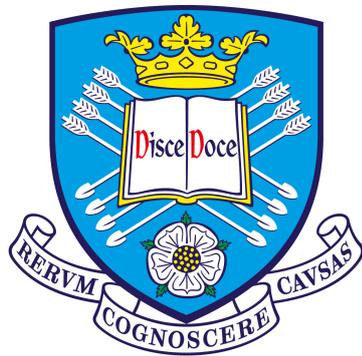


Simple individual behavioural rules for improving the collective behaviours of robot swarms



The
University
Of
Sheffield.

Mohamed Salaheddine Talamali

Supervisors: Prof James A. R. Marshall
Dr Andreagiovanni Reina

Department of Computer Science
University of Sheffield

This thesis is submitted for the degree of
Doctor of Philosophy

July 2020

To my parents, my wife, and my siblings

Acknowledgements

Completing this PhD thesis would not have been possible without the help and support of many persons. Here, I would like to acknowledge and thank these persons for their precious support.

I am extremely grateful to Dr Andreagiovanni Reina for the guidance and support he offered me during my PhD. I want to thank him for being present whenever I needed him for both research and daily life difficulties, for always caring about my success and for continually aiming at making me a better researcher. I very much enjoyed our productive brainstorming sessions, where we looked for solutions and novel ideas. I will never forget the robot experiments we conducted together under pressure to meet deadlines.

I would also like to extend my deepest gratitude to Prof James Marshall for supervising and supporting me during my PhD. I want to thank him for always pointing me toward exciting research directions and giving me advice on improving the quality of my research. I also thank him for financially supporting me during my PhD, by covering most of my tuition fees and offering me a studentship, through his European Research Council (ERC) Grant "DiODE: Distributed Algorithms for Optimal Decision-Making" (grant agreement number 647704).

I am also grateful to the department of computer science at the University of Sheffield for covering part of my tuition fees.

Special thanks to Michael Port, or as I like to call him "the best technician in the world.", for the enormous technical support. The real robots experiments would not have been possible without his help. I also wish to thank Dr Arindam Saha for the fruitful collaborations and for generously sharing knowledge. Thanks should also go to Dr Alex Cope and Dr Thomas Bose for the precious technical support, insightful suggestions, and inspiring discussions.

I gratefully acknowledge the precious help I received from the support and admin staff at the University of Sheffield during my PhD. Special thanks go to Joanne Seal for the valuable help in conferences booking and visa applications.

Finally, I am deeply and sincerely grateful to my family for their endless and exceptional love, help and support. I wish to thank my parents and my wife for their encouragement and support throughout the PhD. I will always be indebted to my parents for teaching me the importance of studying, and for always prioritising my success over their personal needs despite difficulties. I also thank my siblings for always being there for me.

Abstract

Swarm robotics is an ongoing area of research that is expected to revolutionise various real-world domains such as agriculture and space exploration. Swarm robotics systems are composed of a large number of simple and autonomous robots. Each robot locally interacts with other robots and with the environment following a set of behavioural rules. These individual interactions enable the swarm to exhibit interesting collective behaviours and to accomplish specific tasks. The main challenge in designing robot swarms is to determine the behavioural rules that each robot should follow so that the swarm as a whole can perform the desired task. The performance of robot swarms in a given task depends on the designer's choice of appropriate individual behavioural rules. In this thesis, we investigate simple individual behavioural rules for improving the performance of robot swarms in two major tasks. Using simple behavioural rules makes the designed solutions possibly usable with simpler platforms such as micro- and nanorobots.

The first task we address is known as the best-of- n decision problem where the swarm is required to select the best option among n available alternatives. Solving the best-of- n decision problem is considered to be a fundamental cognitive skill for robot swarms as it influences the swarm's success in other tasks. In this thesis, we introduce individual behavioural rules to improve the performance of robot swarms in the best-of- n problem. Through these rules, robots vary their interaction strength over time in a decentralised fashion to balance the acquisition and the dissemination of information. The proposed behavioural rules allow swarms of simple noisy robots with constrained communication to limit the effect of individual errors and make highly accurate collective decisions in a predictable time. In some scenarios where the best option changes over time, the swarm is required to switch its decision accordingly. In this thesis, we introduce individual behavioural rules through which the robots process new information and discard outdated beliefs. These behavioural rules enable robot swarms to adapt their decisions to various environmental changes, including the appearance of better choices or the disappearance of the current swarm's choice. Our analysis shows that relying on local communication is more favourable for achieving adaptation. This result highlights the benefit of the local sensing and communication characterising biological and artificial swarms.

The second task we address in this thesis is the collective resource collection task. In this task, the robots are asked to retrieve objects that are clustered at unknown locations in the environment. We address this task because of its numerous potential real-world applications. In many of these applications, the objects to collect are assigned different importance or value. In this thesis, we introduce a bio-inspired individual behaviour that allows robot swarms to perform quality-based resource collection. Similarly to foraging ants, in our proposed behaviour, the robots coordinate their collection efforts by laying and sensing virtual pheromone trails. The use of pheromone trails offers an advantageous implementation of the memory and communication capabilities necessary for the efficient collection of clustered objects. The proposed behaviour allows robot swarms to satisfy various collection objectives and achieve an optimal resource collection behaviour in the case of relatively small swarms.

In this thesis, we analyse swarm robotics systems using both minimalistic tools such as stochastic and multi-agent simulations, and more advanced tools such as physics-based simulations and real robot experiments. Using these tools, we demonstrate the effectiveness of the proposed individual behavioural rules in improving the performance of robot swarms in the addressed tasks. The results we present in this thesis are of potential interest to both engineers designing robot swarms, and biologists investigating the behavioural rules followed by individuals in living collective organisms.

Table of contents

List of figures	xiii
List of tables	xxv
1 Introduction	1
1.1 Original contributions of the thesis	3
1.2 Publications constituting this thesis	5
1.3 Thesis outline	6
2 Literature review	9
2.1 The best-of- n decision problem	9
2.1.1 Existing individual behaviours for solving the best-of- n problem . . .	10
2.1.2 Limitations of the existing best-of- n behaviours	16
2.2 The collective resource collection task	18
2.2.1 Bio-inspiration	18
2.2.2 Implementation of stigmergic communication in robotics	19
2.2.3 The Central Place Foraging Algorithm (CPFA)	21
2.3 Conclusion	23
3 Analysis of swarm robotics systems	25
3.1 Stochastic simulations	25
3.2 Multi-agent simulations	27
3.3 Swarm robotics implementation	29
3.3.1 The Kilobot robot	30
3.3.2 The Augmented Reality for Kilobots (ARK) system	31
3.3.3 Kilobot's 3D-printed add-on	33
3.4 Swarm robotics simulations	34
3.4.1 ARGoS simulator	34
3.4.2 Minimising the reality-gap for reliable swarm robotics simulations . .	35

3.4.2.1	Modelling of noise and inter-individual variations	35
3.4.2.2	Tuning physical interactions simulation under ARGoS	37
3.4.3	Simulating ARK in ARGoS	39
3.5	Conclusion	39
4	Improving accuracy in collective decision-making	41
4.1	Formalisation of the best-of- n decision problem	42
4.1.1	Experimental setup	43
4.2	Robot behaviour	44
4.2.1	Environment exploration	44
4.2.2	Social interactions	45
4.2.3	Commitment updates	45
4.3	Direct comparison strategy	45
4.4	Collective Decisions through Cross-inhibition	46
4.4.1	The basic CDCI strategy	46
4.4.2	Stochastic analysis of the basic CDCI strategy	50
4.4.3	The time-varying CDCI strategy	52
4.5	Robot swarm simulations	54
4.6	Discussion	58
5	Achieving adaptation in collective decision-making	61
5.1	The best-of- n problem in dynamic environments	62
5.2	Environmental changes	63
5.2.1	The appearance of a new option	64
5.2.2	The disappearance of the best option	65
5.2.3	A swap of the qualities of the two best options	65
5.3	The individual robot's behaviour	66
5.4	The decision-making models	67
5.4.1	The core of the decision-making models	67
5.4.2	Local behavioural rules for achieving adaptation	70
5.4.2.1	The <i>Compare</i> rule	70
5.4.2.2	The <i>Forget</i> rule	71
5.5	Experimental setup	72
5.5.1	Multi-robot experiments	73
5.5.2	Swarm robotics experiments	74
5.6	Results	75
5.6.1	Multi-robot simulations results	76

5.6.1.1	Performance metrics	76
5.6.1.2	Effect of problem difficulty on adaptation	76
5.6.1.3	Effect of the behavioural rules' parameters on adaptation	81
5.6.1.4	Effect of the robot's communication range on adaptation	83
5.6.1.5	Effect of the robot density on adaptation	85
5.6.2	Swarm robotics simulations results	86
5.7	Discussion	87
6	A simple individual behaviour for a tunable collective resource collection behaviour	93
6.1	Resource collection in an unknown environment	94
6.1.1	The resource collection task	94
6.1.2	Getting the required robot capabilities through the ARK system	95
6.2	A simple individual behaviour for complex coordination	96
6.2.1	Adaptivity to relative quality differences	98
6.2.2	Modulation of the individual rules to obtain a plastic behaviour	99
6.3	Optimal foraging model	100
6.3.1	Estimation of model parameters from simulation data	102
6.4	Results	102
6.4.1	Results show tuneable and adaptive swarm responses	104
6.4.2	Comparison of model and simulation data	109
6.5	Discussion	112
7	Conclusion and future work	115
	References	121

List of figures

- 3.1 The visual interface of DeMaMAS. The big circles represent the options (*i.e.* the choices) while the small circles represent the agents. The colour of an agent indicates its choice and its motion state. For example, the red agents (*i.e.* the small red circles) are committed to the red-coloured option and are randomly exploring the environment. The light-red agents are committed to the red-coloured option and are on their way to measure its quality. The grey agents are uncommitted. 28
- 3.2 The general framework of collective decision-making agent-based models proposed in DeMaMAS. 29
- 3.3 A picture of the Kilobot robot 30
- 3.4 Picture of the Kilogrid system (image source [207]). 31
- 3.5 Graphical representation of the ARK architecture (image source [160]). 32
- 3.6 A picture of a Kilobot with a 3D-printed ring (originally designed for the study of [150]) which considerably improves ARK’s performance in terms of tracking and LED colour detection. 33
- 3.7 ((a)) Distribution of the bias in straight motion estimated from measurements over 120 different Kilobots which have been previously calibrated (60 manually, 60 automatically). The bars consider 12 bins in the range $[-6, 6]$ mm. The solid red line shows the approximated Gaussian distribution $\mathcal{N}_b(\mu_b, \sigma_b)$ (with mean μ_b and standard deviation σ_b) used in the default configuration of ARGoS. ((b)) Comparison between simulation (600 robots) and reality (120 robots) in term of MSD when robots are asked to go straight for 1 minute. The default noise values of ARGoS give an accurate match between reality and simulation. 37

- 3.8 ((a)) Initial distribution in four concentric circles with all 50 Kilobots facing towards the centre. ((b)) Comparison between real 50 Kilobots (19 runs) and 50 simulated Kilobots (100 runs) in ARGoS and Kilombo. We show the average mean square displacement (MSD) in a highly dense environment. ARGoS shows a good agreement with reality, whereas Kilombo does not. Video footage is available at <https://youtu.be/6HYti0ABuxc>. 38
- 3.9 Two screenshots of the same experiment in simulation (left) and reality (right). We (re)implemented the Demo C from [160] in which 50 Kilobots sense and modify two virtual environments. The full video is available at <https://youtu.be/kioZR99hnU4>. 40
- 4.1 Sample initial distribution of $S = 200$ simulated Kilobots in a scenario with $n = 6$ options and decision difficulty $\kappa = 0.5$. The red circles represent the areas (radius 25 cm) in which the options can be perceived by the robots via ARK. The colour intensity represents the option's quality $v_i \in [0, 10]$. The swarm is tasked to select the best option. 43
- 4.2 The results of 200-Kilobot swarm (100 simulations in each condition) showing the effect of noise strength $\sigma^2 \in [0, 5]$ on the decision accuracy in the best-of-6 problem with difficulty $\kappa = v_L/v_H \in [0.5, 1]$ ($v_H = 10$). We compare the accuracy of the DC strategy of Section 4.3 (bottom-left triangles) with the accuracy of the time-varying strategy $r_{step}(t)$ (with $\tau_0 = 50$) of Section 4.4.3 (top-right triangles). While DC is highly sensitive to noise, the proposed strategy shows remarkably high performance ($\geq 93\%$) for any tested noise level and difficulty κ up to 0.9. In case of equal-quality options ($\kappa = 1$), the quick dynamics of DC allows to break the symmetry within 2 hours more often than the proposed strategy with a suboptimal parameterisation of τ_0 (see more details in Figure 4.9). 47
- 4.3 The PFSM controlling the robot's decision state. Robots update their commitment state using the probabilities P_{γ_i} and P_{α_i} of Equation (4.2) or upon receiving a message M_i from a robot committed to i . The symbol $|$ on the arrow for the discovery transition indicates conditional probability on the occurrence of the event E_i of encountering option i . The transmission symbols indicate that a robot in state C_i sends an interaction message (for recruitment and cross-inhibition) with probability P_{h_i} 48
- 4.4 The FSM controlling the robot's movements in the CDCI strategy. Red dots indicate that the robot accesses its GPS location (through ARK). 49

- 4.5 Results of the SSA showing the influence of the interaction ratio $r = h/k$ with $k = 1$ (of Equation (4.2)) for various best-of- n problems in the case of $S = 200$ robots. The pie-charts indicate the percentage of 1,000 runs terminating in a decision deadlock, *i.e.* below quorum $Q = 0.8$ after $T_{max} = 10$ (yellow), a decision for the best option $v_H = 10$ (green), or a decision for any $n - 1$ inferior-quality distractor $v_L = \kappa \cdot v_H$ (red). Panel (a) Sensitivity of CDCI to the ratio r for various problem difficulties $\kappa \in [0.5, 1]$ in case of $n = 6$; the minimum r necessary to break decision deadlock grows quadratically with κ . Panel (b) Sensitivity of CDCI to the ratio r for various number of options $n \in [2, 12]$ in case of $\kappa = 0.9$; the minimum r necessary to break decision deadlock grows quadratically with n . Sufficiently high values of interaction rate r always lead to a decision, but accuracy rapidly decreases with increasing n or κ 52
- 4.6 Two forms for the time-varying interaction $h(t)$: $h_{ramp}(t)$ of Equation (4.4) in panel (a) and $h_{step}(t)$ of Equation (4.5) in panel (b). With the ramp function, the robot constantly increases the interaction strength $h_{ramp}(t)$ with a slope proportional to the estimated quality \hat{v}_i ; instead, with the step function, the robot does not interact $h_{step}(t) = 0$ until a time $\tau(\hat{v}_i)$ that is inversely proportional to the estimated quality \hat{v}_i 54
- 4.7 Results of the SSA showing the effect of the time-variant behaviours on the decision outcome of a 200-agents swarm for various best-of- n options problems $n \in \{3, 6, 9, 12\}$ with difficulty $\kappa = 0.9$. Results of each condition are computed over 1,000 runs and represented with a pie-chart that indicates the percentage of runs terminating in a decision deadlock (yellow), a correct decision (green), or an incorrect decision (red). In each run, the swarm makes a correct/incorrect decision when it reaches the quorum threshold $Q = 80\%$ within $T_{max} = 10$ for the best option ($V_H = 10$)/lower-quality option ($V_L = \kappa \cdot V_H$). Otherwise the swarm is stuck in decision deadlock. The $r_{ramp}(t)$ and $r_{step}(t)$ from Eqs. (4.4)-(4.5) with parameters $\tau_0 = 5$ and $H_{max} = 100$ show a considerable improvement in accuracy (accuracy rate reported at the centre of each pie-chart) 55

4.8	200-Kilobot swarm results (100 simulations for each condition) for different decision strategies (in each column) in case of $n \in \{3, 6\}$ options with difficulty $\kappa = 0.9$ and noise strength $\sigma^2 = 1$. Top pie-charts show the decision accuracy (same colour code of Figs. 4.5-4.7). Bottom boxplots show the decision time; the horizontal red line (at 2 hours) is the cutoff time to compute the decision outcome (e.g. indecision vs decision). We let the simulation run a maximum of 5 hours to display the complete decision time dynamics. Low interaction rate ($r = 1$) shows low convergence rate and frequent deadlocks. High interaction rate ($r = 100$) shows low accuracy. Time-varying $r_{ramp}(t)$ shows an improvement in accuracy, which is further improved by $r_{step}(t)$ (both time-varying strategies use $\tau_0 = 50$ min). The DC (of Section 4.3) shows low accuracy due to the spreading of noisy estimates.	56
4.9	Speed (green lines with 95% confidence shades and right y-axis) and accuracy (red lines and left y-axis) of the swarm robotics system for varying interaction speed $\tau_0 \in [0, 60]$ min for the $r_{step}(t)$ strategy. An inaccurate tuning of τ_0 may lead to sub-optimal performance.	57
5.1	A visual illustration of the <i>appearance</i> scenario in which a better option appears in the environment.	64
5.2	A visual illustration of the <i>disappearance</i> scenario in which the best option disappears during the experiment.	65
5.3	A visual illustration of the <i>swap</i> scenario in which the qualities of the two best options are swapped.	66
5.4	The PFSM used to implement our version of the weighted voter model [211]. This PFSM is shown in case of $n = 2$ options but can scale to any number of options with two transitions linking each pair of commitment states. Each robot executes this PFSM to update its opinion about the best option based on the information it acquires through environment exploration and interaction with peers. Upon the encountering E_i of option i (indicated by the symbol of conditional probability), an uncommitted robot U commits to the option i with probability P_{D_i} . We call this transition a <i>discovery</i> of option i . When a robot is uncommitted (state U) or committed to option j (state C_j) and receives a recruitment message $R_{i \neq j}$ from a robot that is committed to option i , the receiving robot commits to option i . At each broadcast time-step, robots that are committed to option i advertise the option i with probability P_{A_i}	69

- 5.5 The PFSM of the weighted voter decision-making model with the *Compare* rule (represented with red colour). This model extends our version of the original weighted voter model (see section 5.4.1 and figure 5.4). Following the *Compare* rule, when a robot that is committed to an option i encounters another option $j \neq i$ (*i.e.* satisfies the condition $E_{j \neq i}$), the robot switches its commitment from option i to j with probability $P_{S_{ji}}$ given by equation (5.3). 71
- 5.6 The PFSM of the weighted voter decision-making model when including the *Forget* rule (represented with red colour). This model extends our version of the original weighted voter model (see section 5.4.1 and figure 5.4). Following the *Forget* rule, a robot that is committed to an option i spontaneously forgets its option and becomes uncommitted with probability P_{F_i} given by equation (5.4). 72
- 5.7 A screenshot of the multi-agent simulations conducted in DeMaMAS (described in section 3.2) for $S = 50$ robots and $n = 3$ options. The frame shows the limits of the 1×1 environment with periodic boundary conditions. The coloured circles represent the areas (radius 0.2) in which the options can be perceived by robots (the small grey circles). 73
- 5.8 Screenshots of the ARGoS simulator showing the state of the environment (*i.e.* the number n and the qualities v_i of the options) at different times of the swarm robotics experiments. The colour-coded circles (radius $S_r = 0.2$ m) represent the area where each option is perceivable by the robots. At $t = 0$ min (panel (a)), four options are available in the environment: option 1 (red) of quality $v_1 = 0.8$, option 2 (green) of quality $v_2 = 0.6$, option 3 (blue) of quality $v_3 = 0.4$, and option 4 (magenta) of quality $v_4 = 0.2$. At $t = 20$ min, option 1 disappears, and the other options remain unchanged. At $t = 40$ min, the qualities of options 2 and 3 are swapped, *i.e.* $v_2 = 0.4$ and $v_3 = 0.6$ 74

