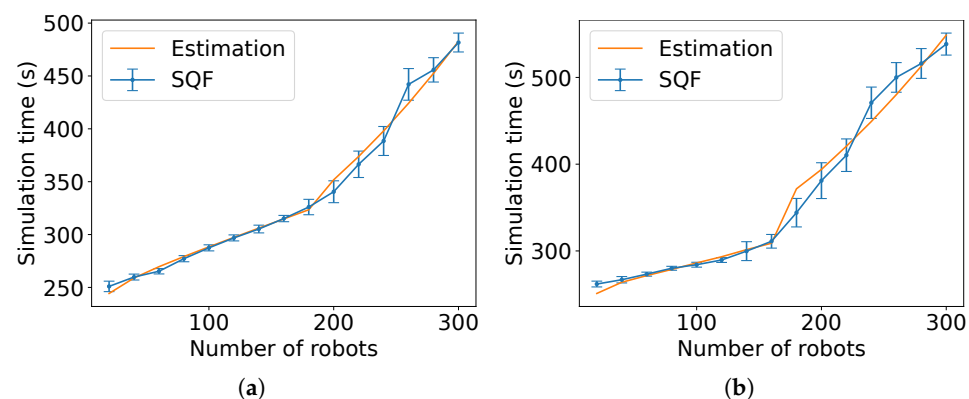


Figure 29. Comparison of the estimated expected time and the experimental data for the SQF algorithm, $s = 7$ m with (a) holonomic robots ($(C_{SQF1}, C_{SQF2}) = (0.4237, 0.0026)$) and (b) non-holonomic robots ($(C_{SQF1}, C_{SQF2}) = (0.0928, 0.0022)$).

Figures 30 and 31 show the comparison for the 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 31). In the algorithms where the lower number of robots differed 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.

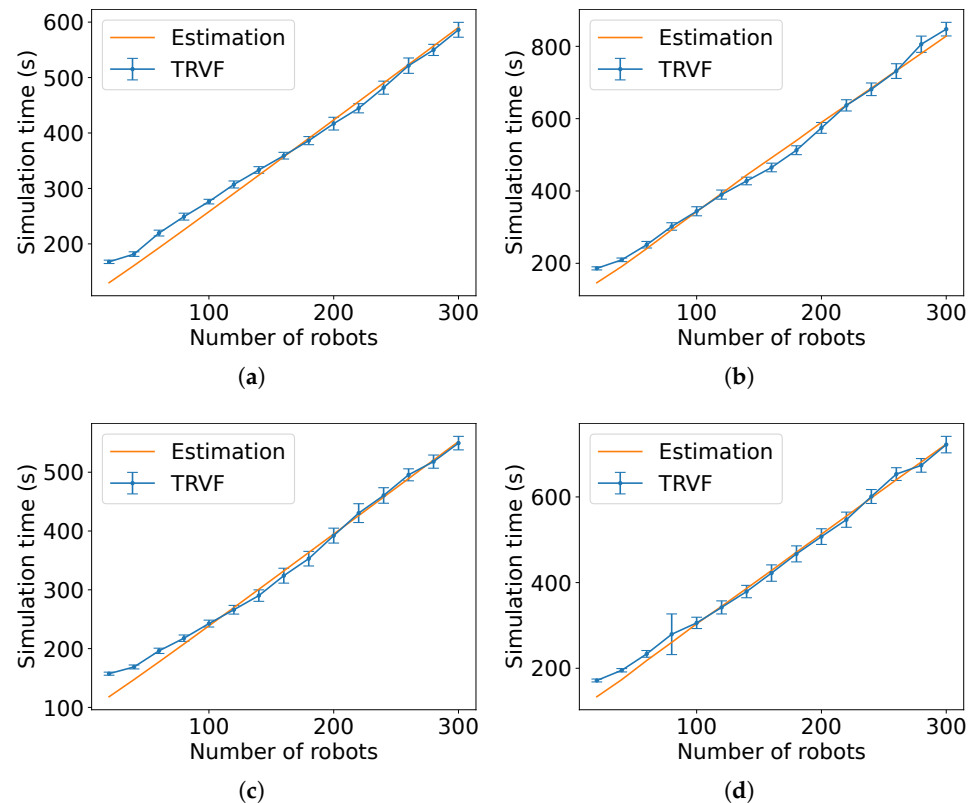


Figure 30. Comparison of the estimated expected time and the experimental data for the TRVF algorithm, $s = 3$ m with (a) holonomic robots ($C_{TRVF} = 8.3866$) and (b) non-holonomic robots ($C_{TRVF} = 10.0287$) and $s = 5$ m with (c) holonomic robots ($C_{TRVF} = 7.9166$) and (d) non-holonomic robots ($C_{TRVF} = 9.3022$).

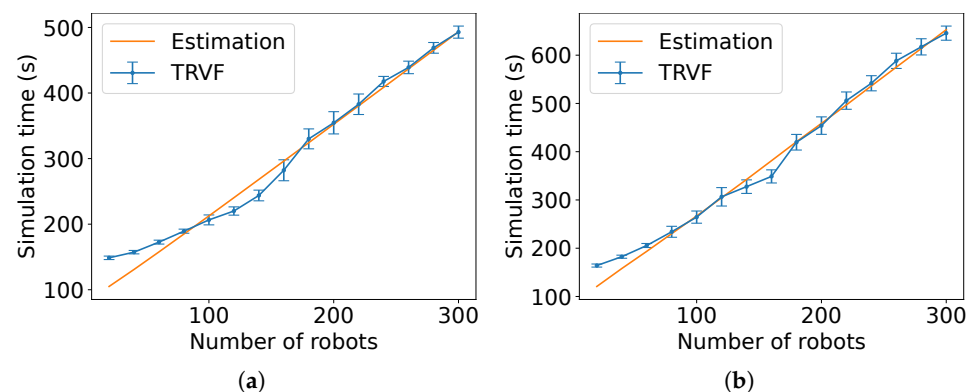


Figure 31. Comparison of the estimated expected time and the experimental data for the TRVF algorithm, $s = 7$ m with (a) holonomic robots ($C_{TRVF} = 7.1029$) and (b) non-holonomic robots ($C_{TRVF} = 8.7103$).

Figure 32 gives the results of the root mean squared error normalised by the y -axis range (NRMSE) for the estimations in Figures 26–31. From it, these estimations are considered close to the experimental data because the NRMSE values were lower than 0.03. In Figure 32, the lowest error for holonomic robots was for SQF with $s = 5$ m, while for the 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 environmental area affect these functions in this sense.

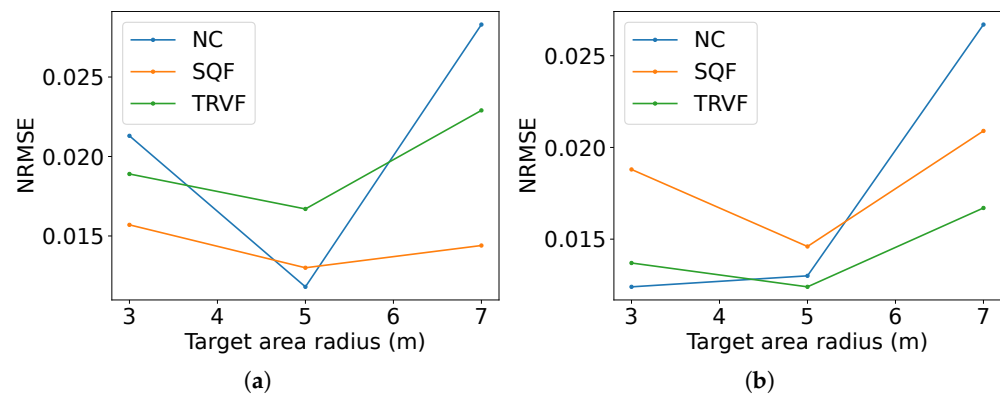


Figure 32. Normalised root mean square error of the estimations in Figures 26–31 for (a) holonomic and (b) non-holonomic robots.

Although it is not explicit from Estimations 1–3 for the estimated expected time, $\bar{d}$ and $\bar{v}$ depend on N . Figure 33 presents the relation between the average speed and the number of robots for the algorithms for $s = 3$ m obtained from the experiments. 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 backward, 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.

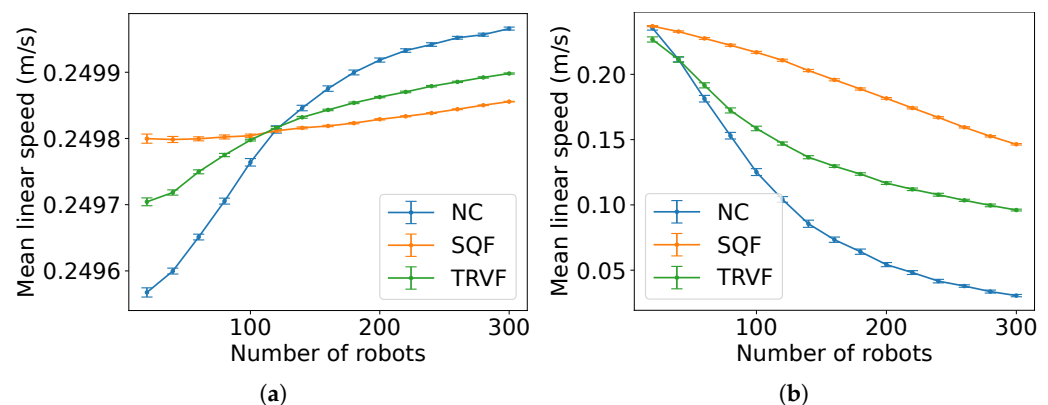


Figure 33. Comparison of the linear speeds of the algorithms for $s = 3$ m with (a) holonomic robots and (b) non-holonomic robots.

Figure 34 shows the relation between the mean distance between the robots and the number of robots obtained from experiments. For both types of robots, the trends in the mean distance are similar: as the number of robots grows, they have to move closer to each other in the algorithms.

Figure 35 presents an instance of the estimations for MT with the proportion $p = 10\%$ of robots in group A and SQF as the control algorithm of the robots in group B. For reference, the task completion time from the experiments of SQF and NC is shown. Appendix B contains the figures for p from 20% to 90% in increments of 10%. Figure 37a 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%.

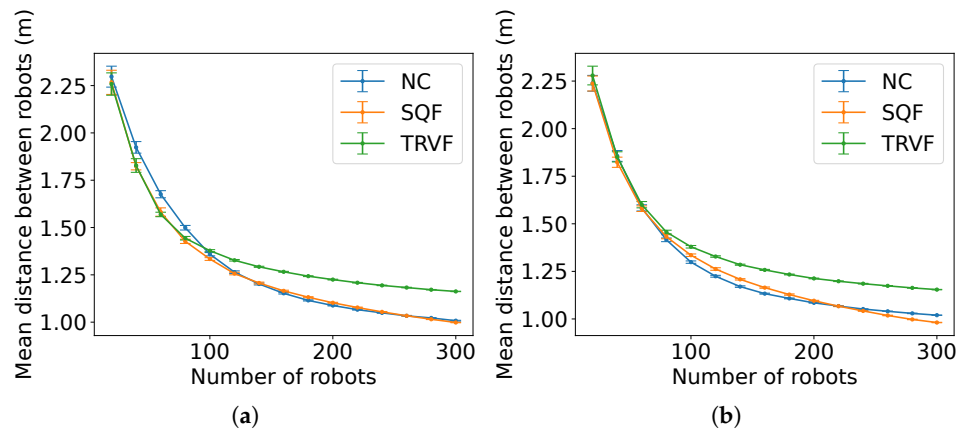


Figure 34. Comparison of the mean distance between the robots of the algorithms for $s = 3$ m with (a) holonomic robots and (b) non-holonomic robots.

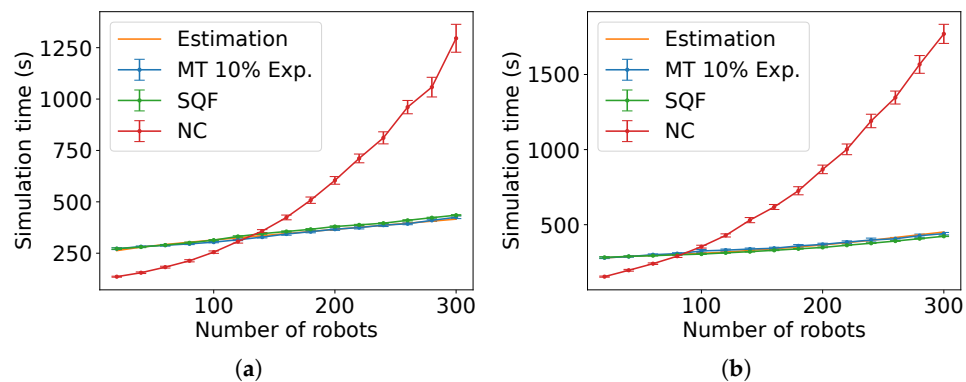


Figure 35. Comparison of the estimated expected time and the experimental data for MT with NC as group A's algorithm, group B is executing the SQF algorithm and $p = 10\%$ for (a) holonomic robots ($C_{MTNC} = -0.0042$ and $C_{MTB} = 0.9779$) and (b) non-holonomic robots ($C_{MTNC} = 0.0069$ and $C_{MTB} = 1.027$).

Similarly, Figure 36 shows an example of the estimations for MT with TRVF as the control algorithm of group B 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 Appendix B. Figure 37b 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 the holonomic case.

Figure 38 displays the change in constants C_{MTB} and C_{MTNC} by the experimented percentage of robots in group A, shown above for each tested control algorithm of the robots in group B and robots' movement type. Note that when C_{MTNC} rises, C_{MTB} decreases because it is expected that 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 (Figure 38a). The same happens for the TRVF algorithm in Figure 38b.

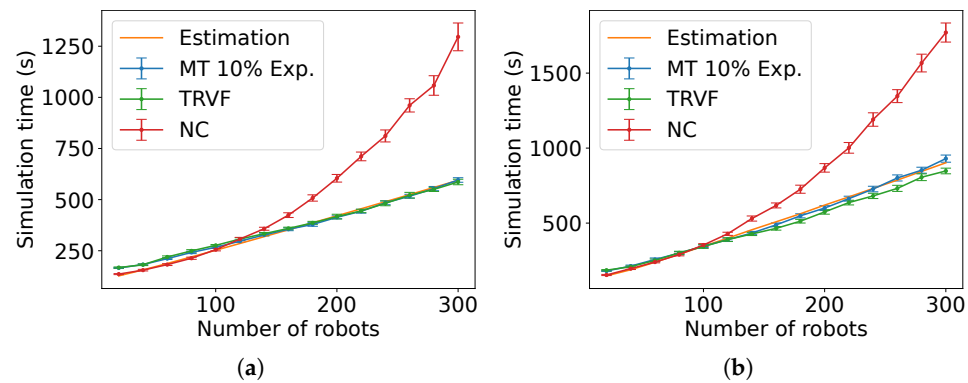


Figure 36. Comparison of the estimated expected time and the experimental data for MT with NC as group A's algorithm, group B is executing the TRVF algorithm and $p = 10\%$ for (a) holonomic robots ($C_{MTNC} = 0.0309$ and $C_{MTB} = 0.9459$) and (b) non-holonomic robots ($C_{MTNC} = 0.0766$ and $C_{MTB} = 0.9323$).

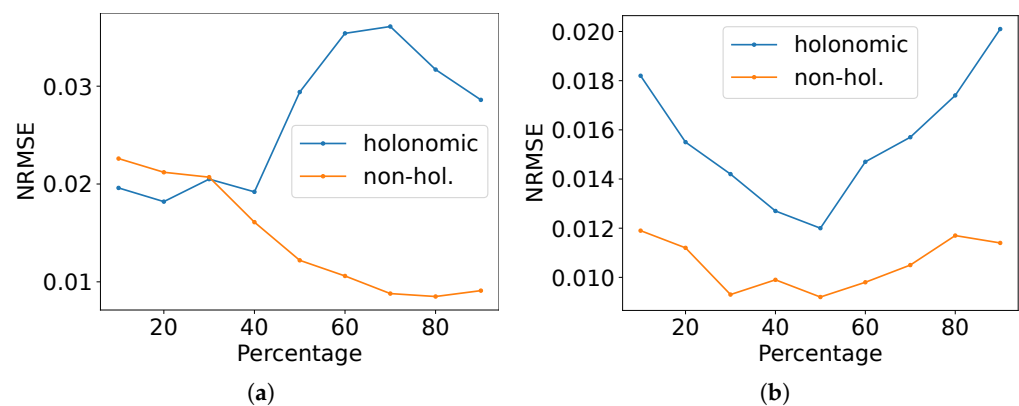


Figure 37. Normalised root mean square error of the estimations with different percentages of robots in group A using NC for an MT with group B using (a) SQF and (b) TRVF.

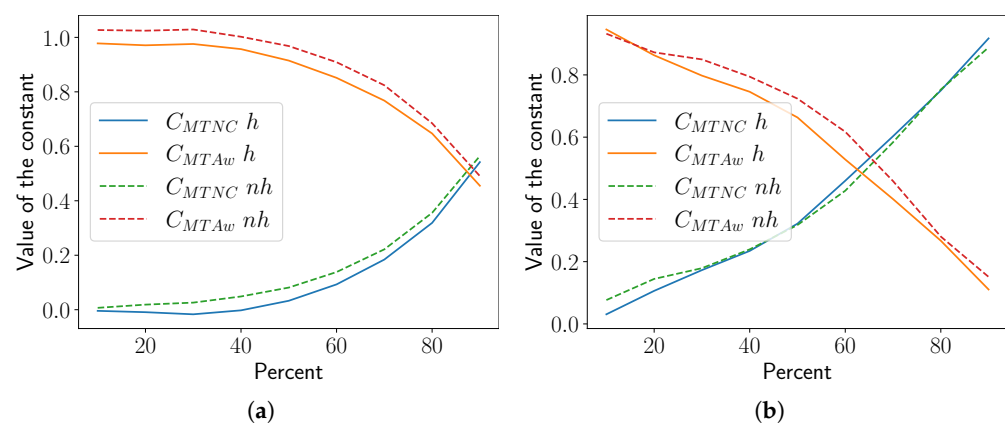


Figure 38. Constants used for MT estimations (C_{MTB} and C_{MTNC}) versus the percentage of robots in group A for the control algorithm of robots in group B (a) SQF and (b) TRVF. In the graphs, h and nh stand for holonomic and non-holonomic robots, respectively.

5. Discussion

In this section, limitations, challenges and future improvements to the presented work are discussed. Firstly, this paper assumes that the only random operation is the initial position of the robots. Aside from that, the robots would behave deterministically. Therefore, the average task completion is due to the random initialisation. Thus, a best/worst-case scenario may be estimated by taking the standard deviation from above and below the

average time in most cases. Although this is true for the simulations discussed in this paper, for real robots, there would be more stochastic events that should be included in these best-case/worst-case scenarios.

Secondly, the presented estimations are valid for changes in the parameters N and s as tested and described in Section 4. However, the estimations were not experimented with different values of every global parameter other than N and s (Table 3).

Thirdly, for an extreme scenario of high density (fixed working area radius D and $N \rightarrow \infty$), the estimations tend to infinity as they are polynomials in N . For low values of N and large s , the estimations tend to differ more from the experimental data. In these situations, the robots are scattered inside the working area and do not generally move as the global behaviour deduced for the estimations, which considered the subsumed action of numerous robots interacting next to each other. The authors believe the estimations work for reasonable changes in the other global parameters. Nonetheless, significant changes were not tested and could cause unexpected results, since the underlying algorithms could have undesired behaviour with a wrong parametrisation.

Fourthly, the estimations for mixed teams (MT) are devised as a linear combination of known estimations and worked well, as evidenced by an NRMSE less than 0.04. Though not tested by the authors, this rationale may be applied to heterogeneous swarms if each homogenous subgroup has different estimations. In other words, if the heterogeneous swarm is composed of n homogeneous subgroups, each having estimators of $\mathbb{E}[t_i]$ for $i \in \{1, \dots, n\}$, the authors expect that a linear combination given by $\sum_{i=1}^n C_i \mathbb{E}[t_i]$ may estimate satisfactorily, where C_i are the constants fitted for each subgroup estimation, similarly to the study presented here for MT.

Fifthly, the presented algorithms may be adapted to three-dimensional robots and experimented with their type of movements due to the recent increase in usage of aerial robots. Because this paper considers robots and the target regions in a plane, the equations must be extended to three dimensions. Three-dimensional robots have different types of movement that may influence the presented algorithms differently. For instance, aircraft-like robots cannot stop in mid-air, and some aerial robots are controlled by Dubins paths.

In addition, adapting a method made for a space with a fixed number of dimensions may not be effortless. The algorithms SQF and TRVF were born from an attempt by the authors to apply the queuing theory. However, this was unsuccessful because, in two-dimensional space, the queues behave differently depending on the path followed by the robots. In other words, the original queuing theory is one-dimensional, and adapting to two dimensions did not seem to be straightforward. Thus, only extending the algorithms may not be sufficient; they also must be tested for feasibility. Should the extended algorithms be ineffective in three dimensions, new algorithms ought to be devised.

Another future step is to apply methods from statistical mechanics, in particular for many-body problems, to this problem and to compare with those developed here. Due to the wide range of applications in multi-agent systems, mean-field theory seems to be capable of solving the estimation of the simulation time in the common-target problem. Nonetheless, it is unclear yet how mean-field-like trajectories from local controllers can be formally calculated to yield the global expected simulation time. Such solutions may help to find a potential field that minimises the relation of the task completion time versus the number of robots. Additionally, these studies may answer whether finding such a potential field is possible. Notice that if this is possible, a time relation for a given field has to be given to be minimised, and this work focused on approximating such a relation by exploring the usage of the geometry of the overall behaviour of the robots. These ideas are the basis for future improvements, which include finding a better potential field minimising the task completion time per number of robots, presuming that finding it is decidable.

Furthermore, although the global metric used here was time, it is essential to note that other metrics are also important in Swarm Robotics. Energy consumption and energy-time product [71] are examples of global metrics that may be investigated in the future. The algorithms presented here have a similar nature, that is, they use potential fields and do not require communication. Hence, as far as we know, if those algorithms take less time, they should also consume less energy.

Finally, this paper presents a method of analysing the geometric behaviour from the experiments to approximate local-to-global functions. It can be applied to different problems of robotic swarms and other geometric settings if a fixed number of robots follow similar paths on average for different starting positions. For instance, a patrolling swarm may go to specific areas regularly, using a predefined route on average. By aggregating the routes, analysing their geometry and averaging by the frequency of usage, the expected time between patrols may be estimated. A similar analysis may be performed on a swarm covering an area to estimate the average time for completing that task. Another example is to approximate the volume per time of a swarm collecting waste. That approximation may be achieved by calculating the average time of collecting and accessing the areas with waste and the volume of waste from them by geometrically examining the used paths from the experiments and averaging by their frequency. That can also be applied to dynamical environments modelled by geometry by analysing and using the aspects involving geometric patterns. An example is a scenario with robots with energy constraints, such as the change of bright spots in open areas under the sun with light-seeking solar-powered robots in a swarm. The predicted spots to recharge and crowded places can be inferred by the presence of shadows or sunny areas, which may be non-convex but modelled by geometry.

6. Conclusions

This paper 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 MT. Experiments with MT of two groups showed that if the percentage of robots in one group 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 algorithm of a group, although it is not suitable for a swarm with only one group.

Besides the number of robots, the expected 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 robots in group A 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.

Therefore, this work is a fundamental first step to examining 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 this paper aims to start a discussion about that by presenting a baseline approximation to compare with future work. There is room for improvement in these estimations. Approaches using adaptations of the methods discussed in Section 2 (the perturbation theory, the coupled cluster technique, the density-functional theory, the mean-field theory and the variational method) are welcome to be compared.