5.9 Results of multi-agent simulation experiments described in section 5.5.1 for $S = 50$ -robots swarm in the case of the dynamic best-of-3 options problem described in section 5.2 when using the *Compare* behavioural rule introduced in section 5.4.2.1. Panels (a), (b), and (c) show the results in the *appearance*, the *disappearance*, and the *swap* scenarios (described in section 5.2.1), respectively. In each panel, we vary the difficulty $\Delta v \in \{0, 0.1, 0.2, \dots, 0.5\}$ by varying the quality of the second-best option $v_2 \in \{0.3, 0.4, \dots, 0.8\}$, and fixing the quality of the best option to $v_1 = 0.8$ and the quality of the worse option to $v_3 = 0.1$. We fixed the noise level of the robots' quality estimates to $\sigma = 0.01$ and the parameter of the *Compare* behavioural rule to $k = 0.1$. In each tested condition, the results are obtained through 100 simulation runs. The outcome of the swarm's decision is reported on the left y-axis using the blue lines and markers. The circular markers report the proportion of runs where the swarm adapted its decision to the environmental change. The square markers show the proportion of runs where the swarm kept its decision unchanged, and the triangular markers show the proportion of runs where the swarm became undecided. The adaptation time is reported on the right y-axis using the red x-shaped markers. 78

5.10 Results of multi-agent simulation experiments described in section 5.5.1 for $S = 50$ -robots swarm in the case of the dynamic best-of-3 options problem described in section 5.2 when using the *Forget* behavioural rule introduced in section 5.4.2.2. Panels (a), (b), and (c) show the results in the *appearance*, the *disappearance*, and the the *swap* scenarios (described in section 5.2.1), respectively. In each panel, we vary the difficulty $\Delta v \in \{0, 0.1, \dots, 0.5\}$ by varying the quality of the second-best option $v_2 \in \{0.3, 0.4, \dots, 0.8\}$, and fixing the quality of the best option to $v_1 = 0.8$ and the quality of the worse option to $v_3 = 0.1$. We fixed the noise level of the robots quality estimates to $\sigma = 0.01$ and the parameter of the *Forget* behavioural rule to $\alpha = 0.01$. In each tested condition, the results are obtained through 100 simulation runs. The outcome of the swarm's decision is reported on the left y-axis using the blue lines and markers. The circular markers report the proportion of runs where the swarm adapted its decision to the environmental change. The square markers show the proportion of runs where the swarm kept its decision unchanged, and the triangular markers show the proportion of runs where the swarm became undecided. The adaptation time is reported on the right y-axis using the red x-shaped markers. 79

- 5.11 Effect of the parameter k of equation (5.3) on adaptation when using the *Compare* rule. 82
- 5.12 Effect of the parameter α of equation (5.4) on the adaptation when using the *Forget* rule. 83
- 5.13 Effect of the robot's communication range on adaptation in the *appearance* scenario described in section 5.2.1. Two options are initially available in the environment (option 2 of quality $v_2 = 0.7$ and option 3 of quality $v_3 = 0.1$). Option 1 of quality $v_1 = 0.8$ appears at time step $T_c = 10,000$. The swarm size is fixed to $S = 200$, and the quality estimation noise is set to $\sigma = 0.01$. The robot's sensing range is fixed to $S_r = 0.2$ and the robot's communication range C_r is varied in $\{0.025, 0.05, 0.075, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5\}$. Panel (a) shows the results when using the *Compare* rule with $k = 0.1$. Panel (b) shows the results when using the *Forget* rule with $\alpha = 0.01$. The results of each tested condition are obtained through 100 simulation runs. In each tested condition, the proportion of runs where the swarm adapted its decision to the new option is reported using the circular markers while the proportion of runs where the swarm kept the same decision is reported using the square markers. 84
- 5.14 Effect of the robot density on adaptation in the *appearance* scenario described in section 5.2.1. Two options are initially available in the environment (option 2 of quality $v_2 = 0.7$ and option 3 of quality $v_3 = 0.1$). Option 1 of quality $v_1 = 0.8$ appears at time step $T_c = 10,000$. The robot density D is varied in $\{50, 100, 200, 300, 400, 500, 1000\}$. The robot's sensing range is fixed to $S_r = 0.2$, the robot's communication range is set to $C_r = 0.1$, and the quality estimation noise is set to $\sigma = 0.01$. Panel (a) shows the results when using the *Compare* rule with $k = 0.1$. Panel (b) shows the results when using the *Forget* rule with $\alpha = 0.01$. The results of each tested condition are obtained through 100 simulation runs. In each tested condition, the proportion of runs where the swarm adapted its decision to the new option is reported using the circular markers while the proportion of runs where the swarm kept the same decision is reported using the square markers. 86

-
- 5.15 Time evolution of the fraction of robots committed to each option in simulated swarm robotics experiments described in section 5.5.2. The coloured lines show the average fraction of robots over 100 simulation runs. The coloured shade around each line shows the 95% confidence interval. The colour of each line and shade matches the colour of the corresponding option. The vertical black dashed lines show the times at which environmental changes occur and mark the different part of the experiment. The available options in each part of the experiment, their qualities, and their colours are shown on the top of each plot. The horizontal black dashed line indicates the quorum threshold $Q = 80\%$. Panel (a) shows the results when each robot executes the *Compare*-based decision-making model of section 5.4.2.1. Panel (b) shows the results when each robot executes the *Forget*-based decision-making model of section 5.4.2.2. 88
- 5.16 Screenshots of the swarm robotics experiment described in section 5.5.2 conducted under ARGoS simulator. The screenshots are taken at different times throughout the experiment. Here, each robot executes the *Compare*-based decision-making model introduced in section 5.4.2.1. A video of the experiment is available at <https://youtu.be/nOU8XCe0J5Y>. 89
- 5.17 Screenshots of the swarm robotics experiment described in section 5.5.2 conducted under ARGoS simulator. The screenshots are taken at different times throughout the experiment. Here, each robot executes the *Forget*-based decision-making model introduced in section 5.4.2.2. A video of the experiment is available at <https://youtu.be/zudCnoPRG3c>. 90
- 6.1 Kilobots sense via ARK the presence of virtual pheromone in front of themselves at a distance of ~ 3.5 cm in four 45° -wide sectors. The virtual sensor indicates the presence or absence of pheromone as binary values, therefore, the Kilobot has no information about the pheromone quantity or concentration difference. In this illustration, pheromone is represented as blue circles, and thus the virtual sensor readings are [1,0,1,0]. When an exploring Kilobot detects pheromone, it interrupts random exploration and moves towards the detected pheromone. If more than one sector has pheromone (as in the illustration), to decide its motion direction the robot compares the sectors' direction with the depot direction (depot illustrated as a house and direction differences as red and green angles) and moves towards the largest angle (green arrow). 96

-
- 6.2 Probabilistic Finite State Machine (PFSM) of the individual robot behaviour. Circles represent states and arrows are transitions. Robots start exploring the environment through a random walk (RW); when they find a source area, they collect an item and return to the depot (GD) laying pheromone according to Equation (6.2). Once arrived at the depot, they either turn back (TB) or resume exploration (RW). When explorer robots detect pheromone, they follow it (FP). When robots detect a wall, they avoid it (AO). Controlling individuals through this simple PFSM leads to sophisticated collective foraging dynamics. 98
- 6.3 Fits of Equation (6.5) to data generated by physics-based simulations in order to obtain the model parameters reported in Table 6.1. Fitting is performed in the case of $n = 2$ source areas with different quality and equal distance in panel (a), equal quality and different distance in panel (b), and equal quality and distance in panel (c). Data points are represented using symbols and fits are represented using lines (circles and solid grey lines show collection from source A_1 while triangles and dash-dotted blue lines show collection from source A_2). Error bars represent 95% confidence intervals. There is a linear growth for small-to-medium numbers of robots on a path, and a nonlinear decay for medium-to-large numbers of robots on a path. This type of growth-decay curve on population size is widespread in nature [104] as in engineering [70]. 103
- 6.4 (Colours online) A picture of a 50 real Kilobots experiment with the virtual environment superimposed on the image. The red (bottom-left) source area A_1 has quality $Q_1 = 10$, while the yellow (top-right) source area A_2 has quality $Q_2 = 4$. The sources are placed at $d_1 = 1$ m and $d_2 = 0.6$ m from the central (blue) depot. The (light-blue) shades represent the pheromone trails that the robots deposit and follow. Full videos are available at <https://www.youtube.com/playlist?list=PLCGKY9OHLZwMaGeB6cxVfxmHwhBFqKF7a>. 104

- 6.5 A 50 simulated Kilobot swarm experiment inspired by the ants' double-bridge experiment by [66] in which two paths, a longer path (1.8 m long) and a shorter path (1.4 m long), connected source to depot. When the simulated swarm had access to only the longer path, panel (a), the Kilobots reinforced pheromone on that path and used it for their collections. Instead, when both paths were available, panel (b), the Kilobots disregarded the longer path and (almost exclusively) used the shorter for their collections. Panel (c) shows the number of robots on the two paths at the end of one simulated hour (boxes range from 1st to 3rd quartile of the data from 100 simulations and indicate the median with a horizontal line, the whiskers extends to 1.5 IQR). The individual Kilobots cannot follow a pheromone gradient nor detect any difference in pheromone concentration. Despite their limited individual capabilities, the robot swarm shows (in certain experimental conditions) behaviour similar to ants' colonies, which instead rely on much higher cognitive abilities at the individual level. 105
- 6.6 Simulation results showing the adaptivity of the system. We measured the number of collected items in panel (a) and the number of robots on each path in panel (b) for the two source areas, the superior A_1 and inferior A_2 , both at equal distance $d_1 = d_2 = 1$ m. We kept the same quality ratio, *i.e.* $Q_2/Q_1 = 0.4$, but varied the absolute value of the objects (indicated on the x-axis). All experiments were conducted with swarms of $S = 50$ Kilobots and an intermediate value of $\alpha = 0.85$ in Equation (6.2) and Equation (6.3). Boxes range from 1st to 3rd quartile of the data from 100 simulations and indicate the median with a horizontal line; the whiskers extend to 1.5 IQR. Having a constant range (dark boxplots) shows good results only if the predefined range matches the actual range of the environment (central experiment). Instead, an adaptive strategy allows the swarm to exploit resources as a function of their relative qualities in a range adapted to the environment. 107

- 6.7 Effect of the modulation of the parameter α from Eqs. (6.2) and (6.3) to favour nearer source areas ($\alpha = 0$), to favour the best-quality sources ($\alpha = 10$), or to balance the distance-quality trade-off ($0 < \alpha < 10$). Results of $\alpha = 0$ are shown in light-grey, $\alpha = 0.85$ in dark-grey, and $\alpha = 10$ in black. We report the results for simulations and physical robots experiments of one hour each in scenarios with $n = 2$ sources. We excluded the initial exploration phase and indicate mean values for the last 30 minutes. Physical robots results are indicated as solid symbols with vertical bars indicating the 95% confidence intervals of 3 runs for each condition (the symbols are slightly shifted to avoid bar overlaps but all represent results for $d_2 = 0.6$ m). Lines represent the mean of 100 simulations (shaded areas are 95% confidence intervals). Source A_1 had quality $Q_1 = 10$ and was located at distance $d_1 = 1$ m; source A_2 had quality $Q_2 = 4$ and varying distance $d_2 \in [0.5, 1.0]$ m. We report the rate of collected items per minute in panel (a), the mean number of robots on each path in panel (b), and the rate per minute of collected items weighted by the normalised quality $q_1 = 1.0$ and $q_2 = 0.5$ in panel (c). Individual robots can locally modulate the decentralised parameter α to lead the swarm to a range of different collective responses, *e.g.* selecting almost exclusively the best-quality source (high α) or balancing the distance-quality trade-off (low α). Physical robots are less efficient than simulations, however ordering between sources is preserved; this confirms the effects of α -modulation observed in simulation. 108
- 6.8 (Colours online) Comparison of model with simulations and experiments: Total yield R as a function of the normalised swarm allocation ρ/ρ_w and the number of worker robots $\rho_w S$. We report the predicted yield R from the model of Equation (6.4) as a colour heatmap and we overlay robot simulations for three strategies: distance-selective $\alpha = 0$ (circles), distance-quality trade-off $\alpha = 0.85$ (diamonds), and quality-selective $\alpha = 10$ (triangles). We report simulations for swarm sizes $S = 50$ (cyan), $S = 100$ (green), $S = 200$ (purple) and $S = 500$ (white). Under the model's assumptions, the simulated robot swarm performs best for $S = 200$ and $\alpha = 0.85$ ($R = 150.6 \text{ m}^{-2}$) in (a), $\alpha = 10$ ($R = 177.1 \text{ m}^{-2}$) in (b) and $\alpha = 10$ ($R = 120.4 \text{ m}^{-2}$) in (c). Swarms of large size ($S = 500$) do not achieve good performance as they equally exploit both sources and do not avoid overcrowding. The star symbol in (c) was obtained from three experiments with 200 Kilobots assuming $\alpha = 0.85$ (see online videos). Error bars represent 95% confidence intervals. Parameters: β_j , $T_{0,j}$ and κ_j are given in Table 6.1. 110

- 6.9 (Colours online) Total yield R as a function of the normalised swarm allocation ρ/ρ_w and the number of worker robots $\rho_w S$ in the collision-free condition. We removed the effect of physical interactions (*i.e.* collisions between robots) that may cause traffic congestions and we report the predicted yield R from the model (6.4) as a colour heatmap and we overlay robot simulations for three strategies: distance-selective $\alpha = 0$ (circle), distance-quality trade-off $\alpha = 0.85$ (diamond), and quality-selective $\alpha = 10$ (triangle). We report simulations for swarm sizes $S = 50$ (cyan), $S = 100$ (green), $S = 200$ (purple) and $S = 500$ (white). Without collision, the predicted best strategy is allocation of all workers to the best-quality or closest source area. The collision-free simulations approximate such result when the corresponding strategy is activated, *e.g.* quality-selective $\alpha = 10$ (triangle) in panel (a) and the distance-selective $\alpha = 0$ (circle) in panel (b). Error bars represent 95% confidence intervals. Parameters: β_j , $T_{0,j}$ and κ_j are given in Table 6.1. 111

List of tables

6.1 Overview of estimated model parameters. The goodness-of-fit is quantified by $R_{\text{GoF},j}^2 = 1 - \sum_i (y_i - y_i^{\text{fit}})^2 / \sum_i (y_i - \bar{y})^2$, where $y_i = \Delta U_i / \text{min}$, the y_i^{fit} correspond to the fitted values, and \bar{y} represents the mean value of all y_i . The index j corresponds to the trail. Mean model parameter values, including one standard deviation errors (values in brackets), are given. 102

Chapter 1

Introduction

Swarm robotics [76] is the subfield of robotics where systems are composed of a large number of relatively simple robots. The robots are simple compared to the task they are required to solve, *i.e.* the robots would be unable to solve the task individually [76]. These robots locally interact with each other and with the environment via simple behavioural rules. These local interactions allow the swarm as a whole to achieve complex tasks without any central control. To achieve local interactions, the individual robots are generally equipped with a limited sensing and communication range [76]. The behaviour of robot swarms is often inspired by the functioning of social insect colonies, such as ants and bees, where the individuals perform simple actions that when combined enable the colony to accomplish tasks that are not achievable by the individual. For instance, army ant colonies are able to build bridges to cross gaps, thanks to two simple rules followed by each ant: stop moving when another ant is walking over its back and resume moving if no other ant is walking over its back [59, 67].

In the last decade, mobile robots became more capable and reliable due to the recent developments in environment sensing technologies such as 3D cameras [63], and the huge advances made in computer science areas including artificial intelligence [124] and 3D machine vision [224, 225]. Therefore, a fair question concerning swarm robotics emerged: Why use a swarm of simple robots instead of a single highly-capable robot? The answer is that, similarly to their biological counterpart (*i.e.* social insects), robot swarms are expected to offer interesting advantages that are not always provided by single complex robots. These advantages include fault tolerance, scalability, adaptivity, and parallelism [16, 85]. *Fault tolerance* [26] is the ability of robot swarms to accomplish their assigned task even under the failure of a reasonable number of robots. The fault tolerance feature is the result of the enormous redundancy achieved by using large numbers of homogeneous robots and the lack of central control that represents a single-point-of-failure. *Scalability* [37, 77] is the capacity of robot swarms to maintain their performance in the event of the addition or the removal of robots without the need to redesign

their controllers. *Adaptivity* [47] is that robot swarms are expected to adapt their behaviour to environmental changes and to successfully perform their task without prior knowledge of the environmental conditions. *Parallelism* [141] is the ability of robot swarms to simultaneously exploit independent pieces of information and perform different actions. This advantage is a direct consequence of using a large number of robots that can occupy different areas of the environment at the same time.

The above advantages make robot swarms suitable for a variety of applications. This includes tasks performed under uncertain conditions such as in disaster areas [62, 6, 22] and space exploration [203, 220], and tasks executed in environments where faults are highly likely and maintenance is not possible such as in hazardous environments [32], underwater operations [95], and space missions [203]. Robot swarms are also desired in scenarios where the swarm size is likely to vary, such as in agricultural applications [3] where the number of robots may be selected depending on the field's size. Moreover, robot swarms are appropriate for applications that benefit from parallelisation, such as construction works [186] where multiple structures can be constructed in parallel to accelerate the process.

Although the numerous advantages offered by swarm robotics and its broad range of possible applications, this technology is still in the research phase and not ready for deployment in the real-world [76]. In the last two decades, researchers focused their efforts in laying the foundation for the swarm robotics field by trying to reproduce the collective behaviours observed in biological swarms such as social insects colonies [105], bird flocks [14], and fish schools [120]. The collective behaviours reproduced by researchers are very simple and constitute a sort of building blocks for the complex behaviours needed in real-world applications. The reproduced behaviours enables robot swarms to achieve spatial organisation such as pattern formation [190, 184] and aggregation [202, 58, 188, 189], efficient navigation such as collective transport [69, 45, 175] and coordinated motion [205, 46], and decision-making abilities such as task allocation [1, 140, 223] and consensus achievement [167, 211, 130].

To allow robot swarms to achieve the desired collective behaviours, researchers are required to determine the behavioural rules that each robot must follow for the swarm to manifest the desired collective behaviour (also known as the *macroscopic behaviour*) [16, 8]. These behavioural rules define how the individual robot interacts with other robots within its local communication range and with environmental elements within its local sensing range. The set of behavioural rules that the individual robot executes constitute what is referred to as the *individual behaviour* [169], or as the *microscopic behaviour* [167, 165], or sometimes as the *agent-based model* [82].

The work presented in this thesis aims at improving the performance of robot swarms in the best-of- n decision and collective resource collection tasks through the extension of

existing individual behaviours, via the addition of simple but effective behavioural rules. We introduce the original contributions made in this thesis and their corresponding publications in Sections 1.1 and 1.2, respectively. We outline the structure of the thesis in Section 1.3.

1.1 Original contributions of the thesis

This work addresses two well-known collective behaviours. The first behaviour is the so-called *best-of- n decision problem* where robot swarms are required to reach a consensus for the best option among n available alternatives [134, 71, 167, 177, 210, 206, 209, 198]. We looked at the best-of- n decision problem because solving this problem is considered as a fundamental cognitive skill for robot swarms due to its high usability within various collective tasks [209]. For instance, to succeed in its landing, a swarm of aerial robots is required to find the most level landing spot. Similarly, bridge-building robots have to select the location with the smallest gap to ensure the building of a solid bridge.

Previous research works introduced various individual behaviours to allow robot swarms to solve the best-of- n decision problem [218, 134, 122, 123, 177, 211, 212, 210, 165, 167, 159, 209]. Most of these behaviours addressed the best-of- n problem in the case of $n = 2$ options. However, a recent theoretical analysis [164] has shown that the dynamics of the decision process are qualitatively different for a higher number of options (*i.e.* $n > 2$) especially when the difference between their qualities is small. The ability of the swarm to reach a consensus for the best option decreases as the number of options and the similarity between their qualities increases. Moreover, this theoretical study [164] demonstrated the existence of a dilemma in these scenarios. If the robots rely mainly on individually acquired information to update their individual belief about the possible best option, the swarm will be unable to reach a consensus. However, when the robots update their individual belief mainly based on information acquired through interaction with others, the swarm accuracy in selecting the best option decreases. In the work presented in this thesis (see Chapter 4), we validate the presence of such a dilemma through stochastic analysis and introduce individual behavioural rules whereby robots control their interaction with others through time to overcome the presented dilemma and achieve better performance.