Author Contributions: Y.T.d.P. and L.S.M. wrote the main manuscript. Y.T.d.P. wrote the source codes, executed the experiments and prepared the figures. Y.T.d.P. and L.S.M. reviewed the manuscript. Conceptualisation, Y.T.d.P. and L.S.M.; methodology, Y.T.d.P. and L.S.M.; software, Y.T.d.P.; validation, Y.T.d.P.; formal analysis, Y.T.d.P.; investigation, Y.T.d.P.; data curation, Y.T.d.P.; resources, Y.T.d.P. and L.S.M.; writing—original draft preparation, Y.T.d.P.; writing—review and editing, Y.T.d.P. and L.S.M.; visualisation, Y.T.d.P.; supervision, L.S.M.; project administration, Y.T.d.P. and L.S.M.; funding acquisition, Y.T.d.P. and L.S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Faculty of Science and Technology (FST) of Lancaster University.

Data Availability Statement: The original data presented in the study are openly available in GitHub at <https://github.com/yuri-tavares/swarm-ad-hoc-follower> (accessed on 3 November 2025).

Acknowledgments: Thanks to Gao Peng and Abdulrahman Kerim for the review of the earlier draft.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Parameter *Alg* of Mixed Teams

This section presents the results of using combinations of NC, SQF and TRVF by the two groups in an MT and different values of M/N , justifying the usage of $Alg = NC$ in Section 3.4. In the following sections, a fixed algorithm is set for group B and the usage of different algorithms for group A is compared by varying the number of robots and the ratio of robots in group A to the total number of robots $\left(\frac{M}{N}\right)$. Based on the results of Appendices A.1 and A.2, Appendix A.3 summarises the reason for choosing $Alg = NC$.

Appendix A.1. Comparison of NC and SQF for the Group A and TRVF for Group B

Figures A1–A3 illustrate the comparison of the total simulation time per total number of robots of an MT using SQF and NC as parameter *Alg* with percentages of robots in group A to the total number of individuals ranging from 10% to 90% in steps of 10%. In these experiments, TRVF is the swarm control algorithm of group B. These figures are summarised in Figure A4, which shows the sum of the average total simulation time over each experimented number of robots by the percentage of robots in group A for MT. The bars correspond to the respective sum of the confidence interval ranges with $\rho = 0.01$ obtained from the Student's t-distribution.

In Figures A1–A3, for 30% to 70% for non-holonomic robots, NC as the algorithm of group A is significantly better on average for 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 robots in group A.

Furthermore, SQF as the group A algorithm takes more or equal time on average than using NC for percentages from 10% to 20% of robots in group A using MT for both types of robots and 30% to 70% for holonomic robots with statistical significance (Figure A4). Although the SQF is faster, when used as the group A 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.

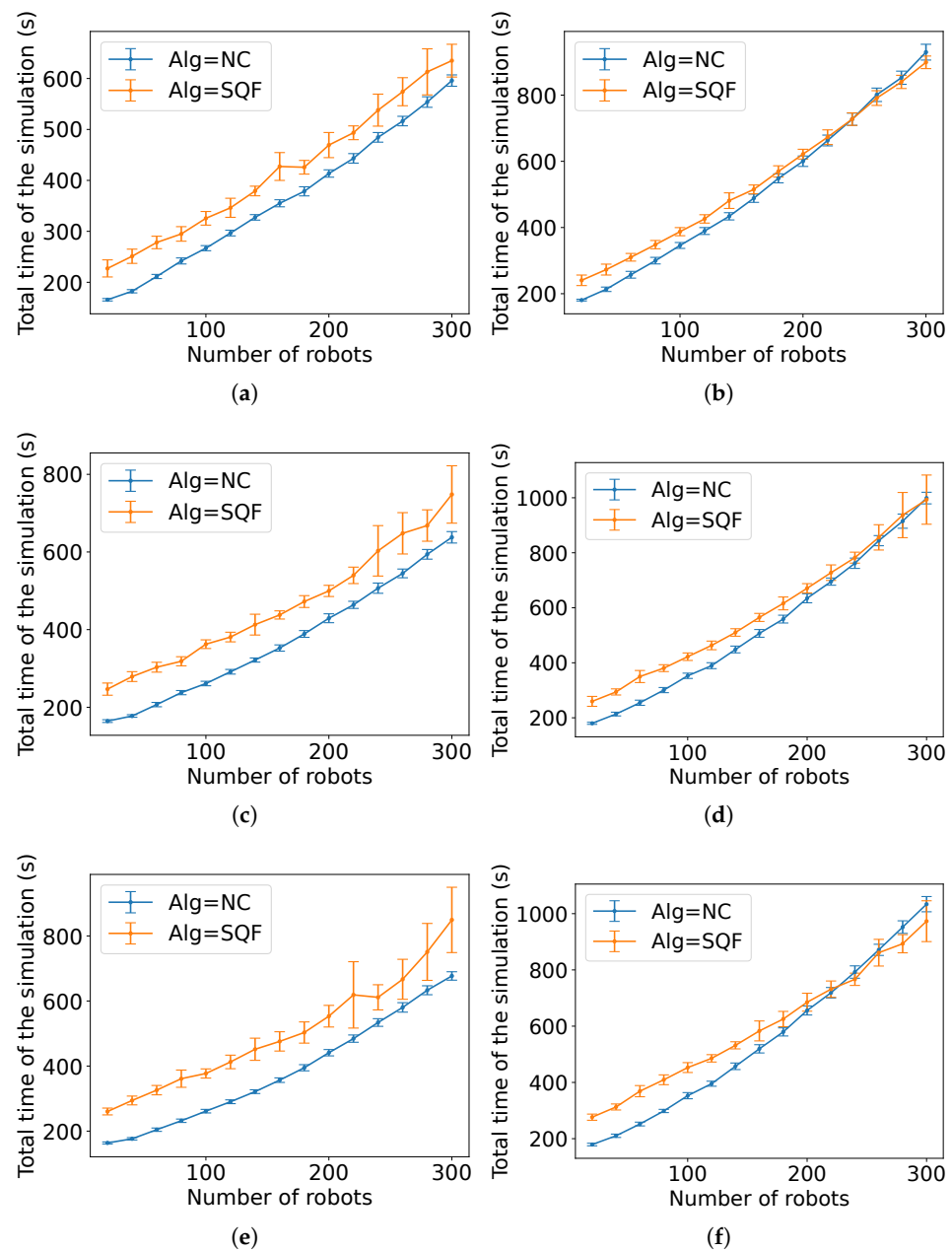


Figure A1. Comparison of the total simulation time by the number of robots for the experiments with MT using TRVF as the group B's algorithm and SQF and NC as the group A's algorithm by 10% of the robots for (a) holonomic and (b) non-holonomic robots, 20% for (c) holonomic and (d) non-holonomic robots and 30% for (e) holonomic and (f) non-holonomic robots.

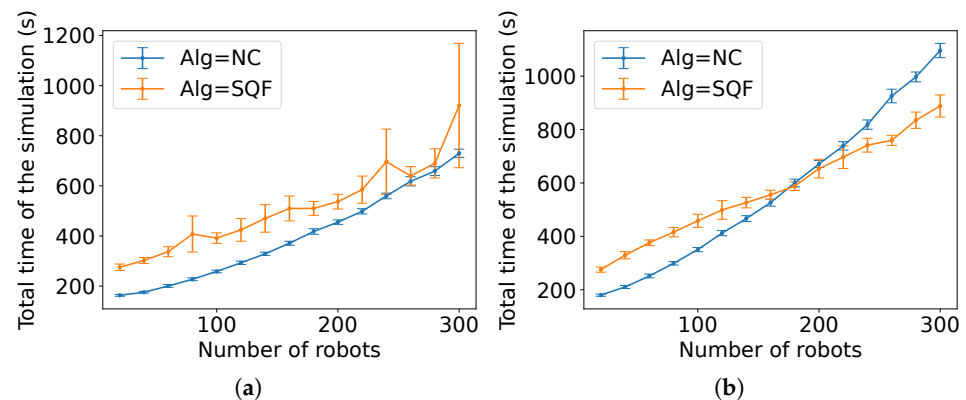


Figure A2. Cont.

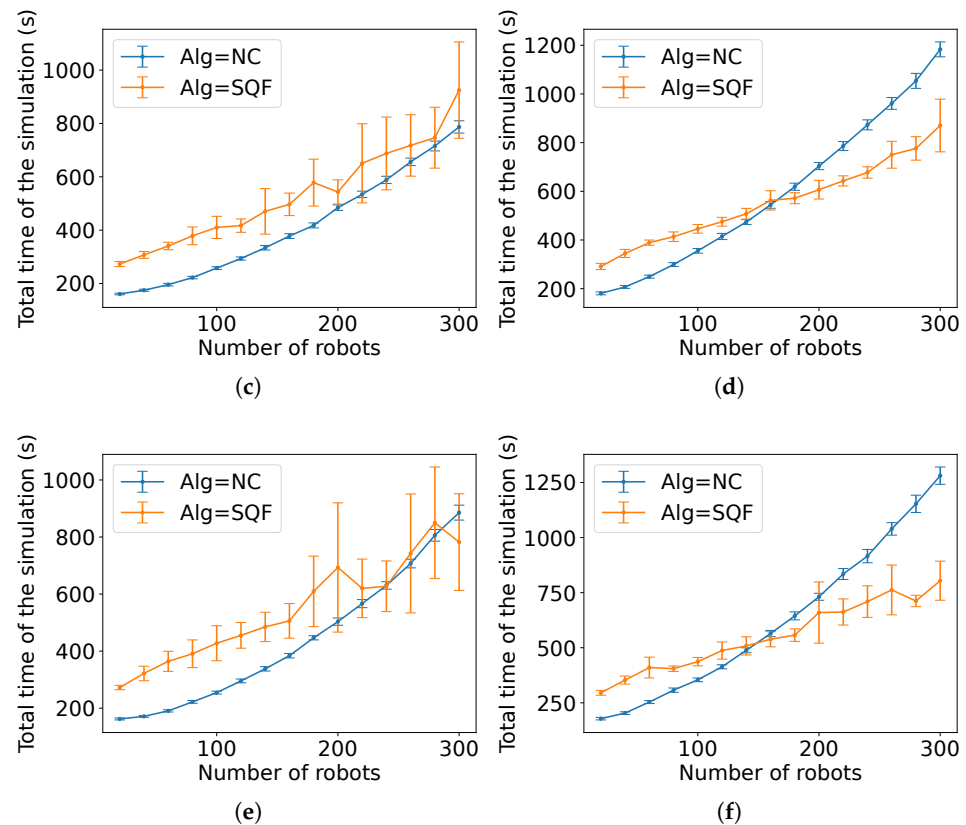


Figure A2. Comparison of the total simulation time by the number of robots for the experiments with MT using TRVF as the group B's algorithm and SQF and NC as the group A's algorithm by 40% of the robots for (a) holonomic and (b) non-holonomic robots, 50% for (c) holonomic and (d) non-holonomic robots and 60% for (e) holonomic and (f) non-holonomic robots.