Almost all existing studies considered static best-of- n decision problems where the number of the available options and their qualities remains constant throughout the decision process [218, 134, 122, 123, 177, 211, 212, 210, 165, 167, 159, 209]. This assumption is suitable to model scenarios where solving the best-of- n problem is followed by the execution of other tasks for which the success depends on the swarm's decision. Therefore, the above works aimed at maximising the swarm's decision accuracy and minimising the time to make a decision. This

is known in the literature as optimising the speed-accuracy trade-off [142, 212, 210]. However, only a little attention has been given to dynamic best-of- n decision problems where the number of options and their qualities may vary during the decision process [149, 148]. In dynamic best-of- n decision problems, the swarm is not only required to select the best available option but also to keep its decision up-to-date with the environmental changes. In the work presented here (see chapter 5), we introduce two individual behavioural rules, each of which, when added to the existing individual behaviours for solving the best-of- n decision problem, enables the swarm to adapt its decision to environmental changes.

The second collective behaviour addressed in this thesis is the collective resource collection task where robots are instructed to collect objects spread in an unknown environment [83, 84, 82, 85, 143, 53, 111, 112, 144, 2, 110]. We address this task for its substantial application potential in various real-world domains including space exploration, search and rescue, and the collection of natural resources [220, 16, 221]. In the real world, resources are generally distributed in the form of clusters [168]. Previous research demonstrated that relying on memory and communication is necessary for the effective collection of clustered objects [85]. Memory and communication can be implemented using *stigmergy* [76] where the robots mark the environment as a way to memorise and share information. Through stigmergy the robots use the environment as a shared memory to store interesting information without the need for individual memory [8]. Moreover, stigmergy allows the robots to exchange information without being present in the same place at the same time [89]. For this reason, researchers in swarm robotics dedicated considerable attention to the use of stigmergy in the collective resource collection task [65, 217, 136, 129, 19, 39, 90, 153, 60, 83, 82, 85, 111, 112]. Most of these works considered that the available objects have the same value, and hence only focused on finding individual behaviours that allow robot swarms to minimise the time to collect the objects available in the environments [83, 82, 111, 112]. However, far too little attention has been paid to the case where the objects have different values. This scenario represents a class of potential real-world applications where the objects to collect may have different importance or priority. For instance, in a human-performed search and rescue operation, victims are rescued based on their risk level, starting from those at higher risk then moving to those at lower risk [38]. Value-based resource collection has been also observed in some species of foraging ants that assign different values to the different food types available in their environment [9, 147, 182]. In this thesis (see Chapter 6), we introduce a simple ant-inspired stigmergy-based individual behaviour that enables robot swarms to collect objects of different quality. Using the proposed behaviour, the swarm is able to satisfy various collection objectives observed in real ants, such as focusing the collection efforts on the best-quality resource [10, 9, 158, 182], the nearest resource [66, 33], or balancing the distance-quality trade-off [127]. Satisfying multiple objectives makes the

proposed behaviour suitable for various real-world applications. For instance, focusing the collection on high-quality objects is required in applications where the quality is essential while focusing the collection on nearest sources is appropriate for applications where quantity is important.

To take a step closer to the application of swarm robotics in the real world, a growing number of research works are validated using real robots. However, most research studies in swarm robotics are still conducted in simulation [16, 51] because experimenting with a large number of robots is very challenging and time-consuming. Moreover, simulation allows researchers to multiply trial-and-error cycles to obtain satisfactory results and to conduct experiments that are not achievable within the space of a typical research lab such as using very large swarms and environments. However, for simulation to not be an impeding factor for the advancement of swarm robotics towards applicability, simulation must allow researchers to correctly predict the performance of their designed collective behaviours in the real world. Therefore, swarm robotics simulators should be as accurate as possible in representing real-world conditions. In this thesis, we validated the results of our works through both physics-based simulations and real robots experiments. We employed the well-known Kilobot platform [174]. As part of our efforts to obtain accurate simulation results, we contributed to the development of a Kilobot simulator which minimises the reality-gap (see Section 3.4.2).

1.2 Publications constituting this thesis

The contributions listed in Section 1.1 have been disseminated through the following research publications:

Journal articles:

- **Talamali, M. S.**, Saha, A., Marshall, J. A. R., and Reina, A. (2021). When less is more: robot swarms can better adapt to changes with constrained communication. (submitted)
- **Talamali, M. S.**, Bose, T., Haire, M., Xu, X., Marshall, J. A. R., and Reina, A. (2020). Sophisticated collective foraging with minimalist agents: a swarm robotics test. *Swarm Intelligence*, 14(1):25–56.

Conference papers:

- **Talamali, M. S.**, Marshall, J. A. R., Bose, T., and Reina, A. (2019). Improving collective decision accuracy via time-varying cross-inhibition. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9652–9659.

- Font Llenas, A., **Talamali, M. S.**, Xu, X., Marshall, J. A. R., and Reina, A. (2018). Quality-sensitive foraging by a robot swarm through virtual pheromone trails. In *Swarm Intelligence (ANTS 2018)*, volume 11172 of LNCS, pages 135–149. Springer.
- Pinciroli, C., **Talamali, M. S.**, Reina, A., Marshall, J. A. R., and Trianni, V. (2018). Simulating Kilobots within ARGoS: models and experimental validation. In *Swarm Intelligence (ANTS 2018)*, volume 11172 of LNCS, pages 176–187. Springer.

1.3 Thesis outline

This thesis is structured into seven chapters.

In Chapter 2, we review the literature related to the collective behaviours addressed in this thesis. In Section 2.1, we look at previous works that introduced individual behaviours for solving the best-of- n decision problem and highlight their limitations. We dedicate Section 2.2 to overview literature related to stigmergy-based collective resource collection in swarm robotics. We review works about foraging ants which are the main source of inspiration for stigmergy-based resource collection. Then we introduce the techniques used by engineers to implement stigmergic communication in swarm robotics. Finally, we present and review the development of the *Central Place Foraging Algorithm (CPFA)* a state-of-the-art collective resource collection algorithm.

In Chapter 3, we introduce the tools we employed to analyse the swarm systems studied in this thesis. In Section 3.1, we introduce the Gillespie algorithm [64] we used to analyse chemical reaction models of our swarm systems. In Sections 3.1 and 3.4, we introduce the multi-agent simulator and the physics-based swarm robotics simulator respectively, that we employed to test our proposed individual behaviours. In Section 3.3, we present the Kilobot platform [174] we employed and the Augmented Reality for Kilobots (ARK) system [160] we used to enhance its capabilities. Part of the content covered in this chapter has been published in [138].

In Chapter 4, we introduce individual behavioural rules for improving the performance of robot swarms in the best-of- n decision problem. We formalise the best-of- n decision problem and describe our experimental setup in Section 4.1. In Section 4.2, we present the overall structure of the individual behaviours each robot follows to solve the best-of- n decision problem. In Section 4.3, we introduce the Direct Comparison (DC) strategy, a simple individual decision-making algorithm that we compare our proposed individual behaviour against. In Section 4.4, we present the honeybee-inspired individual decision-making behaviour, analyse its performance using the Gillespie algorithm [64], and introduce behavioural rules for enhancing its performance. In Section 4.5, we confirm the benefit of our proposed behavioural rules

through physics-based swarm robotics simulations and compare their performance to both the original honeybee-inspired behaviour and the DC algorithm. The work presented in this chapter has been published in [198].

In Chapter 5, we introduce individual behavioural rules that allow robot swarms to adapt their decision in the case of dynamic best-of- n decision problems where the best option changes over time. In Section 5.1, we formalise dynamic best-of- n problems, while in Section 5.2, we present the types of environmental change we consider. In Section 5.3, we describe the overall individual behaviour the robots follow to solve the best-of- n problem and adapt to environmental changes. In Section 5.4, we introduce the individual decision-making behaviour and the individual behavioural rules we proposed to add to it for achieving adaptation. In Section 5.5, we define the experiments we conduct to assess the effect of the proposed behavioural rules. We present and discuss the results of these experiments in Sections 5.6 and 5.7, respectively. The work presented in this chapter is in preparation for submission to a scientific journal.

In Chapter 6, we propose a stigmergy-based individual behaviour that allows robot swarms to accomplish collective resource collection where objects have an assigned quality and to satisfy different collection objectives. In Section 6.1, we define the collective resource collection task, indicate the required robot capabilities in the case of stigmergy-based solutions, and explain how we used augmented reality to achieve these capabilities. In Section 6.1, we introduce the stigmergy-based individual behaviour we propose for solving the considered resource collection task. In Section 6.3, we present the optimality model we used to assess the performance of our proposed behaviour. The results of the analysis are shown and discussed in Section 6.4 and 6.5, respectively.

Finally, in Chapter 7, we summarise the contributions of the thesis and discuss future research work.

Chapter 2

Literature review

In previous research studies, researchers proposed individual behaviours that allow robot swarms to achieve a wide range of collective behaviours [16]. These collective behaviours include spatially-organising behaviours such as aggregation [202, 58, 188, 189], pattern formation [190, 184], and chain formation [128, 129, 191, 40]; navigation behaviours such as coordinated motion [205, 46] and collective transport [69, 45, 175]; and collective decision-making behaviours such as consensus achievement [167, 209, 130] and task allocation [1, 140, 223]. In this thesis, we propose simple individual behavioural rules that improve the performance of robot swarms in the tasks of collective decision-making and collective resource collection. In this chapter¹, we review previous research works that introduced individual behaviours to allow robot swarms to solve these collective tasks. Research works that proposed individual behaviours for collective decision-making are reviewed in section 2.1. In section 2.2, we review research works that investigated individual behaviours for collective resource collection.

2.1 The best-of- n decision problem

Making decisions is a fundamental cognitive skill for all living and artificial systems. Robot swarms are expected to autonomously solve complex tasks [76] and thus are required to make *collective* decisions in which the robots need to reach consensus for one option among numerous possible alternatives. Selecting the same option enables the swarm to respond to external stimuli in a unified and coordinated manner. For example, when the swarm needs to allocate all its resources (e.g. drilling robots) to a single task (e.g. drilling a water well) which is localised in space (e.g. drilling location), the swarm has first to decide which is the

¹In this chapter, parts of section 2.2 are a verbatim reproduce of parts of the "Related works" section of the published manuscript: Talamali, M.S., Bose, T., Haire, M. et al. Sophisticated collective foraging with minimalist agents: a swarm robotics test. *Swarm Intelligence* 14, 25–56 (2020).

best location to perform the task (e.g. the location with a higher likelihood of hosting water underneath), among the candidate spots, to deploy its workforce. Reaching a consensus would lead to the effective usage of the swarm's abilities while dividing the swarm's resources would weaken its response and may impede success. In some applications, the available choices are of equal value to the swarm; hence, the robots are required to reach a consensus for any of them [57, 78, 130, 157]. This type of decision problem is known as the *symmetry-breaking* problem [78, 209, 163]. In other applications, the available choices have different values, and hence the swarm is required to select the best option among them. This problem is known in the literature as the best-of- n decision problem [134, 71, 167, 177, 210, 206, 209, 198], where n is the number of the available alternatives. In swarm robotics, the best-of- n decision problem received considerable scholarly attention for its general formalisation of consensus problems. For instance, in the best-of- n problem formalisation, setting the qualities of the options to the same value allows to study the *symmetry-breaking* problem [159].

Previous research works proposed numerous individual behaviours that allow robot swarms to solve the best-of- n decision problem [209]. Some of these behaviours are task-specific as they enable the swarm to perform a particular consensus-based task (e.g. navigating through the shortest path [136, 71, 165], or aggregating on the spot with the highest light intensity [100]). Other research works proposed task-independent behaviours (also called opinion-based behaviours [209]). Through these behaviours, robot swarms address the best-of- n decision problem as a separate task itself. The swarm then solves its primary task based on the agreed choice. For instance, if the swarm is required to aggregate in the spot with the highest light intensity, the swarm will firstly decide about which of the available spots has the highest light intensity, then after reaching consensus, the swarm starts the aggregation process.

In this thesis, we address the best-of- n decision problem as a fundamental cognitive task that robot swarms may be required to solve as part of any other task. Therefore, we only review previous individual behaviours that allow robot swarm to solve the best-of- n decision problem independently of the context (Section 2.1.1). Some of these behaviours were originally introduced in a specific experimental scenario. Here, we introduce these behaviours in the general form of the best-of- n decision problem. In Section 2.1.2, we highlight the limitations of the reviewed behaviours.

2.1.1 Existing individual behaviours for solving the best-of- n problem

The first task-independent behaviour that aimed to allow robot swarms to solve the best-of- n decision problem has been proposed by [Wessnitzer and Melhuish \[218\]](#). In this behaviour, each robot shares its decision about the best option with other robots within its local communication range. Each robot then uses the received opinions of its peers to update its decision based on the

majority rule. When applying the majority rule, if the opinions of the neighbouring robots are equally split between the different opinions, the robot selects an opinion at random. The authors applied this behaviour in a prey-hunting scenario to allow a robot swarm to reach consensus on which prey to hunt. In their experiments, the authors focused on the binary scenario (*i.e.* two prey). The authors ignored the exploration aspect, and the robots were initially committed to one of the two prey at random. Through this behaviour, the robots' reached consensus on one of the two prey. The authors analysed the performance of the swarm against the robot's communication range and the connectivity degree (number of peers the robot considers when applying the majority rule). Their results revealed that the higher the communication range and the connectivity degree, the faster and more accurate the swarm's decision.

Parker and Zhang [134] introduced another individual behaviour for solving the best-of- n decision problem in swarm robotics. This behaviour is inspired by the nest-site selection behaviour of social insects [114, 151, 200]. The behaviour proposed by Parker and Zhang [134] is one of the most complete best-of- n solving behaviours in the literature as it allows the robots to both decide about the best available alternative and to detect that a consensus has been reached. Detecting consensus enables the robots to start implementing the outcome of the decision process. The proposed individual behaviour can be divided into two concurrent sub-behaviours: the *decision-making* behaviour that allows the robots to reach consensus on the best available option, and the *quorum-sensing* behaviour that enables the swarm to detect that consensus has been reached. In the decision-making sub-behaviour, each robot can be in one of four states. The first is the *searching* state where the robot has no opinion and explores the environment to find the available options. When a robot encounters an option, it estimates the option's quality, commits to it and enters the *advocating* state. In the advocating state, the robot advertises its option by sending recruitment messages to other robots within its local communication range. The frequency at which the robot sends recruitment messages is proportional to the quality of its option. The robot can also be in the *idle* state where it keeps waiting for recruitment messages to act accordingly. When a robot is in the searching, the advocating, or the idle states and receives a recruitment message for an option, it enters the *researching* state where it visits the option's location to estimate its quality and starts advertising it (*i.e.* returning to the advocating state). Since each robot advertises its option with a frequency that is proportional to the estimated option's quality, robots are more likely to be recruited for options with better quality. This drives the swarm's decision to converge to the highest-quality option. In the *quorum-sensing* sub-behaviour, robots estimate how much the other robots agree with their opinion. Each robot sends messages to its peers to ask them whether they agree with its opinion or not, the other robots then reply by a "Yes" or a "No" message. By computing the proportion of "Yes" replies, the robot measures the degree of agreement in the population.

When the degree of agreement reaches a certain threshold (called the quorum threshold), the robot detects that consensus has been reached and enters the *committed* state where it sends special messages to directly convert robots with a different opinion to its option. [Parker and Zhang](#) tested the proposed behaviour on swarms of up to 11 real robots in the case of $n = 2$ options. In the conducted tests, the proposed behaviour allowed the swarm to find the true best option 80% of the time despite the noisy nature of the individual quality estimates made by the robots. The accuracy of the swarm's decision increased by increasing the value of the quorum threshold. It is important to note that applying the quorum-sensing behaviour proposed by [Parker and Zhang](#) [134] requires the robot to store the reply messages and the IDs of the robots within its communication range, hence the individual robot must have enough memory capabilities. This limits the usability of the behaviour in simpler platforms such as micro and nanorobots.

[Montes de Oca et al.](#) [122] modified a theoretical majority-based opinion formation model proposed by [Krapivsky and Redner](#) [103] to build a new individual behaviour for solving the best-of- n decision problem. In the original version of the behaviour by [Krapivsky and Redner](#), in each time step, a team of three randomly selected robots is formed. Then the three robots switch their opinion to the opinion adopted by the majority within the team. This process is continuously repeated until consensus is reached. This behaviour allows the swarm to reach a consensus for the best option only when the robots opinions are initially biased towards the best option [122]. This result is undesired because, in reality, robots have no prior knowledge about which option is the best. [Montes de Oca et al.](#) [122] extended this behaviour to enable the swarm to select the best option even when the initial opinions of the robots are unbiased (*i.e.* robots opinion are randomly distributed over the opinions). In [Montes de Oca et al.](#)'s extension, multiple trios are formed at the same time to accelerate the decision process. Moreover, robots that participate in team-based voting become latent. The robots remain latent for a stochastic time that is inversely proportional to the quality of their options. The lower the quality of the robot's option; the longer the robot remains latent. Through the added latency mechanism, robots committed to better options are less frequently latent and thus are more often selected in team formation. This drives better opinions to spread faster and lead the swarm to reach consensus for the best available option. [Montes de Oca et al.](#) tested their proposed behaviour through a simulated collective transport task where robots are required to transport objects from one point to another through the shortest path. [Montes de Oca et al.](#) selected the well-know double bridge setup [66] where two paths are available, one long, and one short. Each formed team was responsible for transporting one object. The latency time spontaneously arose from the fact that teams selecting the longer path (*i.e.* the less favourable option) took more time to arrive (*i.e.* remained unusable for longer times). Through these experiments, [Montes de](#)

Oca et al. have shown that with an optimal number of simultaneous trios formation, the swarm selects the shortest path (*i.e.* the best option) with 86% accuracy. Montes de Oca et al. tested their proposed behaviour in an easy best-of- n problem (*i.e.* a large difference between the options' qualities). Therefore, it is not clear whether their proposed behaviour allows the swarm to reach consensus for more difficult problems (*i.e.* a small difference between the option's qualities) and to break the symmetry when the options have equal qualities.

In a later work [123], Montes de Oca et al. extended the behaviour they proposed earlier in [122] by replacing the majority rule with a more complex opinion update rule. This update rule is based on the exponential smoothing technique, which is generally used for smoothing time-dependent data by weighting each data point with a weight that is exponentially decreasing with time. Using this update rule, each robot continuously holds a belief about each of the n available options. This belief reflects how much the robot thinks that each of the options is possibly the best. If the robot's belief about an option exceeds a threshold, the robot updates its opinion about the best option to the corresponding option. Each robot shares its opinion about which option is the best with agents within its communication range. Then in each time step, the robot uses the exponential smoothing equation to update its belief about the available options based on the opinion of a randomly selected peer. Similarly to the behaviour proposed in [122], when an agent updates its opinion to a different option, the agent stops updating its opinion for a stochastic amount of time. Montes de Oca et al. used Monte Carlo simulations to demonstrate that the individual behaviour they proposed in [123] allows to solve symmetric binary problems where two options of equal quality are available. This method of updating individual opinion requires each robot to memorise a belief about each of the available options in addition to its opinion about which of them is the best. This requires the robot to have a large memory (especially when the number of options is high) limiting the transferability of the behaviour to very minimal swarm robotic platforms.

Scheidler et al. [177] extended the behaviour proposed by Montes de Oca et al. [122] introducing a new individual behaviour for solving the best-of- n decision problem. In the behaviour introduced by Scheidler et al. [177], instead of forming teams and applying the majority rule, robots update their opinions about the best available option following the so-called *k-unanimity* rule. To apply the *k-unanimity* rule, each robot stores the last k opinions received from other robots. If the last k stored opinions are all the same, the robot switches its opinion to the corresponding option. Each robot updates its opinion with a frequency that is inversely proportional to the quality of its selected option, the higher is the quality of the robot's option, the less frequently the robot updates its opinion. After each opinion update, the robot is allowed to capture the opinions of other robots and to share its own opinion with them for a fixed amount of time independent of the quality of the robot's option. Therefore,

robots that are committed to better options share their opinions with others more frequently than robots that are committed to lower-quality options. This increases the chance that robots switch their opinions to options with better qualities leading the swarm to reach consensus for the best available option. [Scheidler et al.](#) analysed the dynamics of the decision process both analytically and using the master equation for up to $n = 3$ options. Their analysis revealed that it is possible to control the speed-accuracy trade-off through the parameter k of the k -unanimity rule. For higher values of the parameter k , the swarm takes more time to reach consensus but makes more accurate decisions and vice versa. The analysis has shown that the proposed behaviour allows breaking the symmetry (*i.e.* allows the swarm to reach consensus when the options have the same quality). Finally, the analysis revealed that the outcome of the decision process is influenced by the number of options n , the initial distribution of the robots' opinions, and the difficulty of the decision problem (*i.e.* how similar the available options are). Additionally, [Scheidler et al.](#) performed a scalability test where they demonstrated that larger swarms make slower but more accurate decisions. However, this scalability test was limited to 50 robots; hence it is not clear how the decision process is influenced in the case of larger swarms. Previous studies showed that increasing the number of robots might increase the swarms' performance up to a critical/optimal number where the swarm's performance starts decreasing [73]. Similar to [Montes de Oca et al.](#) [122], [Scheidler et al.](#) applied their proposed behaviour for solving the shortest path selection task in the well-known double bridge setup [66] with a 10 real-robot swarm. Through these experiments, [Scheidler et al.](#) validated their analytically-obtained results.

[Valentini et al.](#) [211] proposed another behaviour for solving the best-of- n decision problem. The behaviour introduced in [211] is called the *weighted voter model* and is an extension of the classical voter model [109] wherein each time step a randomly chosen agent embraces the opinion of a randomly picked neighbour. The weighted voter model extended the classical voter model by adding a positive feedback mechanism that allows the swarm to reach a consensus for the best option. In this mechanism, each robot disseminates its opinion for a time that is proportional to its option's quality. This mechanism is similar to how house-hunting honeybees employ recruitment waggle dance for quality-dependent durations [215]. Using the weighted voter model, the robot's behaviour is divided into two states, the *exploration* state where the robot visits its option to estimate its quality and the *dissemination* state where the robot moves randomly in a central place to advertise its opinion. The robot remains in the dissemination state for a fixed amount of time. During the dissemination period, the robot advertises its opinion for a time that is proportional to its quality, the better the quality of the robot's option, the longer the robots advertises its option. When the dissemination period is over, the robot applies the classical voter rule by switching its opinion to the opinion of a randomly selected

robot within its communication range. After updating its opinion, the robot switches to the exploration state, where it estimates the quality of its current option. Valentini et al. employed an ODE model, Gillespie simulations [64], and agent-based simulations to analyse the decision dynamics emerging from using the weighted voter model in case of $n = 2$ options. This analysis demonstrated that the weighted voter model enables finite-size swarms to reach consensus for the best of the two options even under robots' noisy quality estimations and when the two options qualities are similar (*i.e.* symmetry-breaking problem). In later works [212, 210], Valentini et al., modified the weighted voter model by replacing the classical voter rule they used in [211] by the majority rule. At the end of the dissemination period, the robot updates its opinion to the opinion adopted by the majority of robots within its communication range. Through ODE analysis in the binary scenario ($n = 2$ options), Valentini et al. [212, 210] demonstrated that decisions using the majority rule are generally faster than using the voter model especially when the quality of the two options are similar. The analysis also revealed that the maximum number of opinions each robot uses when applying the majority rule and the size of the robot's neighbourhood (*i.e.* the average number of robots each robot communicates with at a time) influence the collective decision of the swarm. The higher the value of these two quantities, the less accurate and faster the swarm's decision is, and vice-versa. In [210], Valentini et al. validated the results they obtained previously using ODE analysis through implementation on a swarm of 100 real Kilobots.

Reina et al. [165] proposed another individual behaviour for solving the best-of- n decision problem called the *collective decision through cross-inhibition (CDCI)* strategy. This behaviour is inspired by the decision-making behaviour of the house-hunting honeybees [181]. Reina et al. considered that robots have no prior knowledge about the decision problem; hence robots were initialised in the uncommitted state and had to explore the environment in search for the available options. While exploring the environment, if an uncommitted robot encounters an option, it probabilistically commits to the option. This transition is called *discovery*. In each time step, a committed robot attempts to recruit uncommitted robots within its local communication range by probabilistically sending recruitment messages, and tries to stop other committed robots from advertising competing options by probabilistically sending cross-inhibition messages (also called stop signal [181]). When an uncommitted robot receives a recruitment message, it commits to the corresponding option. This transition is called *recruitment*. When a committed robot receives a cross-inhibition message from a robot that is committed to a different option, it becomes uncommitted. This transition is called *cross-inhibition*. In each time step, committed robots probabilistically decide to spontaneously become uncommitted. This transition is called *abandonment*. Reina et al. [161, 162, 165, 167] linked the probability with which the individual robot performs the previous transitions (*i.e.* *microscopic level*) to

the rates at which these transitions occur at the *macroscopic level*. This micro-macro link allowed [Reina et al.](#) to have a quantitative match between the collective behaviour predicted by the macroscopic model and the results from robot experiments. In [165], [Reina et al.](#) adopted the value-sensitive macroscopic rates proposed in [132]. In this parameterisation, the robots discover or get recruited for a particular option with a rate that is proportional to the option's quality. The robots spontaneously abandon an option at a rate that is inversely proportional to the option's quality and get cross-inhibited (*i.e.* return to the uncommitted state) with a constant rate. Through ODE analysis, this value-sensitive parameterisation has been demonstrated to allow the emergence of interesting collective decision-making behaviours in the binary case [132]. When the qualities of the two options are different enough, the swarm chooses the best option. When the two options have equal (or nearly equal) quality, the swarm randomly selects any of them if their quality is sufficiently high. The probability of the cross-inhibition transition controls the minimum quality above which two options with similar quality are considered to be of sufficient quality, and any of them is selected at random. If the quality is not sufficiently high, the swarm refrains from deciding and remains undecided, waiting for the appearance of a better alternatives. [Reina et al.](#) validated their proposed micro-macro link and the theoretical findings presented in [132] through physics-based simulations of the shortest path selection scenario [165]. In a later study [159], [Reina et al.](#) tested the *collective decision through cross-inhibition* behaviour using a swarm of 150 real Kilobots. This study revealed that its dynamics are slowed due to spatiality that makes the mixing between agents of different opinions harder.

2.1.2 Limitations of the existing best-of- n behaviours

In most of the behaviours introduced earlier, each robot updates its opinion based on the opinions of other robots within its communication range [218, 134, 122, 123, 177, 211, 212, 210]. Thus, the individual robots need to have considerable memory capabilities to store these opinions, especially in high-density swarms. Requiring a large memory limits the transferability of these behaviours to limited platforms such as nano and micro-robots [185, 108, 222]. The behaviour proposed by [Reina et al.](#) [165, 167, 159] limited the robot's memory requirement to a single piece of information. The authors assumed that updating the robot's opinion based on a randomly-selected neighbour's opinion can be reduced to using the last acquired opinion instead of storing the opinions of all the surrounding robots and selecting one of them at random. The reason for this choice is that a robot that is randomly exploring the environment collects the opinions of other robots in random order each time. Therefore, selecting the last received opinion gives the same result as storing all the opinions and choosing one of them at random.

All the behaviours presented earlier (except the one proposed by [Scheidler et al. \[177\]](#)) were tested in the binary case (*i.e.* $n = 2$ options). However, a later theoretical study by [Reina et al. \[164\]](#) on the best-of- n problem revealed a qualitative change in the dynamics of the decision process for $n > 2$ especially when the qualities of the options are similar. The higher the number of options and the more similar the qualities of the options, the harder it is for the swarm to reach consensus for the best option. Moreover, some of the previous behaviours [[218](#), [123](#)] considered the robots' opinions to be initially equally divided between the available options. This assumption opposes realistic scenarios where robots do not know about the decision problem in advance and need to explore the environment to find the available alternatives [[165](#), [167](#), [159](#)]. The environment exploration may lead to biased initial distribution of the robots' opinions which influences the swarm's decisions as has been demonstrated in other behaviours [[122](#), [211](#), [212](#), [210](#), [177](#)].

Almost all previous studies on the best-of- n decision problem [[209](#)] assessed the performance of the swarm in terms of decision accuracy and speed, *i.e.* how often and quickly the swarm selects the best available option. Only the two works by [Prasetyo et al. \[149, 148\]](#) that evaluated the flexibility of the swarm's decisions in the best-of- n problem, *i.e.* the ability of the swarm to adapt its decision in case of environmental changes. These works [[149](#), [148](#)] demonstrated that the weighted voter model introduced earlier [[211](#)] does not allow the swarm to adapt its decision to environmental changes. Additionally, the authors proposed two engineered mechanisms that allow adaptation under the weighted voter model. The first mechanism requires a subpopulation of robots within the swarm to be stubborn. These stubborn robots remain committed to the same option forever and continuously monitor its quality during their exploration phase. When the quality of an option becomes higher than the others, the stubborn robots committed to that option detects the change and reverts the swarm's decision. In the second mechanism, the robots have a probability of spontaneously switching their commitment to other options. This spontaneous opinion switching drives the robots to occasionally reassess the quality of the available options and detect quality changes. However, these mechanisms allow adaptation only in case of quality changes and are incapable of detecting other types of environmental changes such as the appearance of new options or the disappearance of existing options. Moreover, the two mechanisms require the robots to have prior knowledge about the available options, which is generally not possible in realistic scenarios. Besides, the two mechanisms have been tested only in case of $n = 2$ options, and it is not clear how they will be implemented for $n > 2$ options.

2.2 The collective resource collection task

As part of the work presented in this thesis, we looked at the collective resource collection task for its promising implications in numerous real-world applications such as space exploration, search and rescue, and the collection of natural resources [220, 16, 221]. In the collective resource collection task, the robots are required to retrieve objects that are distributed in an unknown environment [83, 84, 82, 85, 143, 53, 111, 112, 144, 2, 110]. In previous works, the objects were either individually uniformly distributed, or grouped into clusters that are uniformly distributed [137, 83, 85]. The cluster-based distribution is closer to the distribution of natural resources in the real-world [168]. To effectively retrieve clustered objects, the robots should rely on both memory and communication [85]. Relying on memory and communication requires more complex individuals and thus limits the transferability of the solutions to simpler robotic platforms. However, these two elements can be implemented at the collective level through the use of stigmergy [76] where robots modify the environment as a way to communicate with others and as a form of shared memory [8]. For this reason, significant attention has been given to the use of stigmergic communication in the collective resource collection task [65, 217, 136, 129, 19, 39, 90, 153, 60, 83, 84, 82, 85, 111, 112]. To design stigmergy-based resource collection behaviour, engineers have mainly been inspired by social insect behaviours, especially the behaviour of some ant species that we overview in Section 2.2.1. In Section 2.2.2, we survey the techniques that engineers have adopted to implement stigmergy-based foraging robots. Finally, in Section 2.2.3, we review the *Central Place Foraging Algorithm (CPFA)* a state-of-the-art stigmergy-based collective algorithm for effectively performing the resource collection task in swarm robotics.

2.2.1 Bio-inspiration

Some ant species coordinate their food collection by leaving pheromone trails when returning from a discovered resource to their nest [219, 91]. In these ant species, the deposited pheromone trails serve as a positive feedback mechanism of mass recruitment which guides nest-mates to the discovered food sources [194]. Foraging ants, equipped with pheromone concentration sensors [199], reach food sources by following the deposited pheromone trails with a preference to higher concentration trails [81, 213, 24]. The modulation of positive feedback (*e.g.* as a function of the source quality [9, 147, 182] or footprint frequency [35]) allows ant colonies to reach various collective patterns, such as selecting the best-quality food source available in the environment [10, 9, 158, 182], selecting the shortest path linking the food source to the nest [66, 33], and balancing predation-risk and food quality [127].

In addition to the ability of collective resource exploitation, adaptation to environmental fluctuations is a critically important ability for many biological organisms [204], including foraging ants [41]. The mechanisms behind mass recruitment abilities (*i.e.* positive feedback) are generally in opposition to those that allow adaptation and flexibility [196, 204], therefore, organisms showing adaptability are generally capable of more complex behaviour. A remarkably interesting example is offered by *Monomorium pharaonis* ants which make use of repellent pheromone as a form of negative feedback [192, 171, 172, 34]. Ants use this repellent pheromone to mark unrewarding trails, and this could be a strategy to stop the exploitation of trails that lead to depleted food sources. Other evidence of adaptability in ants has been documented by [10] who showed that *Tetramorium caespitum* ants are able to refocus their foraging efforts from a previously selected lower-quality food source, to a newly available higher-quality food source. Ants of this species can adapt to the environmental changes because, in addition to pheromone-based recruitment, they use tandem running to recruit ants to newly available higher-quality food sources [10]. In contrast, *Lasius niger* ants, using pheromone-based recruitment only, are unable to switch their foraging efforts to the newly available food source. In fact, *Lasius niger* ants only rely on indirect forms of negative feedback, which may arise from physical constraints at the food source (*e.g.* overcrowding or food depletion) or within the nest (*e.g.* filling of food reserve) [34]. Finally, in another study, [182] showed that *Temnothorax rugatulus* ants employing quality-dependent linear recruitment and quality-dependent abandonment are able to adapt to the environmental changes. *T. rugatulus* ants select the best-quality food source in the case of two unequal-quality sources, equally exploit the two sources if they have equal qualities, and refocus their foraging efforts in case of changes in relative qualities [182].

2.2.2 Implementation of stigmergic communication in robotics

To implement the pheromone-based recruitment mechanism in a robot swarm, an important question concerns the means of implementing pheromone trails; in particular, how the robots deposit pheromone, how the pheromone trails in the environment evolve, and how pheromone can be sensed by the robots. Here, we categorise state-of-the-art work in this area into three main approaches: beacon robots, robots with onboard actuators and sensors, and smart environments.

In the first category of robotic systems, some robots are tasked as static beacon robots [65, 217, 136, 129, 19, 39, 90], which have the functions of storing pheromone levels and communicating with other robots in their neighbourhood. The most significant advantage of this approach is that the system can be implemented with simple robots in largely unknown and unstructured environments. However, there are some limitations: (i) allocation of beacon robots means they are not actively contributing to the main task, such as foraging; (ii) in large

environments, the number of beacon robots increases in order to cope with the communication requirements, thereby further limiting the number of robots performing main tasks; (iii) beacon robots become obstacles themselves which restrict the movements of other robot agents. These issues can be overcome by the creation of mobile beacon robots, which can contribute to the main task as well as acting as beacons concurrently [191, 40]. However, the performance of the latter approach relies on finding the correct balance between the swarm size and the communication range as a function of the environment size.

Researchers have made several attempts to equip robots with onboard actuators and sensors to implement indirect communication. For example, one early solution by [Svennebring and Koenig \[195\]](#) was to install marker pens on robots so they could draw lines on the path as pheromone trails. This method improved robots' performance in the area coverage task; however, it did not incorporate pheromone evaporation or diffusion which are features of real ant trails; evaporation, in particular, is considered important to avoid runaway positive feedback [60, 56]. Another design proposed by [Purnamadajaja and Russell \[153\]](#) equipped robots with devices to emit and detect gas, which then provided guidance to robots towards a source area. The main limitation of this design was the high volatility of the chemicals used. In [118], [Mayet et al.](#) proposed a technique of energising phosphorescent paint using UV-LEDs mounted on E-Puck robots to mark the path, as well as sensors for picking up the glowing paint signal representing the pheromone trail, was presented. Although this allowed emulation of pheromone decay, diffusion could not be emulated. A more recent study by [Fujisawa et al. \[54, 55\]](#) used ethanol for indirect communication signals between robots, with an ethanol pump and an ethanol sensor installed on each robot. This implementation preserved the four characteristics of pheromone: evaporation, diffusion, locality (*i.e.* pheromone level is only affected by the local environment) and reactivity (*i.e.* pheromone evolution is based on reactions with the environment).

Perhaps the most popular approach in implementing pheromone communication is through a smart environment [193, 60, 83, 56, 5, 207, 49, 197, 125], which has the capability to store and to supply virtual pheromone information to robot agents in real-time. The popularity arises from the fact that this approach is generally low-cost and easily adaptable to different sizes of swarm and environment. Smart environments may be difficult to install and use for real applications; rather, such setups are often employed for targeted research experiments. This category can be further divided into three classes: the usage of (i) Radio-Frequency Identification (RFID) tags [115, 116, 88, 87, 15, 101]; (ii) simulated pheromone environments, using projected light or other custom hardware for virtual pheromone implementations [193, 60, 56, 5, 207], and (iii) augmented reality tools in which a virtual environment is sensed and acted on by robots using virtual sensors and actuators [166, 160].

2.2.3 The Central Place Foraging Algorithm (CPFA)

Previous research works proposed multiple pheromone-based individual behaviours for solving the collective resource collection task in swarm robotics. The behaviour that received the most attention in recent years is the so-called *Central Place Foraging Algorithm (CPFA)* introduced by Hecker et al. [83], in which robots rely on both individual memory and pheromone communication to effectively collect the available resources in an unknown environment. In the CPFA, the robot is initially positioned at the depot where the collected resources are stocked, and it starts its collection trip by visiting a random location in the environment called the *search site*. Once at the randomly selected search site, the robot performs a correlated random walk in search of the available resources. While searching for resources, the robot has a fixed probability to give up and return to the nest, and when finding a resource, the robot brings it back immediately to the depot. If the robot detects more than a single resource at the same site, the robot memorises the corresponding location. The robot uses the memorised location to return to collect the remaining resources. This action is known as *site fidelity*. Additionally, on its way back to the depot, the robot lays pheromone trails to indicate the clustered resource's location to other robots. This action is referred to as *recruitment*. Hecker et al. considered different distributions of the resources: the random distribution where single resources are uniformly distributed in the environment, the clustered distribution where fixed-size clusters of resources are placed at set locations, and the power-law distribution where clusters of power-law-distributed sizes are placed at selected locations in the environment. The authors tested the CPFA's performance using both simulated and physical robots by evaluating the rate at which the robots collect the resources for the different distributions, both when robots act individually or in teams of three. The results revealed that for all distributions the robots collect resources almost twice as fast when acting in trios than when acting individually. Moreover, the collection is faster when using pheromone trails (*i.e. recruitment*) in addition to individual memory (*i.e. site fidelity*), especially for the clustered distribution.

A later work by Letendre and Moses [107] used Genetic Algorithms (GA) to find the CPFA's parameters that maximise the foraging rate. In this work, the authors demonstrated that combining site fidelity and pheromone-based recruitment allows the swarm to achieve the highest foraging rates for all distributions. For the clustered and power-law distributions, using only site fidelity allows better performance than when using pheromone-based recruitment only as it allows the robots to better find resources in known areas. However, the authors suggested that the use of pheromone is better suited in case of depleting resources as it allows information about the available resources to spread faster within the population. Therefore, the authors concluded that using both mechanisms enables the swarm to guarantee the required performance in the different possible scenarios. Another work by Hecker and Moses [84]

employed GA to optimise the parameters of the CPFA in case of positional and resource detection errors which degrade the performance of the swarm. The use of GA revealed that the best strategy in case of positional errors is to spend less time searching for local resources and to rely less on site fidelity. In case of detection error, the best strategy is to deeply search local areas and to increase the use of pheromone-based communication. However, the strategy suggested by the GA in case of a combination of both errors did not improve the performance. The authors suggested that this is due to the contrasting effect of the two types of errors. The GA approach has also been used by [Hecker and Moses \[85\]](#) to determine the combination of the CPFA's behavioural rules that maximises the performance of the swarm in case of different resource distributions, sensing errors, and swarm sizes. For low detection errors and highly clustered resources, the best strategy is to rely on pheromone communication. When the resources are distributed in clusters of variable size, the robots should rely on site fidelity. In the case of larger swarms, the robots' motion has to rely on directed motion to better diffuse in the environment and overcome crowding.

[Hecker et al. \[82\]](#) extended the CPFA to increase the effectiveness of the algorithm at the end of the collection process when fewer resources remain in the environment. Following this extension, when 90% of the resources are collected, the robots switch to the so-called *clusters exploitation* behaviour, in which they concentrate their searching efforts in selected regions in the environment. These regions are computed centrally at the depot based on the location of the previously collected resources provided by the robots. Improvements resulting from the clusters' exploitation are better appreciated in scenarios where resources are highly clustered in the environment.