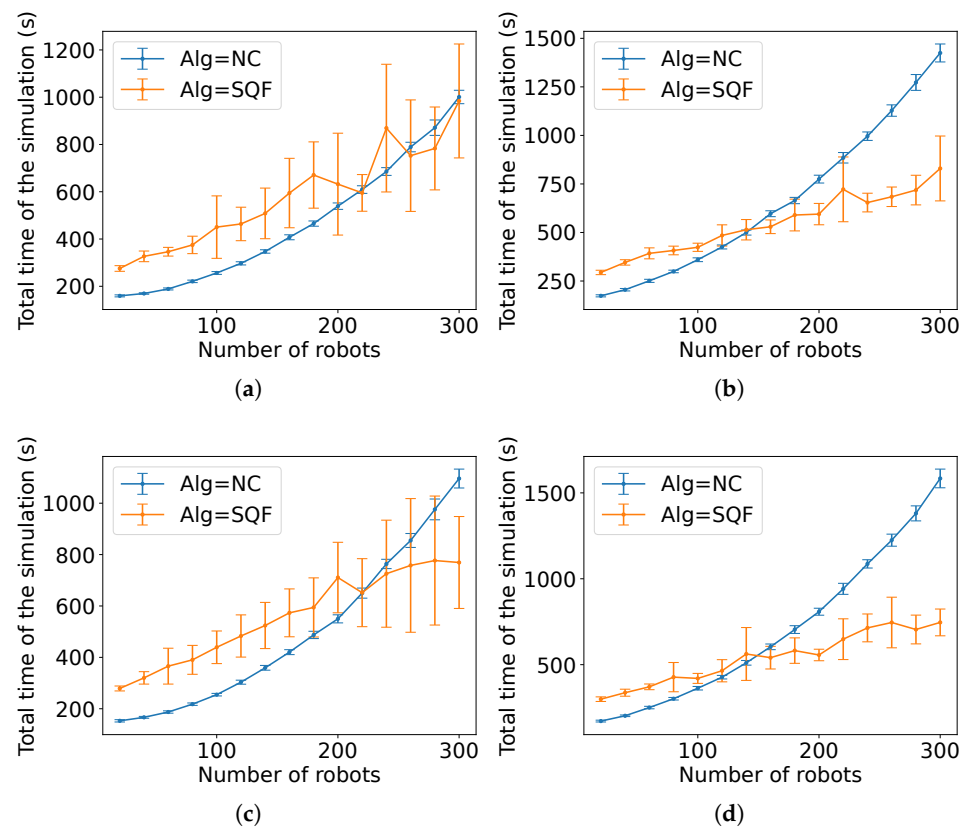


Figure A3. Cont.

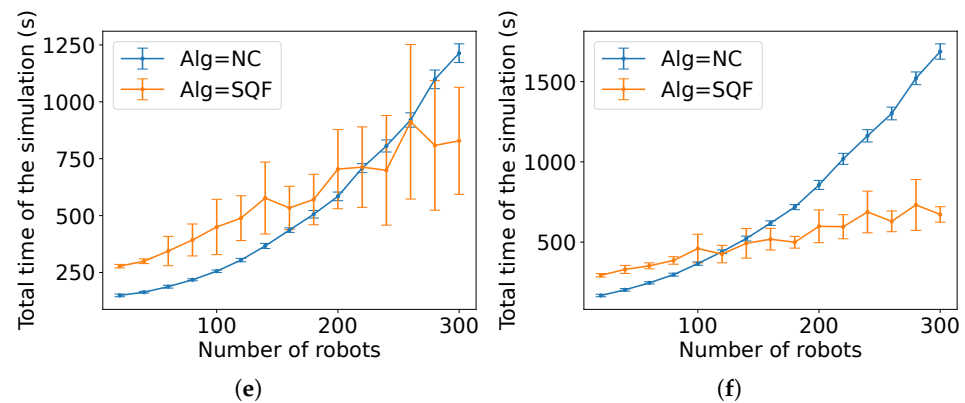


Figure A3. Comparison of the total simulation time by the number of robots for the experiments with MT using TRVF as the group B's algorithm and SQF and NC as the group A's algorithm by 70% of the robots for (a) holonomic and (b) non-holonomic robots, 80% for (c) holonomic and (d) non-holonomic robots and 90% for (e) holonomic and (f) non-holonomic robots.

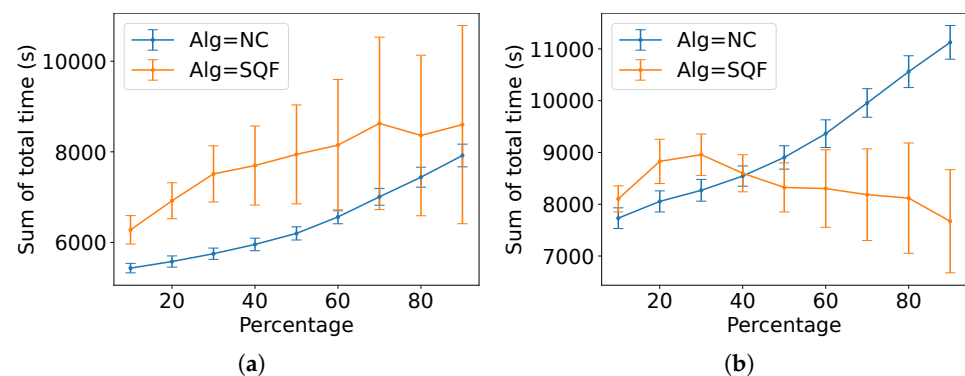


Figure A4. Comparison of the total simulation time sum over the number of robots by the percentage of robots in group A using NC and SQF as group A's algorithms. The swarm control algorithm of the robots in group B is TRVF. (a) Holonomic robots. (b) Non-holonomic robots.

Appendix A.2. Comparison of NC and TRVF for the Group A and SQF for Group B

Figures A5–A7 show the comparison of NC and TRVF as the group A algorithms for ratios varying from 10% to 90% in steps of 10% for holonomic and non-holonomic robots. In these experiments, SQF is the swarm control algorithm of group B. As before, these figures are summarised in Figure A8, which presents the sum over the number of robots of the average total simulation time by the ratio of robots in group A.

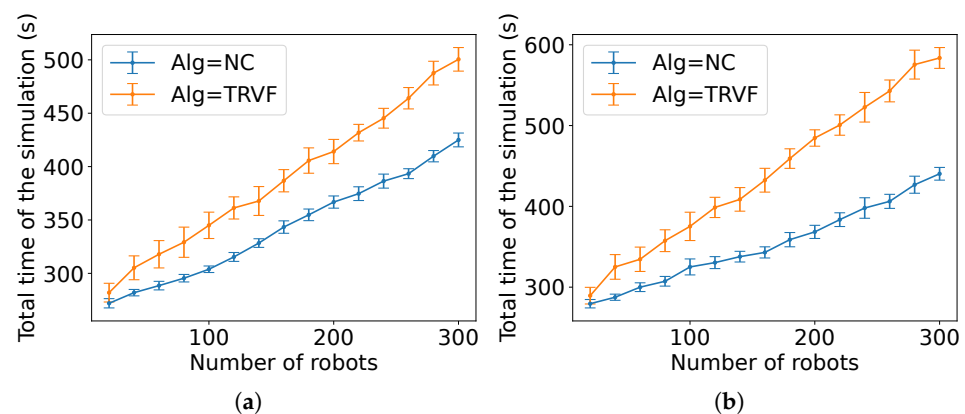


Figure A5. Cont.

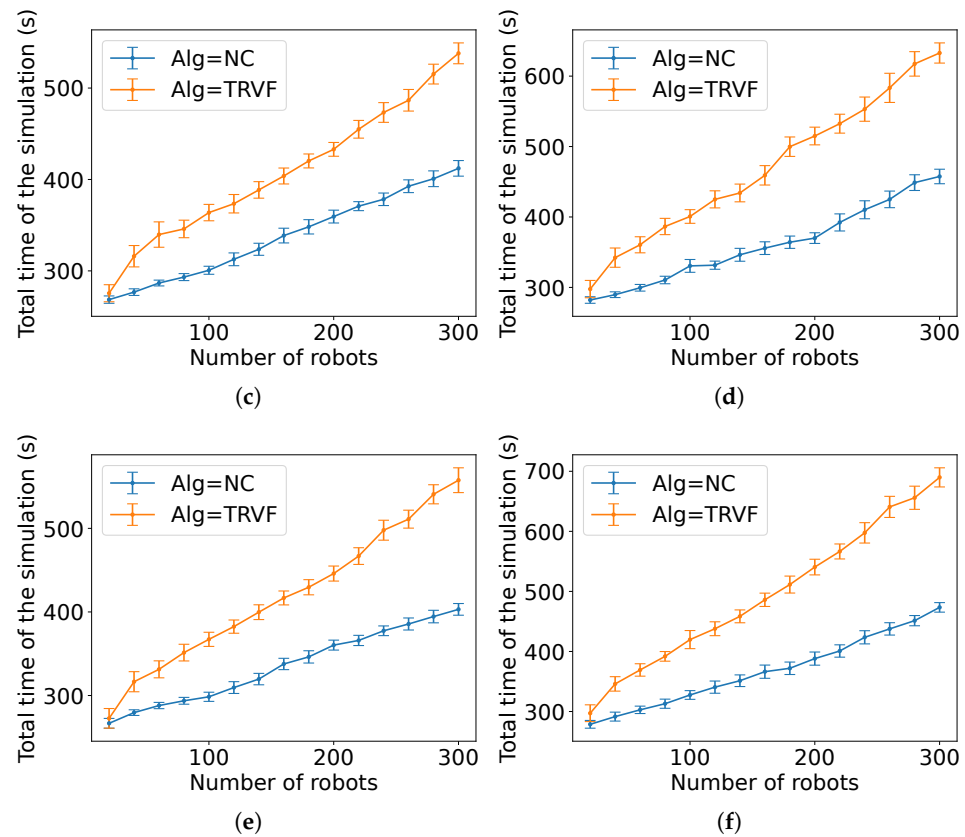


Figure A5. Comparison of the total simulation time by the number of robots for the experiments with MT using SQF as the group B's algorithm and TRVF and NC as the group A's algorithm by 10% of the robots for (a) holonomic and (b) non-holonomic robots, 20% for (c) holonomic and (d) non-holonomic robots and 30% for (e) holonomic and (f) non-holonomic robots.

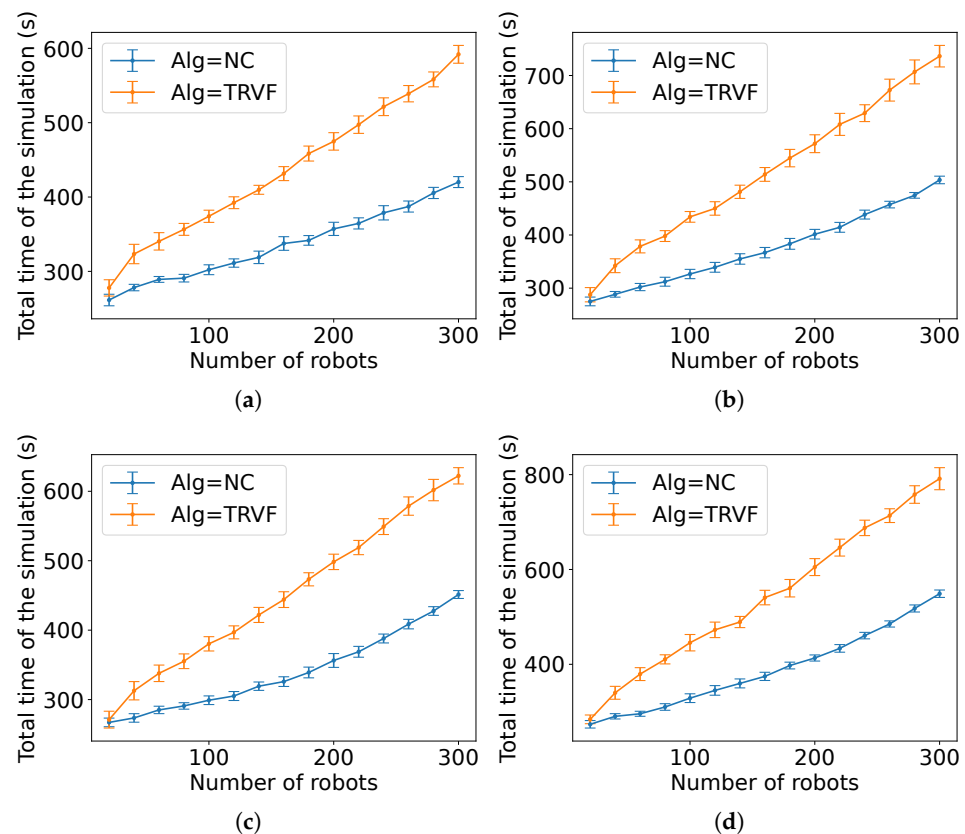


Figure A6. Cont.