The need to bring the retrieved objects to a central depot limits the effectiveness of the CPFA algorithm as it leads to long travel times [\[85\]](#) and crowded areas [\[53\]](#). To mitigate against this issue, [Lu et al. \[111\]](#) extended the CPFA to the case where multiple static depots are distributed in the environment introducing the *Multiple-Place Foraging Algorithm (MPFA)*. The goal of the MPFA is to limit the effect of overcrowding observed in larger swarms and the longer travel times experienced in wider environments on the CPFA's performance. In the MPFA, at the start, the robots are randomly distributed over the multiple depots. During the collection process the robots bring back the collected resources to the nearest depot. Except for these two differences, the individual robot has an identical behaviour as in the CPFA. Using physics-based simulations, [Lu et al.](#) demonstrated that the MPFA outperforms the CPFA as it reduces collisions between robots and results in shorter travel times. A later work by [Lu et al. \[112\]](#) introduced a dynamic version of the MPFA where special robots act as depots that continuously adjust their positioning in the environment to optimise the collection process. The dynamic depots adjust their position by moving to the centroid of the resources' areas

that the collecting robots recently discovered. The dynamic MPFA allows faster collection than the CPFA and the static MPFA in both larger environments and for larger swarms. The performance of the dynamic MPFA remains dominant over the other strategies, even under sensing and navigation errors.

2.3 Conclusion

The work presented in this thesis introduces individual behavioural rules for improving the performance of robot swarms in two well-known swarm robotics tasks. The first task is the best-of- n decision problem where robots are required to reach a consensus for the best option out of n available alternatives. Solving the best-of- n decision problem is considered to be a fundamental cognitive skill of robot swarms [209]. Moreover, the best-of- n problem has a general formalisation that covers multiple decision-making problems such as the symmetry-breaking problem where the options are all the same and the robots only need to select any of them. For these reasons, a considerable amount of literature proposed individual behaviours that allow robot swarms to solve the best-of- n problem. Some works proposed task-specific behaviours that allow to solve the best-of- n problem within specific collective tasks (*e.g.* the aggregation task). Other works proposed generic behaviours for solving the best-of- n problem independently of the main task of the swarm. In this chapter, we reviewed these task-independent behaviours and highlighted their limitations.

The second task addressed in this thesis is the collective resource collection task where robots are required to retrieve objects spread in an unknown environment. This task has various potential real-world application, including space exploration and natural resources collection [220, 16, 221]. Natural resources are generally distributed into clusters [168]. To effectively retrieve clustered objects, the robots need to use both memory and communication [85]. These two capabilities can be implemented at the collective level using stigmergy [76] without the need for individually-complex robots. Stigmergy is a communication mean in which robots mark the environment to communicate with others and to create a shared memory. In this chapter, we reviewed the behaviour of some ant species that mark the environment using pheromone trails to coordinate their efforts when collecting nutrient from the environment. The behaviours of these ant species have been the main source of inspiration for many stigmergy-based resource collection behaviours in swarm robotics. Besides, we looked at the different techniques employed by researchers to implement stigmergic communications in swarm robotics studies. Finally, we introduced the CPFA [83], a state-of-the-art stigmergy-based algorithm for solving the resource collection task, and reviewed its evolution in the past few years.

Chapter 3

Analysis of swarm robotics systems

Swarm robotics systems belong to the family of complex systems. Complex systems are generally very difficult to understand due to their dependence on a large number of parameters and the non-linearities they contain. To cope with the complexity of these systems, researchers study them following a step-by-step process [106, 79, 94, 12, 214]. In each step, the system is represented using an appropriate model which simplifies the system by discarding some of its parameters. This allows researchers to better understand the effect of the remaining parameters. In this thesis, two types of models are used to study swarm robotics systems, chemical reaction models and agent-based models. While these models are introduced in later chapters, the tools used to analyse them are presented in this chapter. Section 3.1 introduces the Gillespie algorithm [64, 154] used to analyse chemical reaction models. Sections 3.2 and 3.4 present the multi-agent simulator and the physics-based swarm robotics simulator, respectively, employed in this thesis to analyse agent-based models. To make a step closer toward the application of swarm robotics systems in the real world, in this thesis, we implement some of the designed individual behaviours on real robots. The swarm robotics platform used for real robot experimentation is presented in Section 3.3.

3.1 Stochastic simulations

In this section, we introduce the Gillespie algorithm [64, 154] used to analyse the chemical reaction models studied in this thesis. Chemical reaction models are employed to describe swarm robotics system. Chemical reaction models symbolise swarm robotics systems as chemical systems where the molecules are the robots, the chemical species of the system are the sub-populations of robots in different states, and the chemical reactions represent the interactions between robots and between robots and the environment.

Chemical (or swarm robotics) systems are stochastic systems as each reaction (or interaction) has a probability of occurring per unit of time. The time evolution of these systems can be described analytically by the so-called *master equation* which is a function of independent variables representing the populations of the chemical species (or robots sub-populations). The solution of the master equation gives the probability of the presence of each molecular species (or robots sub-population) at each instant of time. For systems involving large numbers of species and chemical reactions, analytically solving the master equation is strictly impossible. For this reason, Gillespie [64] proposed a computational method to determine the stochastic time evolution of theoretically any spatially homogenous chemical system (or well-mixed swarm robotics system) independently of solving the master equation. The algorithm proposed by Gillespie [64] employs Monte Carlo methods [80] to simulate the Markov process that describes the system's evolution and hence the solution of the master equation. The Gillespie algorithm is simple but efficient as it can simulate systems with a large number of species and complex reactions. The only requirement for the Gillespie algorithm to be usable is to describe the system as a set of chemical reactions and assign a rate of occurrence for each reaction.

Using the occurrence rates of the chemical reactions governing the system and the initially-known number of molecules of each chemical species, the Gillespie algorithm describes the probability that a given reaction happens after a given amount of time as a time-dependent exponential probability density function (*PDF*). By using the *PDFs* of all the reactions and by applying the Monte Carlo method [80], the algorithm randomly draws which reaction will happen next and when that will be. The algorithm then updates the number of molecules of each chemical species, hence the *PDFs*, and the simulation time. This procedure is then repeated until the simulation reaches a specified stopping time. Full technical details of the Gillespie algorithm are given in [64].

In the original Gillespie algorithm, the occurrence rates of the chemical reactions are considered to be constant throughout the simulation. However, in some systems, in biochemistry, for example, the occurrence rates of the chemical reactions vary over time due to external factors such as changes in temperature and volume. For this reason, Puritan and Udrea [154] proposed a method to make the Gillespie algorithm usable for systems of time-dependent rates. Puritan and Udrea's method is applicable to the version of the Gillespie algorithm using the first-reaction Monte Carlo technique. In its first-reaction version, the original Gillespie algorithm determines the next reaction to happen and the time that will take by integrating the *PDF* of each reaction over time to compute the average time required for the reaction to happen, then considers that the next reaction to happen is the one with the minimum time (for details see [64]). Puritan and Udrea [154] suggested that this version of the Gillespie algorithm can be used to simulate the time evolution for systems with time-variant rates of known time

functions. [Purtan and Udrea \[154\]](#) achieved this by replacing the constant rates in the original version by the time functions of the time-variant rates when integrating the *PDFs* over time to find the time for the next reaction to happen.

In this thesis, we employ both the original Gillespie algorithm [\[64\]](#) and [Purtan and Udrea's](#) extension [\[154\]](#) to study swarm robotics systems with constant and time-variant interactions (see Chapter 4).

3.2 Multi-agent simulations

Multi-agent simulators are used in swarm robotics to analyse and validate the designed collective behaviours. Multi-agent simulations are minimalistic as they ignore realistic aspects such as physical interactions between robots, and generalist as they do not aim to represent any specific robotic platform. However, multi-agent simulations are richer than the stochastic simulations of Section 3.1 because they consider additional aspects such as local interactions between agents and spatiality that have been proved to affect collective behaviours [\[159\]](#). In this section, we introduce DeMaMAS, the multi-agent simulator employed in this thesis.

In swarm robotics, researchers often use the MASON [\[113\]](#) multi-agent simulator for its speed, flexibility and ease of use. In this thesis, we used multi-agent simulations to analyse the designed collective decision-making algorithms only. For this reason, we decided to employ DeMaMAS (Decision Making Multi-Agent Simulator) [\[21\]](#), a simulator designed specifically for collective decision-making research. In DeMaMAS, robots are represented by small coloured circles and options (*i.e.* choices) are represented by bigger colour-coded circles (see Figure 3.1). The colour of an agent indicates its choice and its motion state. For example, the red agents in Figure 3.1 are committed to the red-coloured option and are exploring the environment through an isotropic random walk, light-red agents are committed to the red-coloured option and are on their way to measure its quality. Uncommitted agents are indicated by grey colour. In DeMaMAS, the environment is a square which satisfied periodic boundary conditions [\[178, 156\]](#). When an agent crosses a border, it appears on the opposite one, and when an option is located at a border, it will be seen from the opposite one. This type of environment allows to avoid spatial correlations that may result from having hard borders.

In DeMaMAS, the agent is composed of three main parts (see Figure 3.2) which are the sensors, the actuators, and the mind. The sensors enable the agent to acquire the information necessary to make a decision. The sensors are the eyes and ears which allow the agent to perceive the available options and receive information from other agents, respectively. The actuators are the feet and the mouth which allow the agent to explore the environment in search of the available options, and to disseminate their belief to the other agents, respectively.

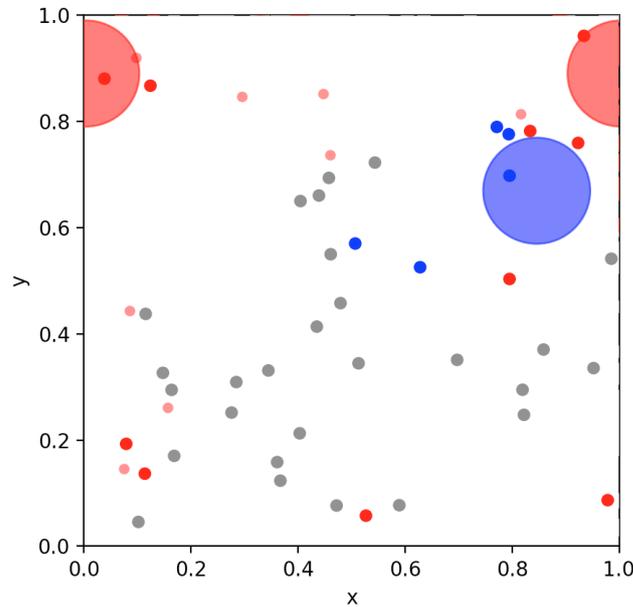


Fig. 3.1 The visual interface of DeMaMAS. The big circles represent the options (*i.e.* the choices) while the small circles represent the agents. The colour of an agent indicates its choice and its motion state. For example, the red agents (*i.e.* the small red circles) are committed to the red-coloured option and are randomly exploring the environment. The light-red agents are committed to the red-coloured option and are on their way to measure its quality. The grey agents are uncommitted.

The mind embeds the agent's collective decision-making model. The mind is in charge of dealing with the information acquired by the sensors, controlling the actuators, memorising the information required by the agent, and updating the agent's decision.

DeMaMAS proposes a general and modular structure for the agent's mind and hence for collective decision-making models. As depicted by Figure 3.2, the proposed structure divides the agent's mind into four main components. The first component is the *Sensors Information Processing* which processes the information received through the sensors and converts it into a single piece of usable information. This information is then transferred to the second part of the mind, the *Update Opinion* component, which is in charge of updating the agent's decision. The third main component of the mind is the *memory* where the agent stores their opinion, parameters of some mind functions, and the information received from the sensors (before their processing). The fourth component of the mind is the *Actuators Control* which controls the navigation and the communication of the agent by setting the agent's motion and shared information.

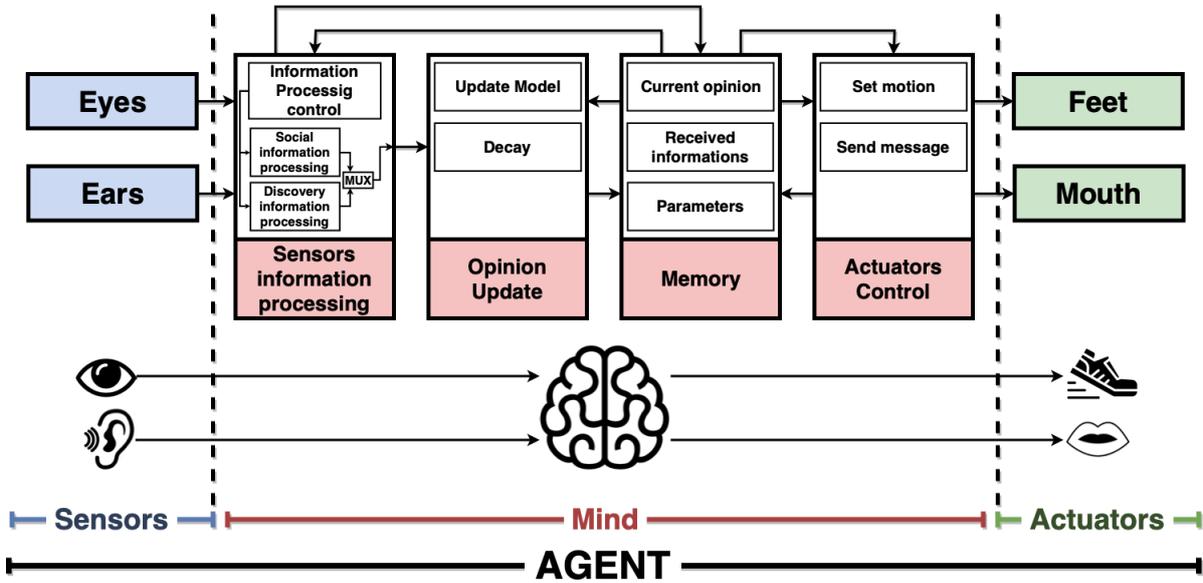


Fig. 3.2 The general framework of collective decision-making agent-based models proposed in DeMaMAS.

Each of the previous components of the mind is composed of multiple unitary modules. For each of these modules, DeMaMAS offers a set of configurations which are basic behavioural rules of collective-decision making algorithms. The available configurations of the unitary modules proposed by DeMaMAS are listed and described in detail in [21]. By setting the configuration of each module, the user is able to build existing collective decision-making models. Moreover, the user can make new combinations of the proposed configurations to build new collective decision-making models. In [20], for instance, DeMaMAS was used to build and compare existing decision models such as the cross-inhibition model [132, 165], the weighted voter model [211], and the k -Unanimity model [177]. Additionally, by varying some modules of the existing models, [20] were able to produce novel collective decision-making models which are more resilient to external attacks.

3.3 Swarm robotics implementation

Stochastic and multi-agent simulations of Sections 3.1 and 3.2 represent effective tools to analyse collective behaviours. However, implementation on robot swarms gives much more complete tests as it takes into account additional aspects such as physical interaction between robots. For this reason, the collective behaviours designed in this thesis are often tested on both real and simulated robot swarms. In this section, we introduce the Kilobot swarm we used to

conduct our swarm robotics experimentation. Details on how we simulated the Kilobot swarm are given in Section 3.4.

3.3.1 The Kilobot robot

The Kilobots (Figure 3.3) are very simple, low cost, scalable, and easy-to-operate robots [174]. Due to these interesting features, the Kilobots quickly became a well-known and broadly used swarm robotics platform. However, the Kilobots have only three actuators and two sensors which make their capabilities very limited. The Kilobot is equipped with two vibration motors that enable its slip-stick differential-drive locomotion. The Kilobot is able to move forward at a nominal speed $v_0 \approx 1 \text{ cm/s}$ and rotate in place both in clockwise and anti-clockwise directions at a rotation speed $\omega_0 \approx 45^\circ/\text{s}$. The Kilobot has an infrared (IR) transceiver that allows it to communicate with other Kilobots in a range of $\sim 10 \text{ cm}$. Additionally, the IR transceiver allows the Kilobots to receive information from external IR emitters such as the overhead controller (OHC) device used to program large numbers of Kilobots simultaneously. The OHC can also be used to send personalised information to the Kilobots, thus researchers employed it to equip the Kilobots with virtual sensors (see Section 3.3.2). The Kilobot is also equipped with an RGB LED that can be used to communicate its internal state to the user or external systems. For instance, as shown in Section 3.3.2, the Kilobot is augmented with virtual actuators that it operates using its RGB LED. Finally, the Kilobot is equipped with an ambient-light sensor.



Fig. 3.3 A picture of the Kilobot robot

3.3.2 The Augmented Reality for Kilobots (ARK) system

Due to the low number of sensors and actuators the Kilobot supports, the experimental paradigms that can be achieved using the Kilobot are very limited. To widen the range of the experimental paradigms where the Kilobot can be used, open-source technology has been developed to expand the Kilobot's sensory and actuation capabilities via customisable virtual sensors and actuators [160, 207]. This technology uses the concept of augmented reality to allow the Kilobots to sense and modify computer-simulated virtual environments in addition to the real world. In recent years, this technology has been proposed by two different implementations, the Augmented Reality for Kilobots (ARK) [160] and the Kilogrid [207] systems.



Fig. 3.4 Picture of the Kilogrid system (image source [207]).

The Kilogrid consists of an electronic table composed of hundreds of IR transceivers and LEDs (see Figure 3.4) to real-time communicate with the Kilobots moving on the table. Via the Kilogrid, each Kilobot can communicate and receive considerable amounts of data at high frequency. The drawback of Kilogrid is its high installation costs. In contrast, the ARK system reduces the communication frequency between robots and the virtual world but consists of a cheap and efficient virtualisation system for Kilobots (see details in [160]). In this thesis, we employ the ARK system due to its low installation cost and its ability to automatically perform several time and effort-demanding house-keeping tasks such as motor calibration, unique ID assignment, and experiment video-recording.

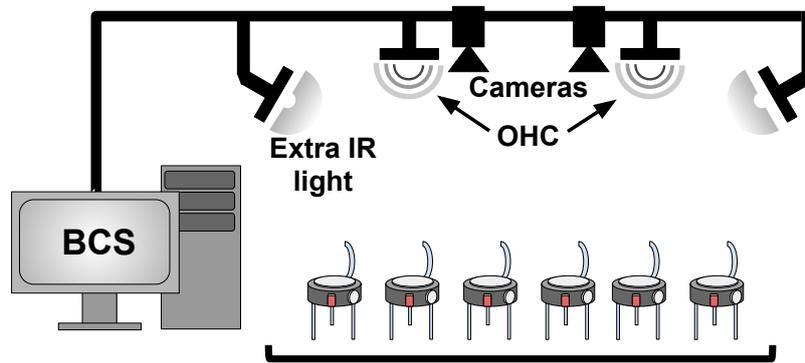


Fig. 3.5 Graphical representation of the ARK architecture (image source [160]).

ARK consists of an overhead camera array to track the Kilobots, an array of IR OHC to communicate to the Kilobots, and a computer (base control station, BCS) to run the ARK software and simulate the virtual environment (see the system architecture in Figure 3.5). The information about the virtual sensors is computed on the BCS, depending on the location of each robot in the virtual environment. This information is then communicated to the specific robot with addressed messages via the OHC. The OHC sends 9 bytes messages (similar to the messages sent between Kilobots) to address three robots at a time, where the information sent to each robot is packed in 3-byte long *ARK messages*. In the ARK message, 10 bits are used to specify the unique Kilobot address assigned to the robots via ARK prior to the experiment. The remaining 14 bits of the ARK message are used to store information about the robot's virtual sensors. Virtual actuation is computed onboard by the Kilobots, communicated with colour-coded messages via LEDs visible by the overhead cameras, and processed by the BCS, which updates the virtual environment accordingly. Additionally, the BCS updates the temporal dynamics of the virtual environment. In this way, each Kilobot can receive personalised information about its virtual sensors depending on its real-time physical position computed through a robust vision algorithm which continuously keeps track of ID-assigned Kilobots. The Kilobot can then autonomously decide when to modify the virtual environment through virtual actuators. An ARK experiment can be composed of multiple virtual environments of different structure and spatio-temporal dynamics.

Here is an example to understand the functioning of the ARK system. Let us imagine a scenario where the robots are tasked to find and transport a resource to a specific location in the environment. The ARK system will simulate the presence of this resource virtually (i.e. in the BCS) while the robots are operating in the real environment. ARK tracks the robots' positions using cameras and interposes the virtual environment (i.e. the resource) on the captured images.

ARK equips the real robots with a virtual sensor to detect the simulated resource. This is achieved by comparing the position of the robots and the position of the virtual (i.e. simulated) resource. ARK informs the robots about the resource's presence when the resource is within a defined distance (i.e. sensing range) from the robot's position. The decision to interact with the resource is made by the robots internally. The robots communicate their interaction choices to the ARK system via their RGB LEDs. For example, the robot can show a blue LED to inform the ARK system about its decision to transport the resource. ARK will then update the virtual environment accordingly, i.e., move the virtual resource according to the robot's movement. More examples on the functioning of the ARK system can be found in the later chapters and in [160].

3.3.3 Kilobot's 3D-printed add-on

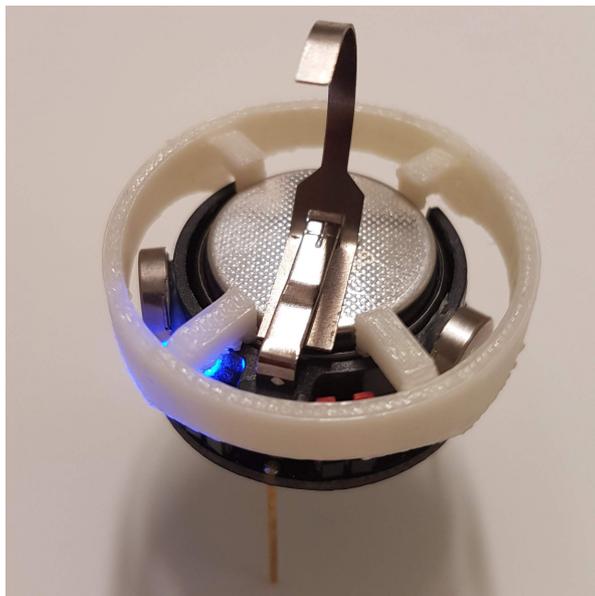


Fig. 3.6 A picture of a Kilobot with a 3D-printed ring (originally designed for the study of [150]) which considerably improves ARK's performance in terms of tracking and LED colour detection.

In this thesis, we improved the ARK tracking system by applying to the Kilobots a 3D-printed light structure illustrated in Figure 3.6. This structure brings the double advantage of improved crowded tracking and LED colour detection. In fact, in crowded situations, Kilobots without the ring form a big black mesh and individual robots were difficult to track. Instead, when using the 3D-printed ring, Kilobots always remain slightly separated, and in the overhead images, they appear as distinct easy-to-track circles. Additionally, the 3D-printed plastic ring

reflects and diffuses the Kilobot's LED light resulting in an improved overhead camera colour detection in any orientation. This feature is particularly useful for ARK's virtual actuators that are implemented through colour-coded LED messages. The 3D-printed structure is open-source and available at <https://diode.group.shef.ac.uk/kilobots>

3.4 Swarm robotics simulations

Results in swarm robotics studies are mostly (when not entirely) produced in simulation [16, 51]. This is because experimentation and debugging with real robots is effort-and-time demanding due to the numerous hardware issues, battery limitations, and logistic adversities. Additionally, using swarm robotics simulation researchers are able to evaluate the designs and parametrisations of their produced behaviours, isolate the process from distracting factors, and test cases which are not achievable in a research lab such as using large swarm sizes and large environments. In order to offer an alternative to real robot experiments, swarm robotic simulations need to be accurate enough to represent reality and fast enough to allow the user to perform the design cycles required to achieve desired swarm behaviours. Here, we introduce the ARGoS simulator [139, 138] used in this thesis to simulate the Kilobot swarm. We also present how this thesis contributed to the improvement of the ARGoS Kilobots extension. Finally, we introduce the simulated counterpart of the ARK system, which we added to the ARGoS simulator.

3.4.1 ARGoS simulator

Since the release of the Kilobots in 2012, new simulators such as Kilombo [96] and KBSim [72] were specially developed to simulate them. Additionally, existing simulators such as V-REP [173] and ARGoS [139, 138] were upgraded to support the Kilobots. In this thesis we employ the ARGoS simulator [139, 138]. The main reason why ARGoS is used instead of the other simulators is its ability to control the trade-off between the simulation speed and accuracy. In contrast to Kilombo and KBSim which are very fast but minimalistic, or V-REP, which is accurate but very slow, ARGoS offers simulations of acceptable accuracy at a reasonable speed. ARGoS achieves a good accuracy by using accurate Kilobot models which implement features observed in reality. The models used by ARGoS were validated against real-world behaviours in appropriate experimental setups (for details see [138]). Kilombo and KBSim, instead use highly simplified models that were never validated. ARGoS achieves reasonable speed and scalability (simulates 1000 robots in real-time) through its multi-threaded architecture which takes advantage of the computational power of multi-core processors. Moreover, to improve the

speed-accuracy trade-off, ARGoS allows the user to specify which elements of the experiment are influential and thus require accurate simulation, and which elements are negligible and thus can be coarsely simulated. For instance, the user can decide to which extent physics should be simulated by selecting one of the supported physics engines.

Another important feature offered by the ARGoS simulator [139] in its Kilobots extension [138] is the support of cross-compiling, which enables the use of the same control software in simulation and with real robots. This allows the user to avoid bugs that may result during the process of translating algorithms between different platforms and to save time by applying improvements and modifications to a single code instead of two. In contrast, KBSim and V-REP do not support this feature and require the user to write two different codes, one for simulation, and one for the real robots. Kilombo, instead, achieves cross-compilation between real and simulated robots by modifying the original Kilobot API. The new Kilobot API required by Kilombo sets restrictions on the way the Kilobot control software can be written and make any improvements or patches released for the Kilobot not directly usable (requiring adaptation to the Kilombo API).

3.4.2 Minimising the reality-gap for reliable swarm robotics simulations

In this thesis, we contributed to the development of ARGoS Kilobots extension [138] by improving its accuracy (*i.e.* minimising the reality-gap). We modelled the noise and inter-individual variations observed in the Kilobots' motion. We also tuned the physical interaction simulation to better simulate collisions between the robots. These contributions were published in [138] and are presented in this section.

3.4.2.1 Modelling of noise and inter-individual variations

The Kilobot is able to move on a flat surface through the slip-stick differential-drive locomotion produced by its two vibration motors. This locomotion system is influenced by small variations such as the position of the motors and the bending of the robot's legs, resulting in strong inter-individual variations. For this reason, before each experiment, Kilobots are individually calibrated to perform their fundamental motions, *i.e.* to move forward and to turn on the spot at desired speeds. The calibration process is very difficult as it depends on which surface the Kilobots move. While most researchers rely on manual calibration, some labs possessing the ARK system (see Section 3.3.2) rely on its automatic calibration functionality to simultaneously calibrate tens of robots. Although ARK allows to save the time and effort invested in manual calibration, the motion of its calibrated robots is still subject to the high noise intrinsically present in Kilobot slip-stick locomotion. Therefore, Kilobots, either when manually or auto-

matically calibrated, are never able to perfectly move straight or rotate in place at their normal speeds. To improve the accuracy of Kilobot simulations, the noisy motion observed in real robots must be reproduced in simulation. To achieve this in ARGoS, we included a noise component in the Kilobots' motion model. We then tuned the noise component to match the motion observed in reality.

We simulated the Kilobot motion using a differential-drive locomotion model where we included a noise component in the speed of each wheel. The applied left (ℓ) and right (r) speeds \hat{v}_i (with $i \in \{\ell, r\}$) are computed as a function of nominal speeds v_i and noise components as follows:

$$\hat{v}_i = f_i^t(v_i + b_i), \quad i \in \{\ell, r\} \quad (3.1)$$

Where f_i and b_i are Gaussian-distributed random parameters with user-defined mean and standard deviations, representing per-step actuation noise and per-robot bias added to the nominal speed v_i , respectively. For each robot, the actuation noise f_i with $i \in \{\ell, r\}$ are drawn from the specified Gaussian distributions at each simulation time-step. The per-robot bias b_i , instead, are drawn only once at the beginning of the experiment.

To reproduce the noisy motion experienced with real Kilobots, we tuned the motion model of equation (3.1) through experiments performed on a sample of 120 real Kilobots. First, we set the nominal (noise-free) speeds v_i of equation (3.1) to values that produce the Kilobot nominal speeds. We put $v_i = 1 \text{ cm/s}$ for $i \in \{\ell, r\}$ to achieve the average Kilobot's forward speed of $\simeq 1 \text{ cm/s}$. We set $v_\ell = 2 \text{ cm/s}$ and $v_r = 0$ for a $\sim 45^\circ/\text{s}$ clockwise rotation and the opposite for anti-clockwise rotation. We then conducted experiments using 120 different calibrated Kilobots (6 robots per experiment) to tune the noise distributions of the model of equation (3.1) (*i.e.* distributions of f_i and b_i). In each experiment, we asked the Kilobots to move forward for 60 s and recorded their trajectories using ARK. We then divided the trajectory of each robot into 10 s displacements. For each displacement, we computed the corresponding speeds \hat{v}_i (with $i \in \{\ell, r\}$) of the differential drive model of equation (3.1). Then using \hat{v}_i , we computed the bias $b_i^t = \hat{v}_i - v_i$ (ignoring white noise, *i.e.* $f_i = 1$) for each 10 s displacement (hence $t \in \{1, \dots, 6\}$ for our 60 s experiments). We then computed the average bias $b_i = \sum b_i^t / 6$ for each robot, and we report the distributions of biases (for both left/right velocities $i \in \{\ell, r\}$) of the 120 tested Kilobots in Figure 3.7(a). Finally, we computed the mean $\mu_b = 0.015 \text{ mm/s}$ and standard deviation $\sigma_b = 1.86 \text{ mm/s}$ of the determined distribution. These values are implemented as default noise values in ARGoS, but the user, if needed, can set their values.

To show the effect of the noise implementation on the accuracy of ARGoS Kilobots simulator, we reproduced the experiments we conducted with real Kilobots in simulation and ran them both with noise and without noise. We then compared the mean square displacement (MSD) of the 120 real Kilobots and the simulated robots. As shown by Figure 3.7(b) the

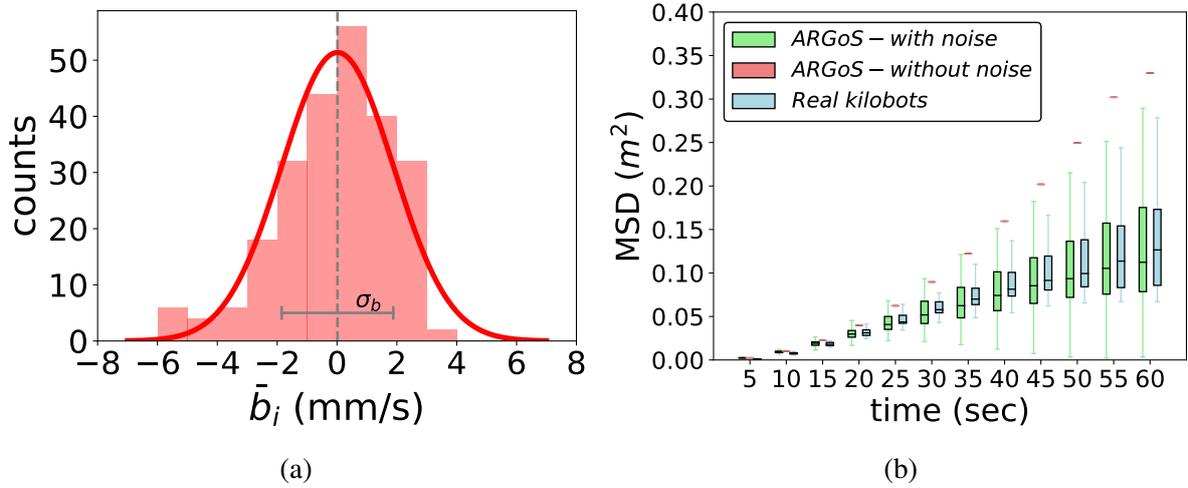


Fig. 3.7 ((a)) Distribution of the bias in straight motion estimated from measurements over 120 different Kilobots which have been previously calibrated (60 manually, 60 automatically). The bars consider 12 bins in the range $[-6, 6]$ mm. The solid red line shows the approximated Gaussian distribution $\mathcal{N}_b(\mu_b, \sigma_b)$ (with mean μ_b and standard deviation σ_b) used in the default configuration of ARGoS. ((b)) Comparison between simulation (600 robots) and reality (120 robots) in term of MSD when robots are asked to go straight for 1 minute. The default noise values of ARGoS give an accurate match between reality and simulation.

noise-free simulations have a huge difference from real robot experiments. Instead, when noise is added, the dynamics of the simulated robots have an outstanding match with reality.

3.4.2.2 Tuning physical interactions simulation under ARGoS

As described in Section 3.3.1, the Kilobots have no sensors which enable the implementation of collision avoidance algorithms. For this reason, collisions between robots and with the environment are highly frequent in real Kilobot experiments. Hence, the correct simulation of physical interactions between Kilobots is expected to improve the overall accuracy of the Kilobot simulation. Here, we tune the ARGoS physics engine in order to improve the simulation of physical interactions and thus the overall accuracy of the Kilobot simulation. We also compared the ARGoS and Kilombo simulators in term of physical interactions simulations.

In ARGoS, physical interactions are simulated using the full dynamics of modern rigid-body simulation engines [121, 139, 173]. An important and tunable parameter of these physics engines is the friction between the bodies of colliding robots. Here, we aimed to reach a match between the real and the simulated physical interactions by tuning the friction value. For this purpose, we designed a repeatable experimental setup in which collisions between Kilobots are maximised. In the designed experiment, we initially placed 50 Kilobots in the compact distribution shown in Figure 3.8(a). The Kilobots are placed on the vertices of four

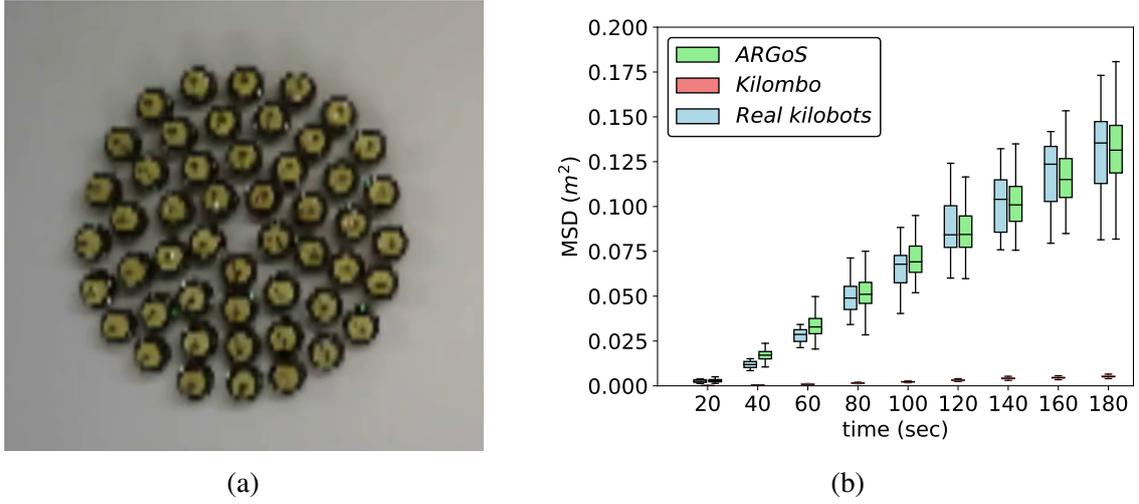


Fig. 3.8 ((a)) Initial distribution in four concentric circles with all 50 Kilobots facing towards the centre. ((b)) Comparison between real 50 Kilobots (19 runs) and 50 simulated Kilobots (100 runs) in ARGoS and Kilombo. We show the average mean square displacement (MSD) in a highly dense environment. ARGoS shows a good agreement with reality, whereas Kilombo does not. Video footage is available at <https://youtu.be/6HYti0ABuxc>.

concentric regular polygons and facing toward the centre of the polygons. Each polygon has twice the number of robots and double the radius of its internal polygon. The Kilobots then perform an isotropic random walk [36] in which they continuously alternate between forward motion for ~ 10 s and turn on the spot in a random direction (clockwise or anti-clockwise) for a random time drawn from a uniform distribution $\mathcal{U}(0, 4)$ s. We performed 20 replicates of this experiment with real robots and 100 runs with simulated Kilobots in both ARGoS and Kilombo. We performed ARGoS runs for different values of the friction parameter (from 0.1 to 2.0 with a step of 0.1). In each run, we recorded the trajectory of each Kilobot for 3 minutes and computed the mean square displacement (MSD) of the 50 Kilobots for each experiment as follows:

$$MSD(t) = \frac{1}{50} \sum_{k=1}^{50} [(x_k(t) - x_k(0))^2 + (y_k(t) - y_k(0))^2] \quad (3.2)$$

where $x_k(t)$ and $y_k(t)$ are the coordinates of the k^{th} robot at time t . For ARGoS, we selected the friction value, which minimises the least square error between the MSD of the real and the simulated experiment. Figure 3.8(b) shows the time evolution of the MSD of the Kilobots in reality (blue), in ARGoS (green), and in Kilombo (red). It is clear that ARGoS simulates physical interactions between robots more accurately than Kilombo.

3.4.3 Simulating ARK in ARGoS

ARK is integrated with ARGoS through the latter's plugin interface that implements the ARK Loop Function (ALF), which is the simulated counterpart of the ARK's base control station. The ALF is executed every ARGoS time-step and is in charge of simulating the virtual environments and orchestrating the simulated OHC entity which sends IR ARK messages to the simulated Kilobots. To facilitate the transfer from simulation to reality, the ALF uses the same structure and the same methods' names as its real counterpart. The virtual environments are set up at the beginning of the simulation via the method `SetupVirtualEnvironments()`, updated every time-step via the method `UpdateVirtualEnvironments()`, and graphically visualised via the method `PlotEnvironment()`. Similarly to the ARK's base control station, the ALF has real-time access to the state of the simulated Kilobots, *i.e.* their position, orientation, and LED colour. This information can be used by the user to code the functioning of the virtual actuators and sensors. The virtual actuators are 'actuated' by updating the virtual environments via the method `UpdateEnvironmentsThroughVirtualActuators()`. The virtual sensors' readings are computed using the Kilobot's state via the method `UpdateVirtualSensors()`, and are transmitted to the robot via the method `TransmitKilobotState(ARK_message)`. The ALF automatically codes the 3-byte ARK messages within standard 9-byte Kilobot messages in the same way ARK does. Therefore the Kilobot control software needs to decode the ARK messages in ARGoS in the same way it does in reality. In line with the ARGoS features, this implementation choice is particularly useful because it allows to work with the same identical code in simulation and on real robot. ALF gives to the user the possibility to limit the communication to a maximum frequency of 60 ARK messages per second (to match real ARK's frequency) or to simulate an unlimited ARK message frequency. Finally, custom parameters specific to the virtual environments, sensors, and actuators can be specified from the experiment configuration file.

To showcase the ALF functioning, we reproduced a simulated version of one experiment based on ARK, the Demo C of [160]. Figure 3.9 shows two screenshots of the experiment in simulation (left) and reality (right) featuring 50 Kilobots that operate in two virtual environments (flower field and nest).