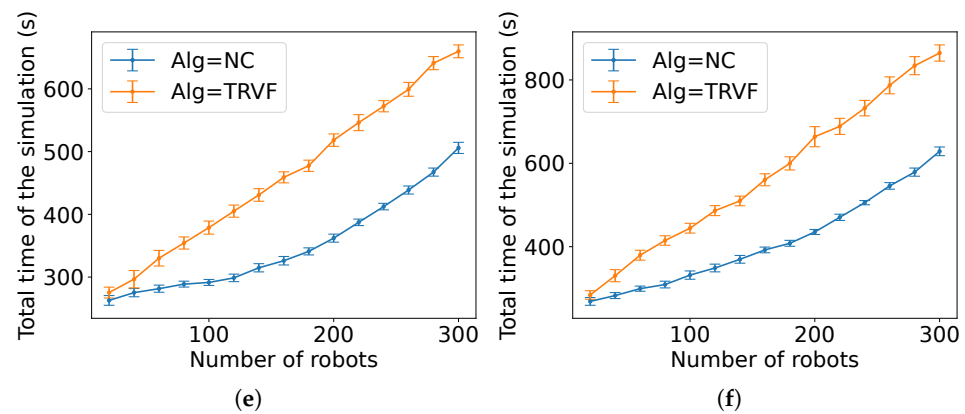


Figure A6. Comparison of the total simulation time by the number of robots for the experiments with MT using SQF as the group B's algorithm and TRVF and NC as the group A's algorithm by 40% of the robots for (a) holonomic and (b) non-holonomic robots, 50% for (c) holonomic and (d) non-holonomic robots and 60% for (e) holonomic and (f) non-holonomic robots.

From Figure A7, 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. When using the TRVF as a group A 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.

In addition, from Figure A8, NC as the group A algorithm, takes less or equal time than using TRVF on average for percentages from 10% to 80% for both types of robots with statistical significance.

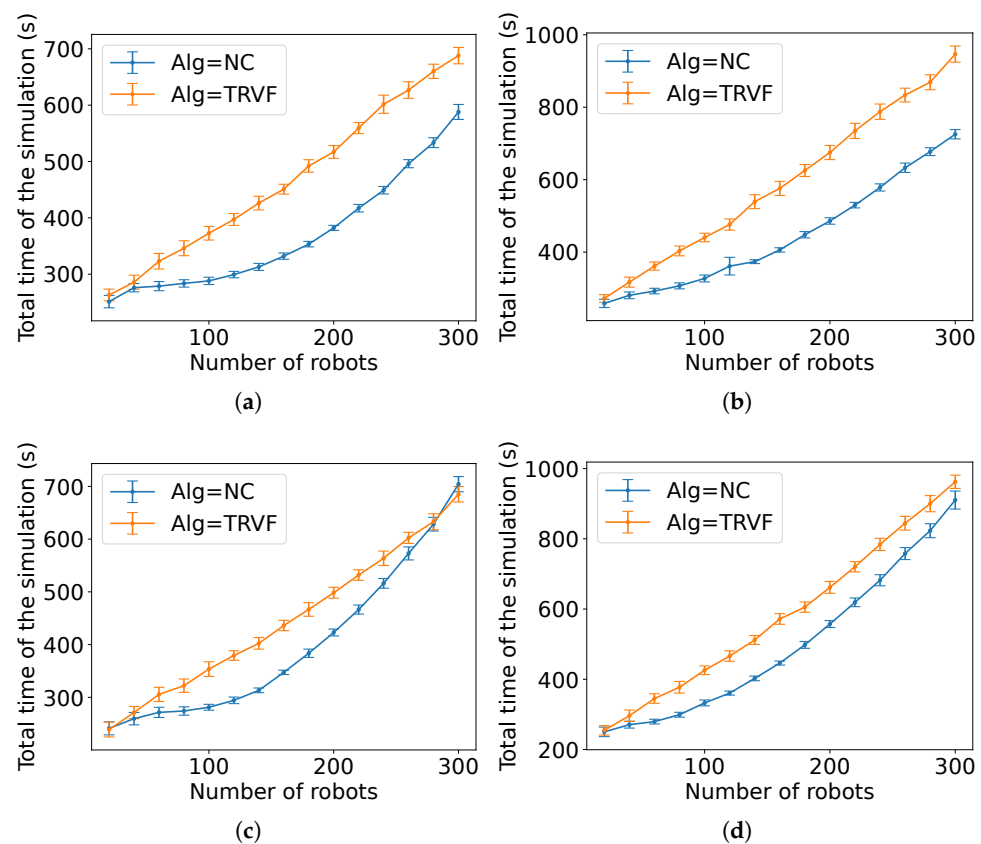


Figure A7. Cont.

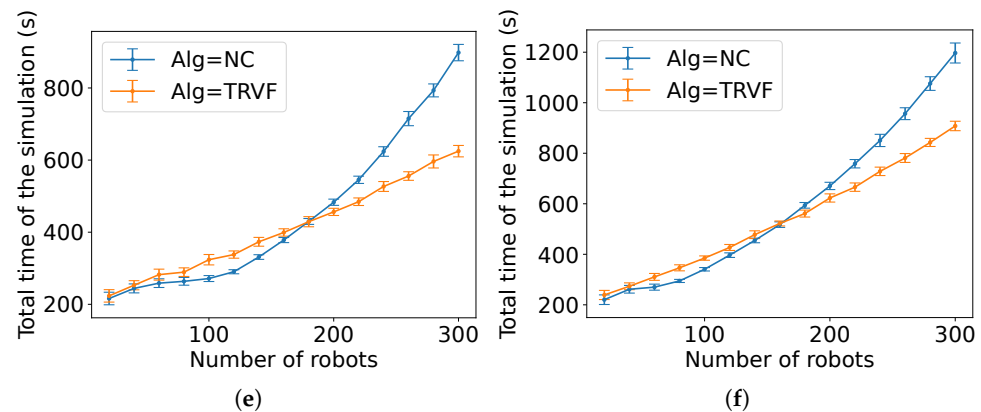


Figure A7. Comparison of the total simulation time by the number of robots for the experiments with MT using SQF as the group B's algorithm and TRVF and NC as the group A's algorithm by 70% of the robots for (a) holonomic and (b) non-holonomic robots, 80% for (c) holonomic and (d) non-holonomic robots and 90% for (e) holonomic and (f) non-holonomic robots.

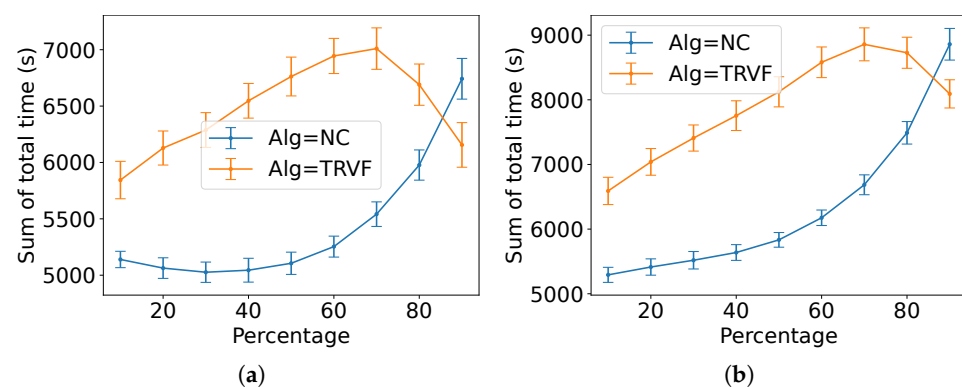


Figure A8. Comparison of the total simulation time sum over the number of robots by the percentage of robots in group A using NC and TRVF as group A's algorithms. The swarm control algorithm of the robots in group B is SQF. (a) Holonomic robots. (b) Non-holonomic robots.

Appendix A.3. Selection of Alg = NC

Based on the results of Appendices A.1 and A.2, although NC is the slowest algorithm for handling the common target problem for numerous robots [8], it was the best group A algorithm in most cases with statistical significance. Therefore, NC was selected as the default group A algorithm in the experiments. In addition, MT are expected to be used for small ratios of robots in group A when assuming it would be made of auxiliary robots not knowing the control algorithm of the other group in an ad hoc teamwork scenario. Remember, however, that SQF and TRVF still outperform NC for this problem as the swarm control algorithm [23], and the results only relate to the choice of group A's algorithm. Nevertheless, if the proportion of robots in group A and the estimated number of robots are known beforehand, the group A algorithm can be chosen based on the graphs.

Appendix B. Additional Graphs

Figures A9–A11 present estimations for MT with the proportion p of robots in group A from 20% to 90% in increments of 10% and SQF as the control algorithm of the robots in group B.

Figures A12–A14 show the estimations for MT with TRVF as the control algorithm of group B with p from 20% to 90% in steps of 10%.

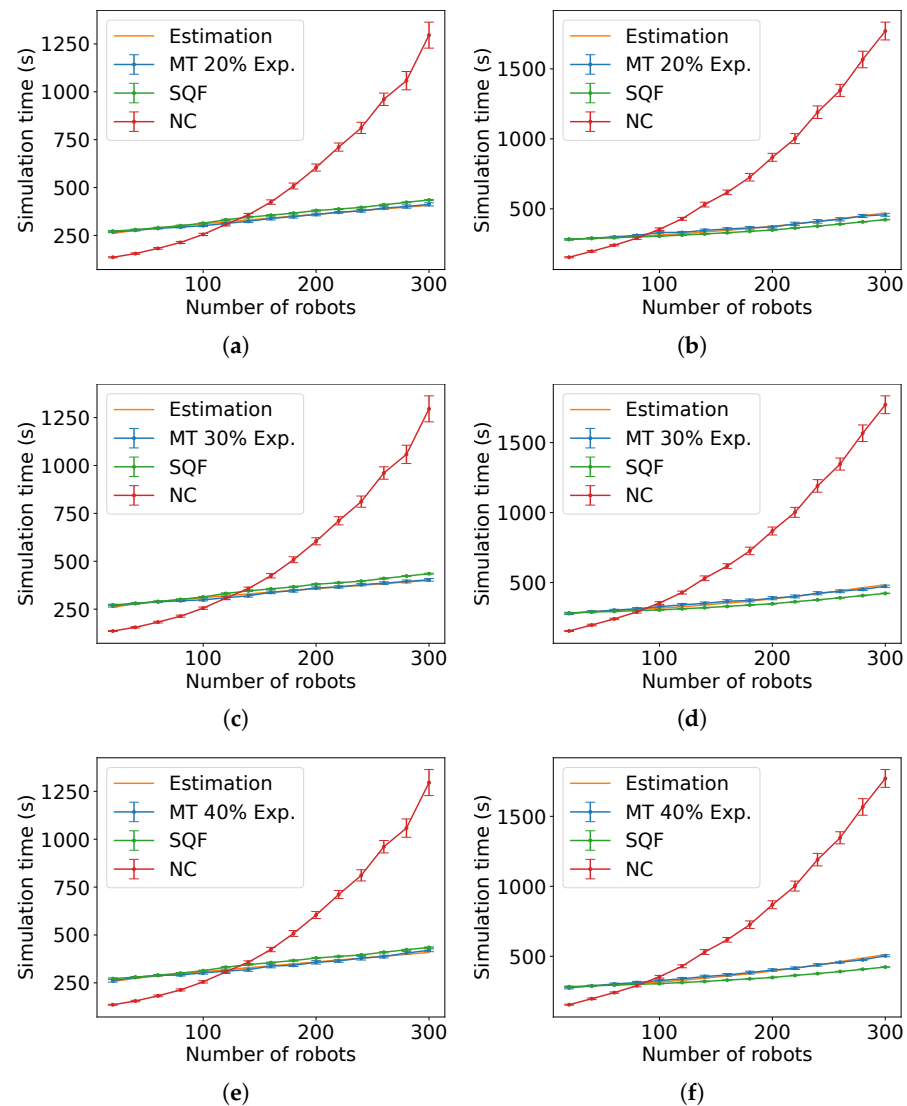


Figure A9. Comparison of the estimated expected time and the experimental data for MT with NC as group A's algorithm, group B is executing the SQF algorithm, $p = 20\%$ for (a) holonomic robots ($C_{MTNC} = -0.0091$ and $C_{MTB} = 0.9709$) and (b) non-holonomic robots ($C_{MTNC} = 0.0187$ and $C_{MTB} = 1.0246$), $p = 30\%$ for (c) holonomic robots ($C_{MTNC} = -0.0169$ and $C_{MTB} = 0.9757$) and (d) non-holonomic robots ($C_{MTNC} = 0.026$ and $C_{MTB} = 1.029$) and $p = 40\%$ for (e) holonomic robots ($C_{MTNC} = -0.0023$ and $C_{MTB} = 0.9569$) and (f) non-holonomic robots ($C_{MTNC} = 0.0486$ and $C_{MTB} = 1.0023$).

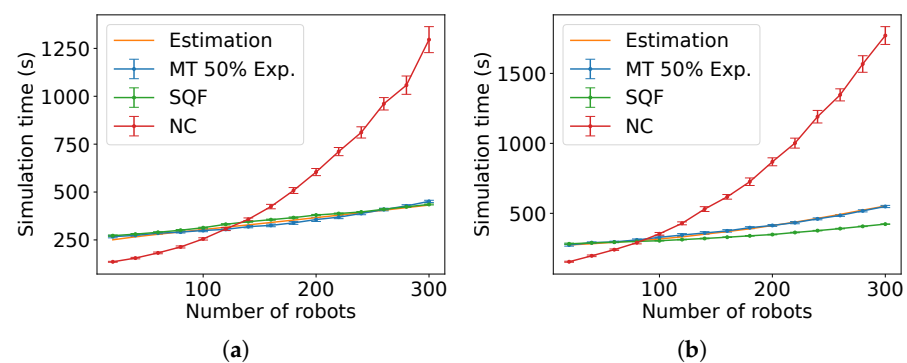


Figure A10. Cont.

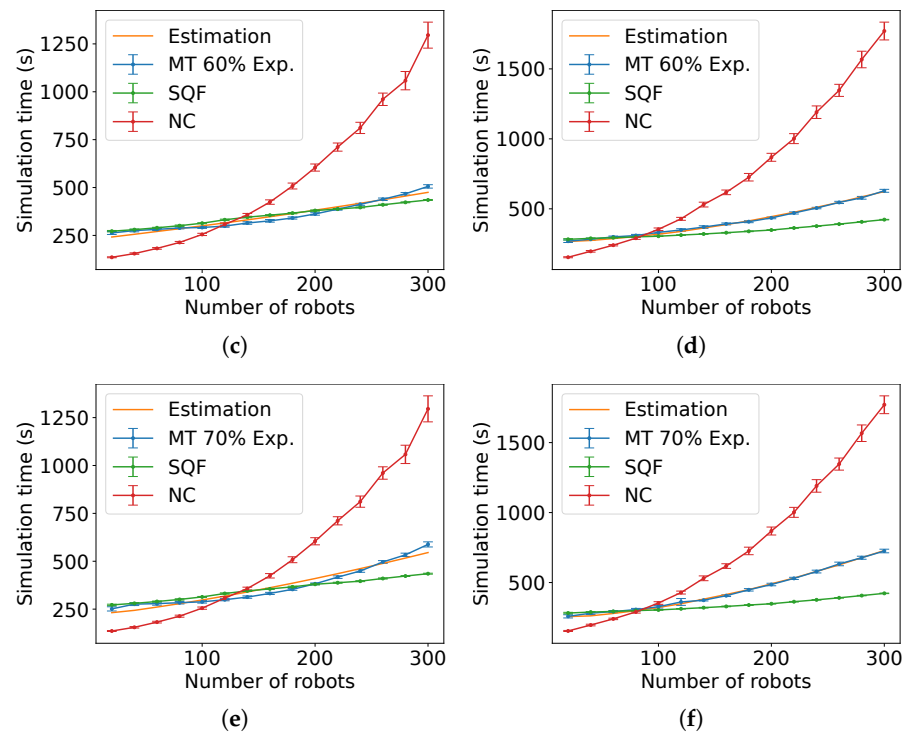


Figure A10. Comparison of the estimated expected time and the experimental data for MT with NC as group A's algorithm, group B is executing the SQF algorithm, $p = 50\%$ for (a) holonomic robots ($C_{MTNC} = 0.0328$ and $C_{MTB} = 0.9148$) and (b) non-holonomic robots ($C_{MTNC} = 0.0813$ and $C_{MTB} = 0.9682$), $p = 60\%$ for (c) holonomic robots ($C_{MTNC} = 0.093$ and $C_{MTB} = 0.8514$) and (d) non-holonomic robots ($C_{MTNC} = 0.1383$ and $C_{MTB} = 0.9088$) and $p = 70\%$ for (e) holonomic robots ($C_{MTNC} = 0.1842$ and $C_{MTB} = 0.7678$) and (f) non-holonomic robots ($C_{MTNC} = 0.2217$ and $C_{MTB} = 0.8241$).

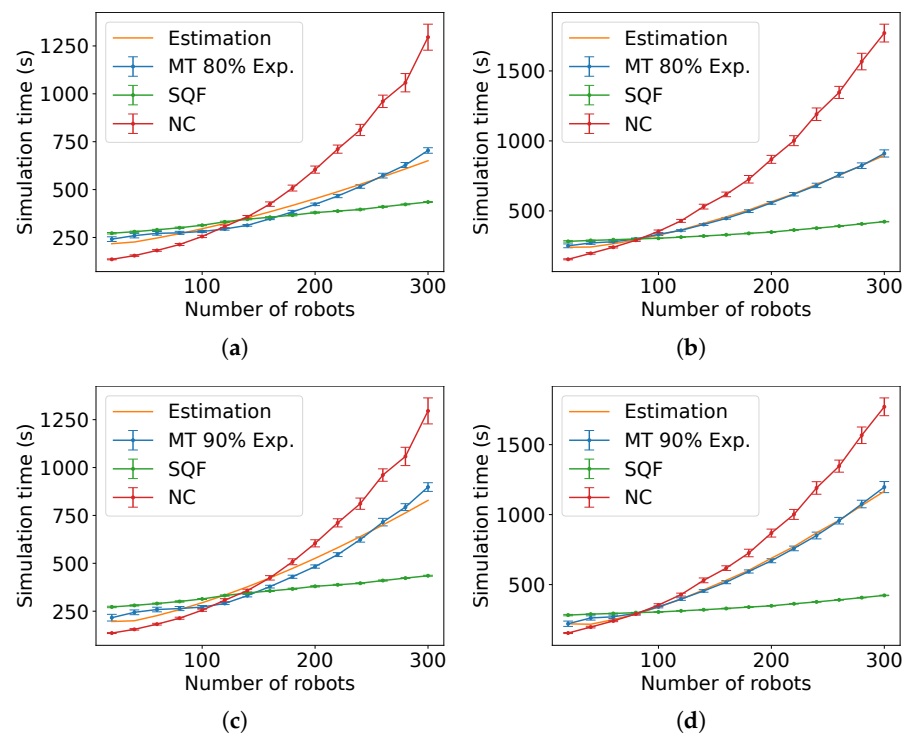


Figure A11. Comparison of the estimated expected time and the experimental data for MT with NC as group A's algorithm, $p = 80\%$, group B is executing the SQF algorithm, $p = 80\%$ for (a) holonomic robots ($C_{MTNC} = 0.3188$ and $C_{MTB} = 0.6471$) and (b) non-holonomic robots ($C_{MTNC} = 0.3557$ and $C_{MTB} = 0.6847$) and $p = 90\%$ for (c) holonomic robots ($C_{MTNC} = 0.5419$ and $C_{MTB} = 0.4552$) and (d) non-holonomic robots ($C_{MTNC} = 0.5644$ and $C_{MTB} = 0.4906$).

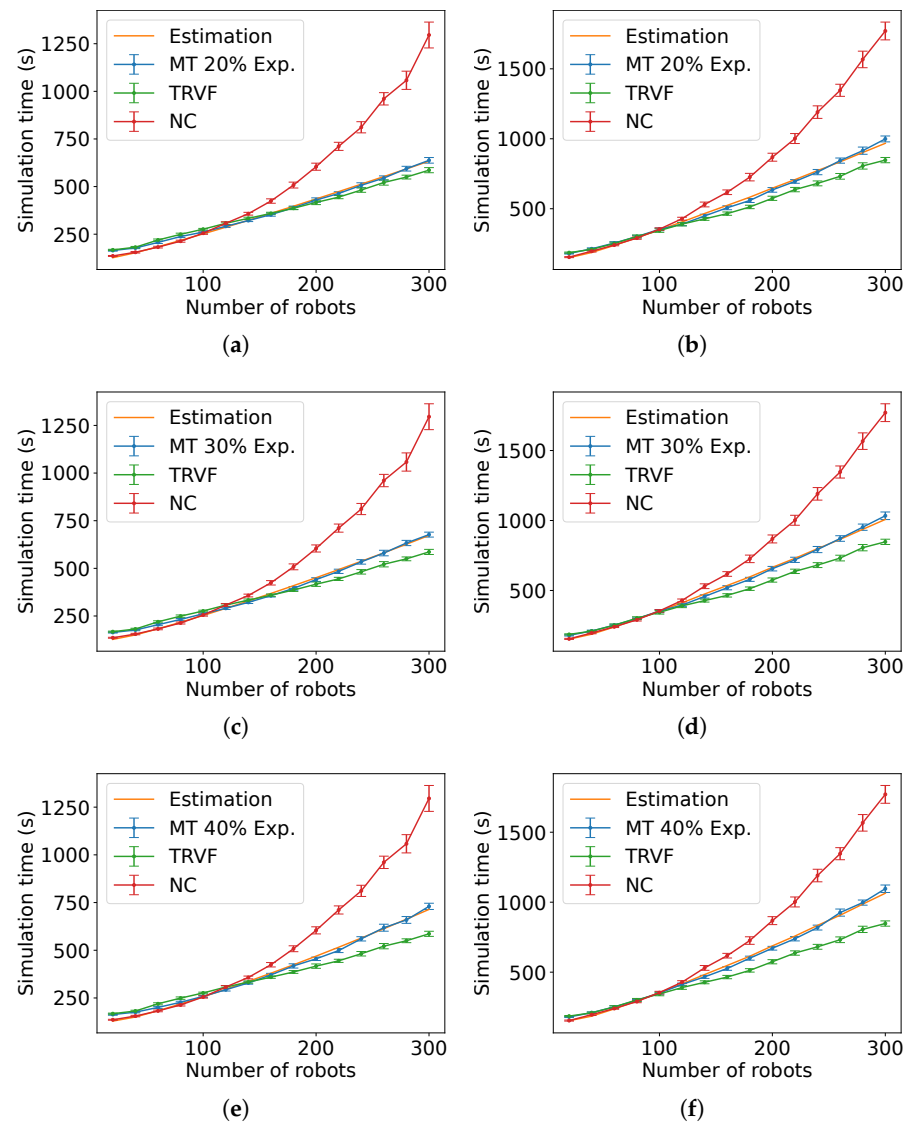


Figure A12. Comparison of the estimated expected time and the experimental data for MT with NC as group A's algorithm, group B is executing the TRVF algorithm, $p = 20\%$ for (a) holonomic robots ($C_{MTNC} = 0.1062$ and $C_{MTB} = 0.863$) and (b) non-holonomic robots ($C_{MTNC} = 0.1445$ and $C_{MTB} = 0.8724$), $p = 30\%$ for (c) holonomic robots ($C_{MTNC} = 0.1729$ and $C_{MTB} = 0.7976$) and (d) non-holonomic robots ($C_{MTNC} = 0.1791$ and $C_{MTB} = 0.8497$) and $p = 40\%$ for (e) holonomic robots ($C_{MTNC} = 0.2346$ and $C_{MTB} = 0.7458$) and (f) non-holonomic robots ($C_{MTNC} = 0.2395$ and $C_{MTB} = 0.7943$).