3.5 Conclusion

In this chapter, we introduced the tools employed in this project to analyse models of swarm robotics systems. Section 3.1 described the Gillespie algorithm [64, 154] used to analyse chemical reaction models. Section 3.2 presented DeMaMAS [21, 20], the multi-agent simulator we employed to test collective decision-making agent-based models. In Section 3.3, we

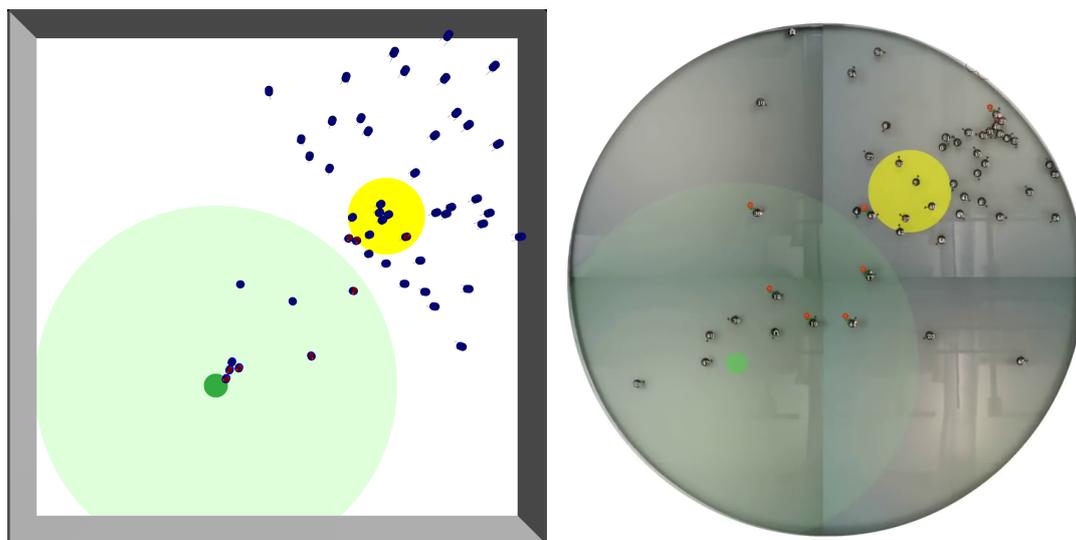


Fig. 3.9 Two screenshots of the same experiment in simulation (left) and reality (right). We (re)implemented the Demo C from [160] in which 50 Kilobots sense and modify two virtual environments. The full video is available at <https://youtu.be/kioZR99hnU4>.

introduced the Kilobot platform [174] used in this project to test the designed collective behaviours in the real world, the Augmented Reality for Kilobots (ARK) system [160] employed to extend the capabilities of the Kilobot robot, and the 3D-printed ring added to the Kilobot in order to improve ARK's tracking performance. In Section 3.4, we presented ARGoS simulator [139, 138] used to simulate the Kilobot experiments. This section also explained how, as part of this project, we contributed to the improvement of the ARGoS Kilobots simulator [138] by modelling and tuning the Kilobot's motion noise, calibrating the simulator's physics engine to better simulate physical interactions between Kilobots, and implementing a simulated ARK system.

Chapter 4

Improving accuracy in collective decision-making

The work presented in this thesis investigates individual behavioural rules that improve collective behaviours in swarm robotics. In this chapter¹, we propose individual behavioural rules that control the interactions between the robots to improve the collective decision-making of the swarm in terms of decision accuracy and speed. We consider the best-of- n decision problem in which the swarm is required to select the best option among several alternatives. Previous research revealed the existence of a dilemma on how to weight the individually-acquired information and social information [164]. When robots update their belief about the best option using mainly individually-acquired information, the swarm is not able to reach a consensus. While when the robots rely mainly on information acquired through interaction with other robots, the swarm frequently selects an inferior option. This situation usually manifests when the number of options is high, or when the difference between the quality of the options is tight. To solve the previous dilemma, we propose individual behavioural rules using which each robot vary the strength of its interactions with other robots over time. To quantify the performance improvement resulting from the proposed individual behavioural rules, we integrated them into the honeybees-inspired decision-making strategy [181, 132] and compared them with previous decision-making strategies. We performed our tests and comparisons using both stochastic analyses and swarm robotics simulations.

In this chapter, we first formalise the best-of- n decision problem in Section 4.1 and describe our experimental setup in 4.1.1. Next, in Section 4.2, we present the generic robot behaviour

¹This chapter is a modified version of a published manuscript: M. S. Talamali, J. A. R. Marshall, T. Bose and A. Reina, "Improving collective decision accuracy via time-varying cross-inhibition," *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 9652-9659. The "Abstract" and "Introduction" sections of the manuscript has been renamed and modified to maintain consistency with other chapters.

that we used to implement collective decision-making strategies. In Section 4.3, we introduce the direct comparison strategy, which is a simple collective decision-making strategy. Then in Section 4.4.1, we present the honeybee-inspired decision strategy used to integrate the proposed individual behavioural rules introduced in Section 4.4.3. In Section 4.4.2 we present the chemical reactions model of the honeybee-inspired decision strategy and the stochastic analyses we performed to confirm the existing dilemma regarding the strength of interaction and show the benefits of the proposed individual behavioural rules. In Section 4.5, we validate the benefits of these individual behavioural rules through swarm robotics simulations. Finally, we discuss the limitation and possible extensions of the proposed individual behavioural rules in Section 4.6.

4.1 Formalisation of the best-of- n decision problem

In the best-of- n decision problem studied in this project, a swarm of S robots is required to reach consensus on the best option among the n options available in the environment. Each of the available options has a unique ID i (with $i \in \{1, \dots, n\}$), a position in the space χ_i , and a quality $v_i \in [v_{min}, v_{max}]$ (v_{min}/v_{max} are the min/max quality that the robot can sense). To successfully solve the best-of- n decision problem, the swarm must select the option with the highest quality. This can be formalised as follows:

$$\arg \max_{\chi_i} v_i, \quad \text{with } i \in \{1, \dots, n\}. \quad (4.1)$$

The number of the available options n , their locations χ_i , and their qualities v_i (with $i \in \{1, \dots, n\}$), are a priori unknown to the robots. A robot is only able to acquire information about the options located within its field of view. Therefore, to find the available options, estimate their qualities, and decide on the best option, the robots are required to explore the environment.

In this study, we assumed that the robots could only make noisy estimates of the options qualities \hat{v}_i (with $i \in \{1, \dots, n\}$). We simulated the noisy estimate \hat{v}_i of the absolute quality v_i as drawing a random number from a normal distribution $\mathcal{N}(v_i, \sigma^2)$ with mean v_i and variance σ^2 . The variance σ^2 represents how noisy are the quality estimates made by the robots. When a robot draws an estimate outside its sensing range $[v_{min}, v_{max}]$, we reassigned its estimate to the value of the nearest bound. We also assumed that a robot has limited memory and can store the quality and the location of its preferred option (*i.e.* its commitment) only. Additionally, similarly to previous collective decision-making studies [152, 180, 135, 27], we considered

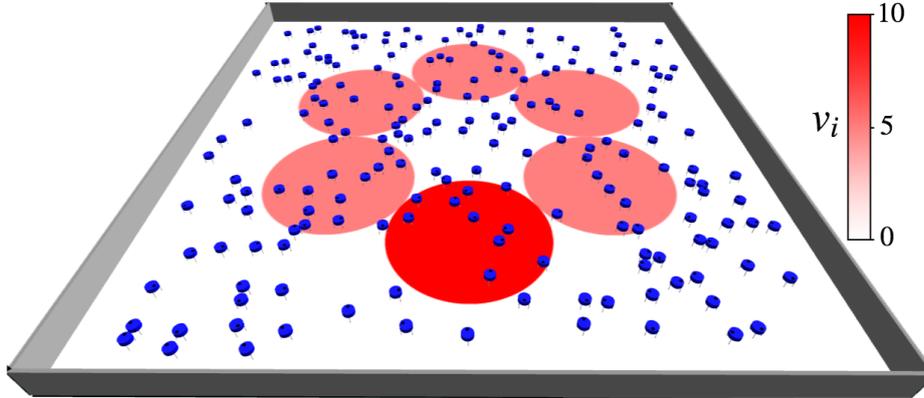


Fig. 4.1 Sample initial distribution of $S = 200$ simulated Kilobots in a scenario with $n = 6$ options and decision difficulty $\kappa = 0.5$. The red circles represent the areas (radius 25 cm) in which the options can be perceived by the robots via ARK. The colour intensity represents the option's quality $v_i \in [0, 10]$. The swarm is tasked to select the best option.

that consensus is reached for an option i when the number of robots committed to the option i reaches or exceeds a quorum threshold $Q = 80\%$ of the full population S .

4.1.1 Experimental setup

Almost all previous studies on collective decision-making investigated binary decision problems ($n = 2$); in this project, we focused on the best-of- n problem with $n \geq 2$ options. Moreover, we considered the case of one superior-quality option and $n - 1$ equal low-quality options. This experimental scenario has been used in many previous studies in different domains [179, 52, 170, 176, 164] as it allows to vary the difficulty of the best-of- n decision problem using a single parameter. In this study, this parameter is denoted by $\kappa = v_L/v_H \in [0, 1]$, where v_H is the quality of the superior option and v_L is the quality of the other ($n - 1$) inferior equal-quality options (with $v_H, v_L \in [v_{min}, v_{max}] = [0, 10]$).

In the study described in this chapter, we tested the designed collective decision-making algorithms in simulation using a swarm consisting of $S = 200$ Kilobots [174]. To simulate the Kilobots, we employed the ARGOS Kilobots simulator described in Section 3.4.1. We also employed the ARK system introduced in Sections 3.3.2 and 3.4.3 to allow the robots to perceive the options, compute their locations, and estimate their qualities. When an option i is within the robot's field of view (25 cm), the robot receives an ARK message containing the option's location χ_i and quality \hat{v}_i . We also employed ARK to inform the robots about their GPS location. A Kilobot can request its GPS information from ARK by lighting up its LED in red colour. ARK sends messages containing GPS information to all robots with a red LED. In

this study, the Kilobots use their GPS information to navigate toward an option to self-estimate its quality.

To avoid the spatial correlations that may be caused by the initial position of the robots and the positioning of the options, we uniformly distributed our $S = 200$ robots in a $2\text{ m} \times 2\text{ m}$ square environment as illustrated in Figure 4.1. We also distributed the n options on the vertices of a regular polygon with n edges and radius 50 cm. An example with $n = 6$ options is depicted in Figure 4.1. Moreover, we randomised the location of the best option in each simulation run.

4.2 Robot behaviour

In the study described in this chapter, we implemented the Kilobot behaviour following the guidelines of the design pattern for decentralised decision-making described in [167]. According to these guidelines, each robot contributes to the collective decision by performing three concurrent actions:

4.2.1 Environment exploration

Since the information about the available options (their number, locations, and qualities) are a priori unknown to the robots; the robots are required to explore the unknown environment in order to locate and estimate the quality of the available options. Searching for options in an unknown environment can be simply achieved by performing a diffusive isotropic random walk [36]. For this reason, in this study, we programmed the Kilobots to continuously perform a straight motion for approximately 10 s then rotate in a random direction for a random number of seconds drawn from a uniform distribution $U(0, 5)$ s. Besides allowing the discovery of the available options, the random walk allows the robots to exchange information with different peers and hence better estimate the overall state of the swarm. When encountering an option i , the Kilobot memorises the option's location χ_i and its estimated quality \hat{v}_i . The Kilobot then uses this information to update its commitment state.

In this study, a robot makes a single noisy quality estimate of the encountered options and does not average multiple measurements to compute a more accurate estimate. This is because we are not interested in individual strategies to reduce noise; instead, we are interested in collective strategies that allow a swarm of noisy robots to effectively aggregate noisy measurements. We assume that in real-world applications, performing individual strategies to attenuate noise would improve the accuracy of the estimates, but does not eliminate noise. Therefore, we modelled the robots' noisy estimates through sampling from a normal distribution $\mathcal{N}(v_i, \sigma^2)$.

4.2.2 Social interactions

Whilst searching for the available options in the environment through a random walk, each robot communicates with other robots within its local communication range of about 10 cm. Every second, each robot transmits a message to share its commitment state with others. When committed to an option, the robot shares the option's location with others, and in some decision-making strategies, such as the one described in section 4.3, the robots also share their estimate of the option's quality. Robots use the options' information received from others to update their commitment state.

4.2.3 Commitment updates

While solving the best-of- n decision problem, each robot in the swarm can be either in the uncommitted state U or committed to an option (*i.e.* in the committed state C). The robots have no prior knowledge about the decision problems; hence all the robots are initially uncommitted. A robot committed to option i stores its location χ_i and its estimated quality \hat{v}_i . When a Kilobot acquires information about an option, either through physically encountering the option or by being informed by another robot, the Kilobot updates its commitment state accordingly. The Kilobot can change from an uncommitted state to become committed to an option or can revert its commitment from one option to another. In this study, we implemented multiple strategies for updating the robot's commitment and compared their performance in solving best-of- n decision problems. The implemented strategies are introduced in Sections 4.3 and 4.4.

4.3 Direct comparison strategy

If one thinks about collectively solving the best-of- n decision problem, the first strategy that may come to mind is to compare the options' qualities and every time select the one with the highest quality. This strategy is known in the literature as the *direct comparison strategy* (DC) [208]. In the DC strategy, the robots are required to exchange their estimates of the option's quality \hat{v}_i ; then each robot updates its commitment state accordingly. When an uncommitted robot gets access to information about an option i , either through a message from a peer or by encountering the option while exploring, the robot commits to the option i and stores its location χ_i and its estimated quality \hat{v}_i . When a robot that is committed to option i receives information about a different option $j \neq i$ from a peer, and in case the options j has a better quality than the option i (*i.e.* $\hat{v}_j > \hat{v}_i$), the robot changes its commitment to option j and stores the received information χ_j and \hat{v}_j . In case the two options i and j are equally good, *i.e.* $\hat{v}_i = \hat{v}_j$,

the robot chooses one of them at random. Making this random choice is expected to allow the swarm to reach consensus in case of equal-quality options.

As discussed in Section 4.2, the robots make only one noisy estimate of the option's quality that they disseminate within the swarm. We consider this estimate to be the most accurate measurement they can make even when performing individual strategies to attenuate noise. The DC strategy allows information to spread at high speed; hence consensus is reached very quickly. However, the DC strategy also allows errors to spread at the same high speed. It is enough that a single robot overestimates the quality of an option for this one to be selected by the swarm because robots rely on *second-hand* quality estimates made by other robots to update and advertise their opinion.

To assess the effect of noise on the DC strategy, we implemented it on a swarm of $S = 200$ Kilobots using the ARGoS Kilobots simulator (presented in Section 3.4.1). We performed our tests in case of $n = 6$ options, and for multiple problem difficulties $\kappa = v_L/v_H \in [0.5, 1.0]$ and noise magnitudes $\sigma^2 \in [0, 5]$. As depicted by Figure 4.2, the results of these tests showed that the accuracy of the DC strategy quickly drops as the quality estimation noise magnitude or the problem difficulty increases. Additionally, the results have shown that although the robots select a random option in case of equal-quality, the swarm is not always able to break the decision deadlock in this case of equal-quality options.

4.4 Collective Decisions through Cross-inhibition

In the study described in this chapter, we extended the Collective Decisions through Cross-Inhibition (CDCI) strategy proposed in [167] to improve its performance and overcome the poor performance of the direct comparison strategy described in Section 4.3. The CDCI strategy has been inspired by the strategy employed by European honeybees [181] to decide on their future nest-site from the possible nesting locations available in their environment. The CDCI strategy has been employed in multiple swarm robotics studies [165, 159]. In our extensions of the CDCI strategy, we introduced time-varying interactions between the robots and removed the requirement of sharing the quality estimates of the options between the robots. Through these additions, we were able to improve the performance of the CDCI strategy in both decision speed and accuracy. These improvements are shown in later sections through multiple comparisons.

4.4.1 The basic CDCI strategy

In the CDCI strategy, each robot updates its commitment state using the probabilistic finite state machine (PFSM) depicted by Figure 4.3. The active state of the PFSM depends on the

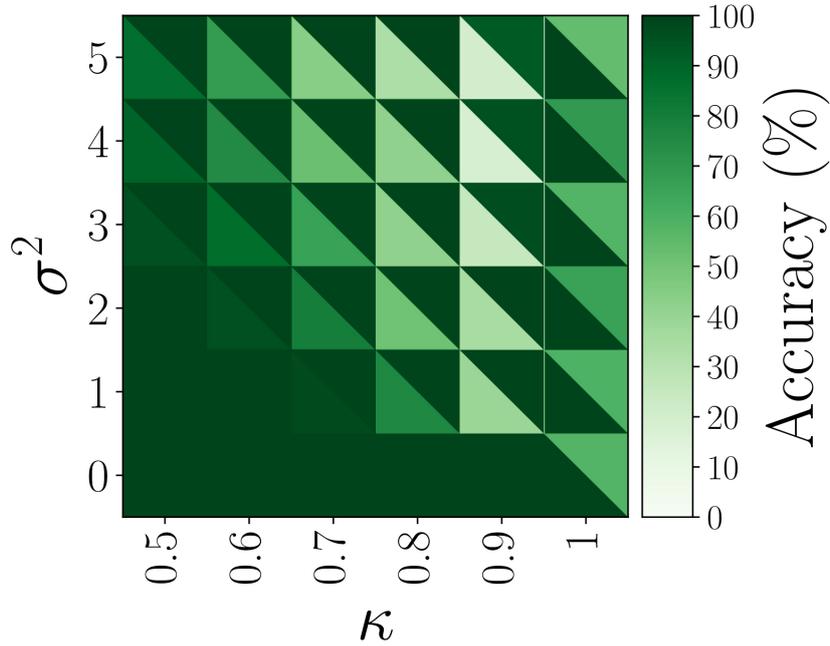


Fig. 4.2 The results of 200-Kilobot swarm (100 simulations in each condition) showing the effect of noise strength $\sigma^2 \in [0,5]$ on the decision accuracy in the best-of-6 problem with difficulty $\kappa = v_L/v_H \in [0.5, 1]$ ($v_H = 10$). We compare the accuracy of the DC strategy of Section 4.3 (bottom-left triangles) with the accuracy of the time-varying strategy $r_{step}(t)$ (with $\tau_0 = 50$) of Section 4.4.3 (top-right triangles). While DC is highly sensitive to noise, the proposed strategy shows remarkably high performance ($\geq 93\%$) for any tested noise level and difficulty κ up to 0.9. In case of equal-quality options ($\kappa = 1$), the quick dynamics of DC allows to break the symmetry within 2 hours more often than the proposed strategy with a suboptimal parameterisation of τ_0 (see more details in Figure 4.9).

commitment state of the robot. When the robot is committed to option i , with $i \in \{1, \dots, n\}$, the state C_i is active, while when the robot is uncommitted, the state U is active. Following the CDCI design pattern [167], the robots update their commitment based on the information they gather about the commitments distribution of the other agents in their local neighbourhood. In the study described in this chapter, the robots gather information for $\delta_u = 50$ clock cycles (*i.e.* ~ 1.5 s) before updating their commitment as follows. When an uncommitted robot encounters the option i in the last δ_u clock cycles (*i.e.* the condition E_i is satisfied), the robot commits to option i with a probability P_{γ_i} . In this case, we say the robot *discovered* option i . Robots committed to option i may spontaneously *abandon* their commitment to option i and revert to the uncommitted state with probability P_{α_i} . When an uncommitted robot receives a message from a robot committed to option i in the last δ_u clock cycles (*i.e.* the condition M_i is satisfied), the robot commits to option i . In this case, we say that the robot *was recruited*. When a robot

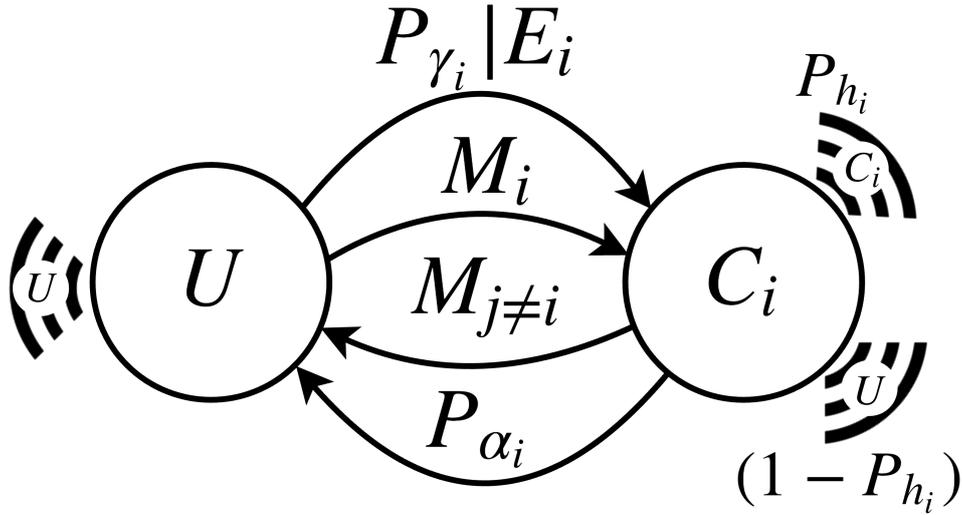


Fig. 4.3 The PFSM controlling the robot's decision state. Robots update their commitment state using the probabilities P_{γ_i} and P_{α_i} of Equation (4.2) or upon receiving a message M_i from a robot committed to i . The symbol $|$ on the arrow for the discovery transition indicates conditional probability on the occurrence of the event E_i of encountering option i . The transmission symbols indicate that a robot in state C_i sends an interaction message (for recruitment and cross-inhibition) with probability P_{h_i} .

committed to option i receives a message from another robot committed to a different option j in the last δ_u clock cycles (*i.e.* the condition $M_{j \neq i}$ is satisfied), the robot becomes uncommitted. In this case, we say that the robot *was cross-inhibited*. At each broadcast tick, robots broadcast messages to share information about their commitment state with the other robots in their surrounding. A robot committed to option i probabilistically decides about the nature of the information to share with others at each broadcast tick. The robot interacts with its peers and shares information about its option i (*i.e.* the option's location χ_i) with a probability χ_i , or decides to not interact with them and appears as uncommitted with a probability $1 - P_{h_i}$.