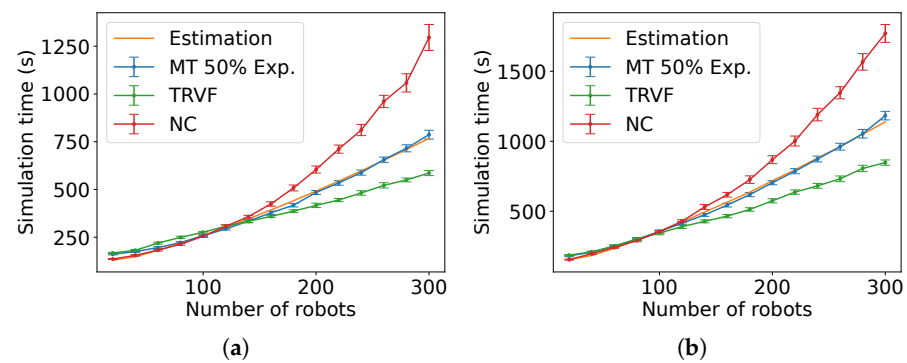


Figure A13. Cont.

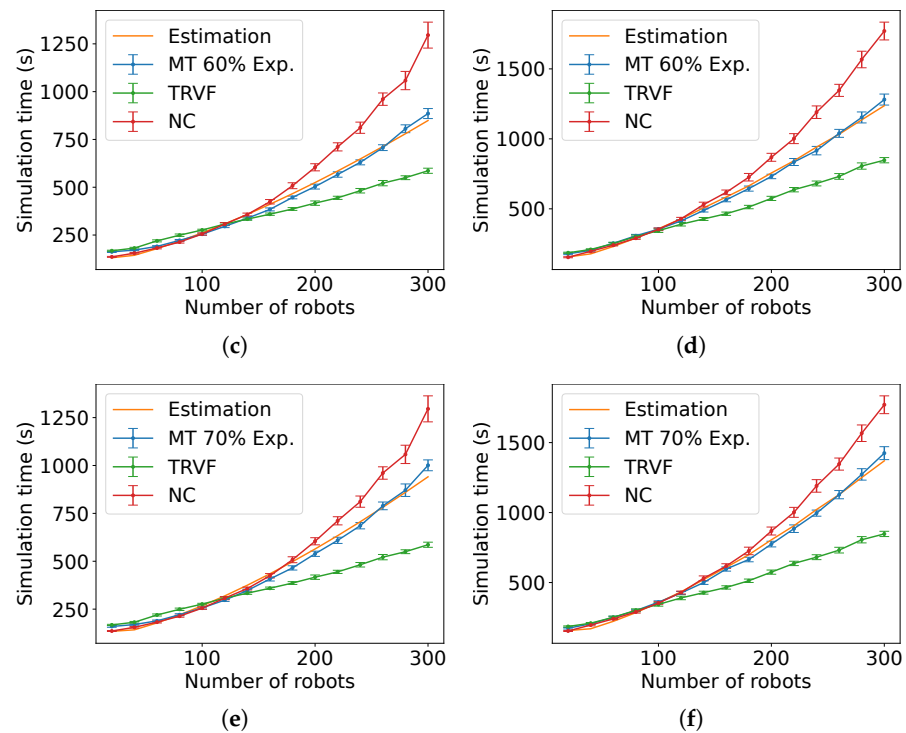


Figure A13. Comparison of the estimated expected time and the experimental data for MT with NC as group A's algorithm, group B is executing the TRVF algorithm, $p = 50\%$ for (a) holonomic robots ($C_{MTNC} = 0.3221$ and $C_{MTB} = 0.6633$) and (b) non-holonomic robots ($C_{MTNC} = 0.3178$ and $C_{MTB} = 0.7238$), $p = 60\%$ for (c) holonomic robots ($C_{MTNC} = 0.46$ and $C_{MTB} = 0.5282$) and (d) non-holonomic robots ($C_{MTNC} = 0.4279$ and $C_{MTB} = 0.6173$) and $p = 70\%$ for (e) holonomic robots ($C_{MTNC} = 0.603$ and $C_{MTB} = 0.4011$) and (f) non-holonomic robots ($C_{MTNC} = 0.5836$ and $C_{MTB} = 0.4588$).

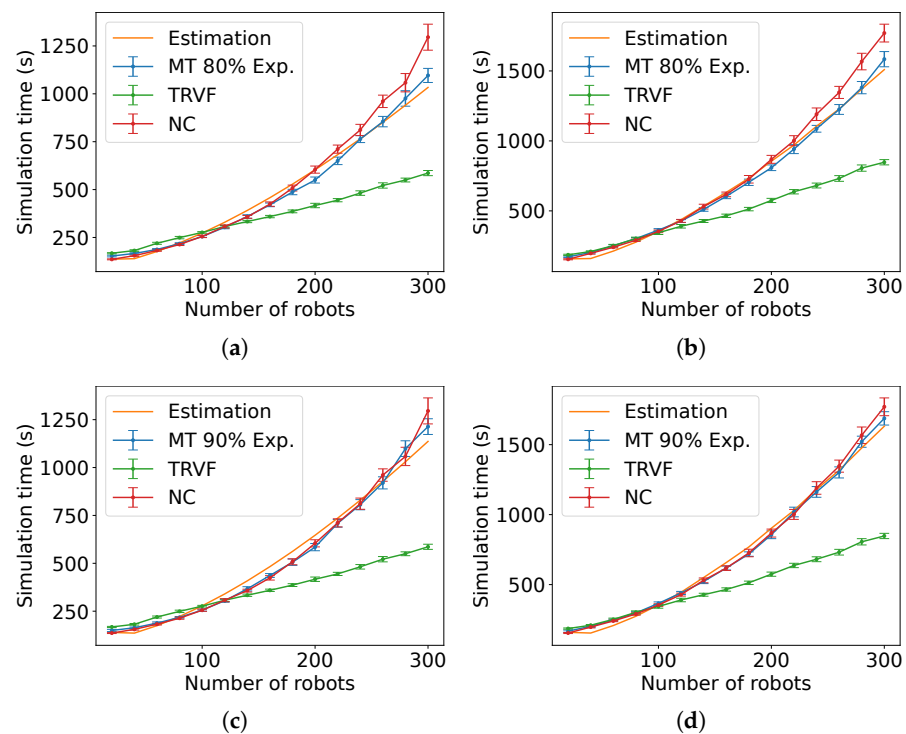


Figure A14. Comparison of the estimated expected time and the experimental data for MT with NC as group A's algorithm, group B is executing the TRVF algorithm, $p = 80\%$ for (a) holonomic robots ($C_{MTNC} = 0.7498$ and $C_{MTB} = 0.2673$) and (b) non-holonomic robots ($C_{MTNC} = 0.753$ and $C_{MTB} = 0.2806$) and $p = 90\%$ for (c) holonomic robots ($C_{MTNC} = 0.9173$ and $C_{MTB} = 0.1106$) and (d) non-holonomic robots ($C_{MTNC} = 0.8884$ and $C_{MTB} = 0.1511$).

References

1. Navarro, I.; Matía, F. An introduction to swarm robotics. *Int. Sch. Res. Not.* **2013**, *2013*, 608164. [\[CrossRef\]](#)
2. Sahin, E. Swarm Robotics: From Sources of Inspiration to Domains of Application. In *Swarm Robotics: SAB 2004 International Workshop, Santa Monica, CA, USA, 17 July 2004*; Sahin, E., Spears, W.M., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 10–20. [\[CrossRef\]](#)
3. Beni, G.; Wang, J. Swarm Intelligence in Cellular Robotic Systems. In *Robots and Biological Systems: Towards a New Bionics? Proceedings of the NATO Advanced Workshop on Robots and Biological Systems, Held at Il Ciocco, Toscana, Italy, 26–30 June 1989*; Dario, P., Sandini, G., Aebischer, P., Eds.; Springer: Berlin/Heidelberg, Germany, 1993; pp. 703–712. [\[CrossRef\]](#)
4. Dudek, G.; Jenkin, M.; Milios, E.; Wilkes, D. A taxonomy for swarm robots. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Online, 26–30 July 1993; Volume 1, pp. 441–447. [\[CrossRef\]](#)
5. Garnier, S.; Gautrais, J.; Theraulaz, G. The biological principles of swarm intelligence. *Swarm Intell.* **2007**, *1*, 3–31. [\[CrossRef\]](#)
6. Mataric, M.J. Designing and Understanding Adaptive Group Behavior. *Adapt. Behav.* **1995**, *4*, 51–80. [\[CrossRef\]](#)
7. Panait, L.; Luke, S. Evolving Foraging Behaviors. In Proceedings of the Second International Workshop on the Mathematics and Algorithms of Social Insects, Atlanta, GA, USA, 15–17 December 2003; pp. 131–138.
8. Marcolino, L.S.; Passos, Y.T.; Souza, A.A.F.d.; Rodrigues, A.d.S.; Chaimowicz, L. Avoiding Target Congestion on the Navigation of Robotic Swarms. *Auton. Robot.* **2017**, *41*, 1297–1320. [\[CrossRef\]](#)
9. Chaudhari, M.S.; Patil, B.; Raut, V. IoT based Waste Collection Management System for Smart Cities: An Overview. In Proceedings of the 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 27–29 March 2019; pp. 802–805. [\[CrossRef\]](#)
10. Mărgăritescu, M.; Ancuța, P.N.; Canale, E.V.; Stanciu, D.I.; Dumitriu, D.; Brișan, C.M. Control of an Autonomous Mobile Waste Collection Robot. In Proceedings of the International Conference of Mechatronics and Cyber—MixMechatronics, Bucharest, Romania, 5–6 September 2019; Gheorghe, G.I., Ed.; Springer: Cham, Switzerland, 2020; pp. 51–63.
11. Costa, L.R.; Aloise, D.; Gianoli, L.G.; Lodi, A. OptiMaP: Swarm-powered Optimized 3D Mapping Pipeline for emergency response operations. In Proceedings of the 18th International Conference on Distributed Computing in Sensor Systems (DCOSS), Marina Del Rey, LA, USA, 30 May–1 June 2022; pp. 269–276. [\[CrossRef\]](#)
12. Stewart, R.L.; Russell, R.A. A Distributed Feedback Mechanism to Regulate Wall Construction by a Robotic Swarm. *Adapt. Behav.* **2006**, *14*, 21–51. [\[CrossRef\]](#)
13. Liu, J.; Siragam, V.; Gong, Z.; Chen, J.; Fridman, M.D.; Leung, C.; Lu, Z.; Ru, C.; Xie, S.; Luo, J.; et al. Robotic Adherent Cell Injection for Characterizing Cell–Cell Communication. *IEEE Trans. Biomed. Eng.* **2015**, *62*, 119–125. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Galstyan, A.; Hogg, T.; Lerman, K. Modeling and mathematical analysis of swarms of microscopic robots. In Proceedings of the IEEE Swarm Intelligence Symposium (SIS), Pasadena, CA, USA, 8–10 June 2005; pp. 201–208. [\[CrossRef\]](#)
15. Bogdan, P.; Wei, G.; Marculescu, R. Modeling populations of micro-robots for biological applications. In Proceedings of the IEEE International Conference on Communications (ICC), Ottawa, ON, Canada, 10–15 June 2012; pp. 6188–6192. [\[CrossRef\]](#)
16. Wang, B.; Liu, D.; Liao, Y.; Huang, Y.; Ni, M.; Wang, M.; Ma, Z.; Wu, Z.; Lu, Y. Spatiotemporally Actuated Hydrogel by Magnetic Swarm Nanorobotics. *ACS Nano* **2022**, *16*, 20985–21001. [\[CrossRef\]](#) [\[PubMed\]](#)
17. Arnold, R.D.; Yamaguchi, H.; Tanaka, T. Search and rescue with autonomous flying robots through behavior-based cooperative intelligence. *J. Int. Humanit. Action* **2018**, *3*, 18. [\[CrossRef\]](#)
18. Wen, J.; He, L.; Zhu, F. Swarm Robotics Control and Communications: Imminent Challenges for Next Generation Smart Logistics. *IEEE Commun. Mag.* **2018**, *56*, 102–107. [\[CrossRef\]](#)
19. Yogi, G.B.; Bhasha, S.M.; Ram, C.S.; Latha, M.P. Swarm Robots for Warehouse Management. *Int. J. Res. Eng. Sci. Manag.* **2020**, *3*, 706–709.
20. Khaluf, Y.; Dorigo, M. Modeling Robot Swarms Using Integrals of Birth-Death Processes. *ACM Trans. Auton. Adapt. Syst.* **2016**, *11*, 8. [\[CrossRef\]](#)
21. Marcolino, L.S.; Chaimowicz, L. Traffic control for a swarm of robots: Avoiding target congestion. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), St. Louis, MO, USA, 10–15 October 2009; pp. 1955–1961. [\[CrossRef\]](#)
22. Passos, Y.T.d.; Duquesne, X.; Marcolino, L.S. On the Throughput of the Common Target Area for Robotic Swarm Strategies. *Mathematics* **2022**, *10*, 2482. [\[CrossRef\]](#)
23. Passos, Y.T.d.; Duquesne, X.; Marcolino, L.S. Congestion control algorithms for robotic swarms with a common target based on the throughput of the target area. *Robot. Auton. Syst.* **2023**, *159*, 104284. [\[CrossRef\]](#)
24. Cai, Y.; Guo, E.; Li, W.; Gao, H. A Self-organized Circular Formation Algorithm for Planar Robot Swarm. In Proceedings of the 5th Chinese Conference on Swarm Intelligence and Cooperative Control (CCSICC), Shenzhen, China, 19–22 November 2021; Ren, Z., Wang, M., Hua, Y., Eds.; Springer: Singapore, 2023; pp. 702–710. [\[CrossRef\]](#)
25. Barnes, L.; Fields, M.; Valavanis, K. Unmanned ground vehicle swarm formation control using potential fields. In Proceedings of the Mediterranean Conference on Control and Automation (MED), Athens, Greece, 27–29 June 2007; pp. 1–8. [\[CrossRef\]](#)

26. Galvez, R.L.; Faelden, G.E.U.; Maningo, J.M.Z.; Nakano, R.C.S.; Dadios, E.P.; Bandala, A.A.; Vicerra, R.R.P.; Fernando, A.H. Obstacle avoidance algorithm for swarm of quadrotor unmanned aerial vehicle using artificial potential fields. In Proceedings of the IEEE Region 10 Conference (TENCON), Penang, Malaysia, 5–8 November 2017; pp. 2307–2312. [\[CrossRef\]](#)
27. Patwardhan, A.; Murai, R.; Davison, A.J. Distributing Collaborative Multi-Robot Planning with Gaussian Belief Propagation. *IEEE Robot. Autom. Lett.* **2023**, *8*, 552–559. [\[CrossRef\]](#)
28. Opper, M.; Saad, D. *Advanced Mean Field Methods: Theory and Practice*; Neural Information Processing Series; MIT Press: Cambridge, MA, USA, 2001.
29. Chen, Y.; Georgiou, T.T.; Pavon, M. Steering the Distribution of Agents in Mean-Field Games System. *J. Optim. Theory Appl.* **2018**, *179*, 332–357. [\[CrossRef\]](#)
30. Lasry, J.M.; Lions, P.L. Mean field games. *Jpn. J. Math.* **2007**, *2*, 229–260. [\[CrossRef\]](#)
31. Zhang, Y.; Ding, M.; Yuan, Y.; Zhang, J.; Yang, Q.; Shi, G.; Jiang, J. Large-scale UAV swarm path planning based on mean-field reinforcement learning. *Chin. J. Aeronaut.* **2025**, *38*, 103484. [\[CrossRef\]](#)
32. Ménec, S.L. Swarm Guidance Based on Mean Field Game Concepts. *Int. Game Theory Rev.* **2024**, *26*, 2440008:1–2440008:18. [\[CrossRef\]](#)
33. Lin, M.; Yuan, G.; Liu, D. Federated Learning Adaptive Dynamic Programming for Massive Multiagent Mean-Field Games-Based Optimal Consensus. *IEEE Trans. Cybern.* **2025**, 1–12. [\[CrossRef\]](#)
34. Boudec, J.Y.L.; McDonald, D.; Mundinger, J. A Generic Mean Field Convergence Result for Systems of Interacting Objects. In Proceedings of the Fourth International Conference on the Quantitative Evaluation of Systems (QEST 2007), Edinburgh, UK, 17–19 September 2007; pp. 3–18. [\[CrossRef\]](#)
35. Beavers, G.; Hexmoor, H. Teams of agents. In Proceedings of the 2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace, Tucson, AZ, USA, 7–10 October 2001; Volume 1, pp. 574–582. [\[CrossRef\]](#)
36. dos Santos, R.F.; Ramachandran, R.K.; Vieira, M.A.; Sukhatme, G.S. Parallel multi-speed Pursuit-Evasion Game algorithms. *Robot. Auton. Syst.* **2023**, *163*, 104382. [\[CrossRef\]](#)
37. Salimi, M.; Ibragimov, G.I.; Siegmund, S.; Sharifi, S. On a Fixed Duration Pursuit Differential Game with Geometric and Integral Constraints. *Dyn. Games Appl.* **2016**, *6*, 409–425. [\[CrossRef\]](#)
38. Flammini, B.; Azzalini, D.; Amigoni, F. 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, London, UK, 29 May–2 June 2023; pp. 2562–2564.
39. Gerkey, B.P.; Vaughan, R.T.; Howard, A. The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. In Proceedings of the 11th International Conference on Advanced Robotics (ICAR), Coimbra, Portugal, 30 June–3 July 2003; pp. 317–323.
40. Soni, A.; Dasannacharya, C.; Gautam, A.; Shekhawat, V.S.; Mohan, S. Multi-Robot Unknown Area Exploration Using Frontier Trees. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; pp. 9934–9941. [\[CrossRef\]](#)
41. Calderón-Arce, C.; Brenes-Torres, J.C.; Solis-Ortega, R. Swarm Robotics: Simulators, Platforms and Applications Review. *Computation* **2022**, *10*, 80. [\[CrossRef\]](#)
42. Miteva, L.; Yovchev, K.; Chikurtev, D. Software and Hardware Infrastructure for Research and Development of Intelligent Control for Robotic Manipulators. In Proceedings of the 2022 XXXI International Scientific Conference Electronics (ET), Sozopol, Bulgaria, 13–15 September 2022; pp. 1–5. [\[CrossRef\]](#)
43. Boogerd, F.C.; Bruggeman, F.J.; Richardson, R.C.; Stephan, A.; Westerhoff, H.V. Emergence and Its Place in Nature: A Case Study of Biochemical Networks. *Synthese* **2005**, *145*, 131–164. [\[CrossRef\]](#)
44. Holland, J.H. *Emergence: From Chaos to Order*; Oxford University Press: Oxford, UK, 1998.
45. Feltz, B.; Crommelinck, M.; Goujon, P. *Self-Organization and Emergence in Life Sciences*; Synthese Library; Springer: Dordrecht, The Netherlands, 2006.
46. Couzin, I. Geometric Principles of Individual and Collective Decision-Making. Keynote Talk Presented at the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2023), London, UK, 29 May–2 June 2023. Available online: <https://www.southampton.ac.uk/~eg/AAMAS2023/pdfs/p3.pdf> (accessed on 11 July 2023).
47. Haken, H. *Synergetics: An Introduction: Nonequilibrium Phase Transitions and Self-Organization in Physics, Chemistry, and Biology*; Springer: Berlin/Heidelberg, Germany, 1983.
48. March, N.; Young, W.; Sampanthar, S. *The Many-Body Problem in Quantum Mechanics*; Dover Books on Physics; Dover Publications: London, UK, 1995.
49. Kuzemsky, A. *Statistical Mechanics and the Physics of Many-Particle Model Systems*; World Scientific Publishing Company: Singapore, 2017.
50. Mandl, F. *Statistical Physics*; Manchester Physics Series; Wiley: Bristol, UK, 1971.

51. Huang, K. *Statistical Mechanics*, 2nd ed.; John Wiley & Sons: Toronto, ON, Canada, 1987.
52. Kato, T. *A Short Introduction to Perturbation Theory for Linear Operators*; Springer: New York, NY, USA, 2012.
53. Holmes, M. *Introduction to Perturbation Methods*; Texts in Applied Mathematics, Springer: New York, NY, USA, 1998.
54. Bartlett, R.J.; Musiał, M. Coupled-cluster theory in quantum chemistry. *Rev. Mod. Phys.* **2007**, *79*, 291–352. [[CrossRef](#)]
55. Shavitt, I.; Bartlett, R. *Many-Body Methods in Chemistry and Physics: MBPT and Coupled-Cluster Theory*; Cambridge Molecular Science, Cambridge University Press: Cambridge, UK, 2009.
56. Glossman-Mitnik, D. *Density Functional Theory*; IntechOpen: London, UK, 2019.
57. Dreizler, R.M.; Gross, E.K.U. *Density Functional Theory*; NATO ASI Series. Series B, Physics; Plenum Press: New York, NY, USA, 1995; Volume 337.
58. Jiao, C.; Liang, J.C.; Yu, Z.F.; Chen, Y.; Zhang, A.X.; Xue, J.K. Bose—Einstein condensates with tunable spin—Orbit coupling in the two-dimensional harmonic potential: The ground-state phases, stability phase diagram and collapse dynamics. *Front. Phys.* **2022**, *17*, 61503. [[CrossRef](#)]
59. Pérez-Torres, J.F. Dilemma of the “Best Wavefunction”: Comparing Results of the STO-NG Procedure versus the Linear Variational Method. *J. Chem. Educ.* **2019**, *96*, 704–707. [[CrossRef](#)]
60. Carleo, G.; Troyer, M. Solving the quantum many-body problem with artificial neural networks. *Science* **2017**, *355*, 602–606. [[CrossRef](#)]
61. Mannone, M.; Seidita, V.; Chella, A. Quantum computing for swarm robotics: A local-to-global approach. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2025**, *383*, 20240139. [[CrossRef](#)]
62. Hamann, H. *Swarm Robotics: A Formal Approach*; Springer: Berlin/Heidelberg, Germany, 2018.
63. Prigogine, I.; Andrews, F.C. A Boltzmann-Like Approach for Traffic Flow. *Oper. Res.* **1960**, *8*, 789–797. [[CrossRef](#)]
64. Iannini, M.L.L.; Dickman, R. Kinetic theory of vehicular traffic. *Am. J. Phys.* **2016**, *84*, 135–145. [[CrossRef](#)]
65. Prigogine, I.; Herman, R. *Kinetic Theory of Vehicular Traffic*; Elsevier: New York, NY, USA, 1971.
66. Elamvazhuthi, K.; Berman, S. Mean-field models in swarm robotics: A survey. *Bioinspiration Biomim.* **2019**, *15*, 015001. [[CrossRef](#)] [[PubMed](#)]
67. Morin, D. *Introduction to Classical Mechanics: With Problems and Solutions*; Cambridge University Press: Cambridge, UK, 2007; p. 719.
68. Siegwart, R.; Nourbakhsh, I.R. *Introduction to Autonomous Mobile Robots*; Bradford Company: Scituate, MA, USA, 2004.
69. Casella, G.; Berger, R. *Statistical Inference*, 2nd ed.; Duxbury-Thomson Learning: Pacific Grove, CA, USA, 2002.
70. Nelson, D.; Barber, D.; McLain, T.; Beard, R. Vector field path following for small unmanned air vehicles. In Proceedings of the American Control Conference (ACC), Minneapolis, MN, USA, 14–16 June 2006; pp. 5788–5794. [[CrossRef](#)]
71. Stirling, T.; Floreano, D. Energy-Time Efficiency in Aerial Swarm Deployment. In *Distributed Autonomous Robotic Systems: The 10th International Symposium*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 5–18. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.