Following the guidance given by the CDCI design pattern [167], each robot performs interaction-based transitions (*i.e.* recruitment and cross-inhibition) probabilistically as a function of the commitments of its neighbouring robots. These probabilistic transitions can be achieved through the selection of one random message M among all the messages a robot receives during each update cycle δ_u when performing the interaction-based transitions (see Figure 4.3 and [167] for more details). In this study, the robot stores the received messages at the same memory location and uses the last saved information to update its commitment. This implementation allows the robot to use the most up-to-date information to update its commitment. Additionally, this implementation requires minimal memory and hence makes the CDCI decision strategy usable on minimalistic robotics platforms.

In contrast to the DC strategy of section 4.3 and the previous implementation of the CDCI strategy [159], in the CDCI implementation of this study, robots do not share their quality estimates of the options with other robots. Instead, after getting recruited to an option, the robots navigate toward the option's location to self-estimate the option's quality. This behaviour is similar to what honeybees [18] and ants [99] do when collectively selecting their future nest-site. While moving toward an option to self-estimate its quality, robots suspend their interaction with other robots. Despite being a time-consuming action, making individual estimates of the options qualities allows mitigating against the spreading of individual misestimations. Our analysis of the DC strategy revealed that the use of second-hand information (*i.e.* the shared quality) spreads individual misestimations and lead to inaccurate collective decisions. To evaluate the effect of self-estimations on the performance of the DC strategy, we modified the DC strategy by asking each recruited robot to self-estimate the quality of the option. This modification increased the performance of the DC strategy to a similar level as the CDCI strategy for $\kappa \leq 0.9$. However, the DC strategy was still unable to break symmetry (*i.e.* $\kappa \approx 1$) even for long-running times. The DC strategy has shown quicker decisions compared to CDCI. It is because the DC response is independent of the options' quality, meaning that the swarm will take the same time to decide between low-quality options or high-quality options. This behaviour may lead the swarm to select a low-quality option before fully exploring the environment. In contrast, using the CDCI strategy, the swarm will take a longer time to decide between low-quality options in the hope to find something better in the environment.

Requiring individual estimates may lead numerous robots to move towards a popular option for resampling its quality. To avoid the overcrowding and interference issues that may result in this scenario, every robot, after resampling the quality of an option, goes away from the option to a random location at least 50 cm distant before resuming a random walk. Figure 4.4 shows the FSM of the robots' movement.

In the CDCI strategy, each robot performs individual behaviours probabilistically as a function of its option's quality. The higher is the option's quality, the more likely the robot commits to it. This behaviour allows the swarm as a whole to reach consensus for the best

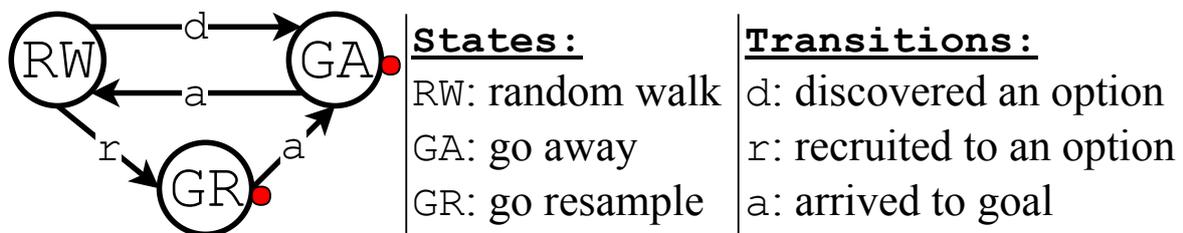


Fig. 4.4 The FSM controlling the robot's movements in the CDCI strategy. Red dots indicate that the robot accesses its GPS location (through ARK).

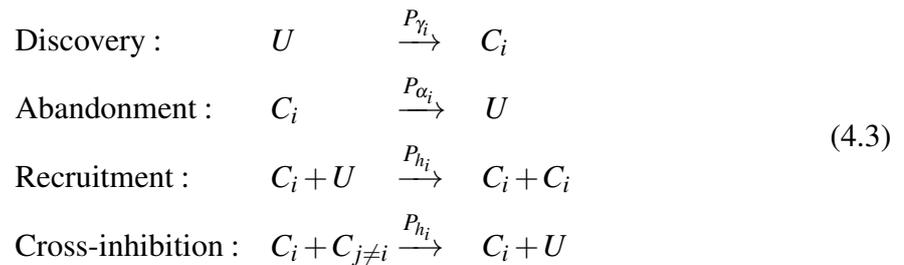
option. In the study described in this chapter, we employ the parameters proposed in [164] for the probabilities of the robots individual behaviours. This parameterisation allows the swarm to conserve the value-sensitive properties of the CDCI strategy even when the number of options is higher than two:

$$P_{\gamma_i} = k\hat{v}_i\Delta, \quad P_{\alpha_i} = k\hat{v}_i^{-1}\Delta, \quad P_{h_i} = h\hat{v}_i\Delta, \quad i \in \{1, 2, \dots, n\} \quad (4.2)$$

where \hat{v}_i is the estimated quality of option i , while h and k are parameters to control the frequency at which the robots send interaction messages and perform individual behaviours, respectively. The ratio $r = h/k$ represents the relative interaction rate. Following [167], the parameter Δ is required to scale probabilities within the valid range $[0, 1]$ and guarantee a match between the microscopic and the macroscopic description of the process. $\Delta = \delta_u\delta_c\delta_s$ is determined by three components: the number of Kilobot clock cycles between two updates ($\delta_u = 50$), the Kilobot clock period ($\delta_c \simeq 31$ ms), and the temporal scaling factor $\delta_s = 0.000594$ which controls the speed of the decision process. As shown in [164], the critical parameter in the swarm decision dynamics is the relative interaction rate r .

4.4.2 Stochastic analysis of the basic CDCI strategy

To identify which values of the relative interaction rate r allows the swarm to make better decisions, we analysed the effect of the relative interaction rate r on the decision outcome of the CDCI strategy using the stochastic simulation algorithm [64] introduced in section 3.1. The stochastic simulation algorithm allows approximating the solution of the master equation describing the macroscopic dynamics of the CDCI strategy [181, 167]. Approximating the solution of the master equation takes into account the random fluctuations caused by the finite size of the swarm $S = 200$ on the macroscopic dynamics of the system. In the study described in this chapter, we performed 1,000 runs of the stochastic simulation algorithm (SSA) [64] to obtain a reliable approximation of the solution of the master equation solution. To employ the SSA, we represented the CDCI model of section 4.4.1 using the following chemical reactions model:



where U and C_i stand for uncommitted and committed to option i respectively. While $P_k, k = \{\gamma_i, \alpha_i, h_i\}$ are the probabilities of Equation (4.2).

Using the SSA, we analysed the collective decision given by the CDCI strategy for values of $r \in [1, 100]$ with $k = 1$. At first, we fixed the number of options $n = 6$ and varied the difficulty of the decision problem $\kappa \in [0.5, 1]$ (Figure 4.5(a)). Then we fixed the difficulty of the decision problem $\kappa = 0.9$ and varied the number of options $n \in [2, 12]$ (Figure 4.5(b)). Each run of the SSA is stopped if the simulation reaches a maximum decision time $T_{max} = 10$ or when the swarm reaches consensus for one of the options (*i.e.* the number of robots committed to the same option reaches of the quorum threshold $Q = 80\%$). The selected maximum decision time $T_{max} = 10$ is far above the average decision time; to ensure that the swarm is given enough time to reach consensus. Figure 4.5 depicts the results of our analysis. Each tested condition is represented by a coloured pie-chart. The yellow colour in the pie-charts indicates the proportion of runs that terminated in a decision deadlock (*i.e.* the number of robots committed to any of the options was smaller than quorum threshold $Q=80$). The green colour indicates the proportion of runs where the swarm selected the best option. The red colour indicates the proportion of runs where the swarm selected one of the $(n - 1)$ inferior equal-quality options. Our analysis revealed that breaking decision deadlock requires high positive and negative feedback (*i.e.* high relative interaction rate r). This result is in accordance with the findings of previous deterministic mean-field analyses [164, 68]. Additionally, our analysis, similarly to stochastic analyses of decision-making models of ants and slime moulds [126], has shown that high positive feedback decreases decision accuracy.

The results shown in Figure 4.5(a) reveals the presence of a dilemma: On the one hand, low values of r allow the swarm to accurately select the best option when its quality is much higher than the other options' quality ($v_H \gg v_L$) but lead to a decision deadlock when the qualities of the available options are similar ($v_H \approx v_L$). On the other hand, high values of r allow the swarm to break the decision deadlock but may lead to wrong decisions. It is because using high values of r increases the speed at which opinions spread and hence first discovered options have a higher chance to be selected even when having inferior quality. Moreover, as shown in figure 4.5(b), the minimum r necessary to break deadlock increases quadratically with the number of options. Furthermore, the interaction rate r that gives a maximum accuracy depends on the decision problem (*i.e.* the number and the quality options), which is usually not known to the swarm in advance. Therefore, fixing the value of r makes the CDCI strategy limited as it may lead to poor performance in some decision problems.

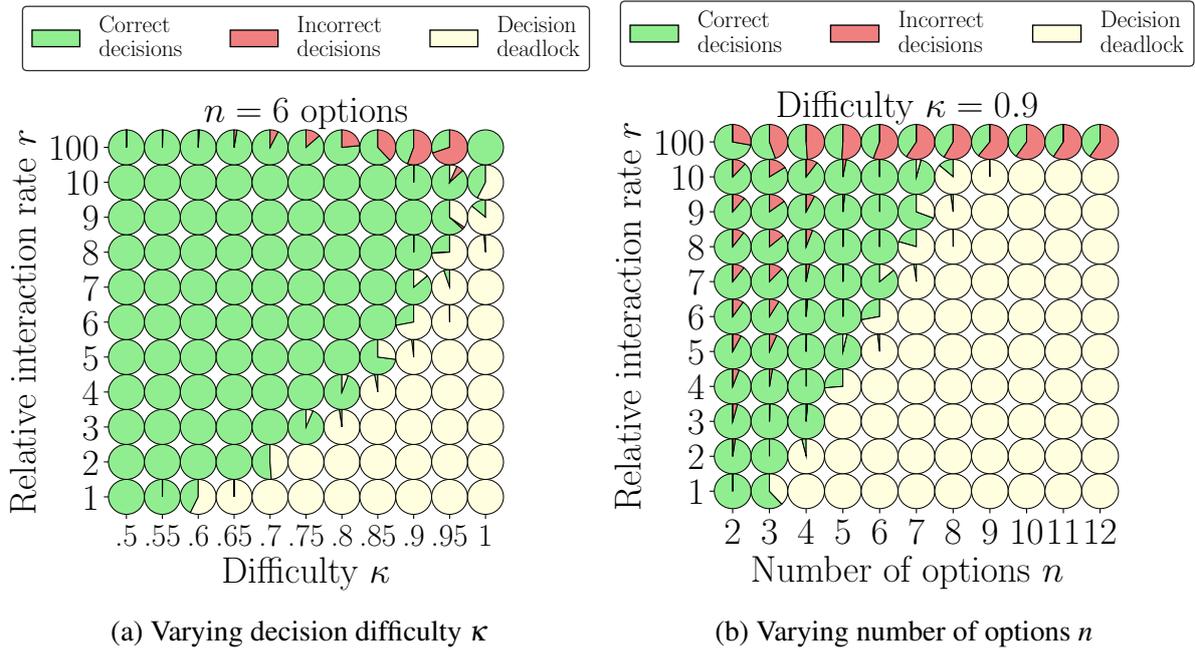


Fig. 4.5 Results of the SSA showing the influence of the interaction ratio $r = h/k$ with $k = 1$ (of Equation (4.2)) for various best-of- n problems in the case of $S = 200$ robots. The pie-charts indicate the percentage of 1,000 runs terminating in a decision deadlock, *i.e.* below quorum $Q = 0.8$ after $T_{max} = 10$ (yellow), a decision for the best option $v_H = 10$ (green), or a decision for any $n - 1$ inferior-quality distractor $v_L = \kappa \cdot v_H$ (red). Panel (a) Sensitivity of CDCI to the ratio r for various problem difficulties $\kappa \in [0.5, 1]$ in case of $n = 6$; the minimum r necessary to break decision deadlock grows quadratically with κ . Panel (b) Sensitivity of CDCI to the ratio r for various number of options $n \in [2, 12]$ in case of $\kappa = 0.9$; the minimum r necessary to break decision deadlock grows quadratically with n . Sufficiently high values of interaction rate r always lead to a decision, but accuracy rapidly decreases with increasing n or κ .

4.4.3 The time-varying CDCI strategy

In the study described in this chapter, we proposed a novel decentralised strategy to solve the previous dilemma without the need to know the decision problem (*i.e.* n and κ) in advance. Our strategy consists of starting with a low interaction rate r to limit the effect of initial random fluctuations and then increasing the interaction rate to reach consensus. Starting with a low r restricts interactions to stop the quick spreading of first discovered options which may be of inferior quality v_L . Increasing the interactions over time allows the swarm to break the decision deadlock and reach consensus for the best discovered option. Moreover, in our strategy, the speed at which the robots increase/start their interactions depend on the estimated quality of their options; *i.e.* the higher is the quality of a robot's option, the higher/earlier the robot recruits and cross-inhibits other robots. This quality-sensitive increase of interactions is expected to

allow better options to spread quicker, and hence improve the accuracy and speed of the swarm's decisions.

In our study, we investigated two possible implementations for our proposed time-varying strategy. In the first implementation, the robots gradually increase their interactions rate r following the ramp function $r_{ramp}(t)$ shown in Figure 4.6(a). In the second implementation, the robots abruptly vary their interaction rate r from zero to high values following the step function $r_{step}(t)$ shown in Figure 4.6(b). The two functions $r_{ramp}(t)$ and $r_{step}(t)$ are mathematically described as follows:

$$r_{ramp}(t) = \frac{h_{ramp}(t)}{k}, \quad h_{ramp}(t) = \begin{cases} \frac{H_{max}}{\tau(\hat{v}_i)} t & \text{if } t < \tau(\hat{v}_i) \\ H_{max} & \text{if } t \geq \tau(\hat{v}_i) \end{cases}, \quad (4.4)$$

$$r_{step}(t) = \frac{h_{step}(t)}{k}, \quad h_{step}(t) = \begin{cases} 0 & \text{if } t < \tau(\hat{v}_i) \\ H_{max} & \text{if } t \geq \tau(\hat{v}_i) \end{cases}, \quad (4.5)$$

The two functions described by equations (4.4)-(4.5) keep the individual transitions strength k constant ($k = 1$) all the time and increase the interaction strength $h(t)$ over time from 0 to a maximum value H_{max} . The time at which the interaction strength $h(t)$ reaches the maximum value H_{max} depends on the robot's quality-estimate \hat{v}_i of an option i and is expressed by the function $\tau(\hat{v}_i) = \tau_0 v_{max} / \hat{v}_i$. The time $\tau(\hat{v}_i)$ defines the slope of the ramp function $r_{ramp}(t)$ and the jump time of the step function $r_{step}(t)$. For instance, if the robot's option quality estimate is equal to the maximum value ($\hat{v}_i = v_{max}$), the interaction strength $h(t)$ reaches the maximum value H_{max} at τ_0 . In case the option's quality estimate is smaller than the maximum value ($\hat{v}_i < v_{max}$), the maximum interaction strength is reached later. A decentralised implementation of the functions of Equations (4.4)-(4.5) can be achieved by programming the robots to increase their interaction strength over time. This results in a higher probability of sending recruitment and cross-inhibition messages given by equation (4.2) with the time-varying term h from Equations (4.4)-(4.5).

To assess the performance of our time-varying strategy, we implemented its two variants using the chemical reactions model of Equation (4.3) and approximated the solution of their corresponding master equation using the stochastic simulations algorithm (SSA) [64]. In the systems representing our time-varying strategy, the transition rates vary over time; hence it was not possible to employ the original SSA as it only deals with systems of constant transition rates. Therefore, we employed an extended version of the SSA proposed in [154], which considers time-varying transition rates. We compared the decision outcome of the proposed time-varying interaction strategies and constant values of the interaction rate r for different

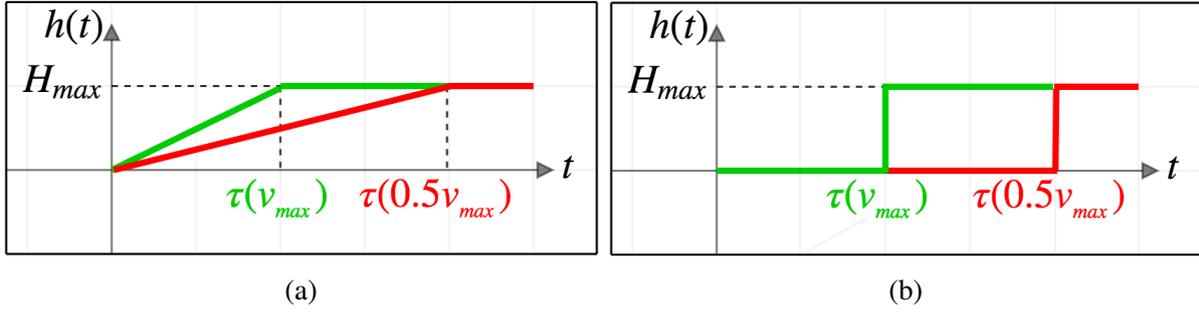


Fig. 4.6 Two forms for the time-varying interaction $h(t)$: $h_{ramp}(t)$ of Equation (4.4) in panel (a) and $h_{step}(t)$ of Equation (4.5) in panel (b). With the ramp function, the robot constantly increases the interaction strength $h_{ramp}(t)$ with a slope proportional to the estimated quality \hat{v}_i ; instead, with the step function, the robot does not interact $h_{step}(t) = 0$ until a time $\tau(\hat{v}_i)$ that is inversely proportional to the estimated quality \hat{v}_i .

decision problems of $n \in \{3, 6, 9, 12\}$ and $\kappa = 0.9$. The results of our comparison are reported in Figure 4.7 following the same colour code of Figure 4.5. In each experimental condition, the results were obtained through 1,000 SSA simulations. For a low value of the relative interaction rate $r = 1$ the individual behaviours (*i.e.* discovery and abandonment) and the interaction behaviours (*i.e.* recruitment and cross-inhibition) happen with a similar frequency and does not allow the swarm to break the decision deadlock for more than three options of almost similar quality ($\kappa = 0.9$). In case of high values of the relative interaction rate $r = 100$ where the individual behaviours are 100 times less frequent than the interaction behaviours, the swarm always breaks the decision deadlock but makes a wrong decision more than half of the time. It is because the initially randomly discovered options, which may be of lower quality, quickly spread within the swarm. Thus the probability of a lower quality option to be selected is high and increases with n . Both variants of our time-varying interaction rate $r(t)$ improve the accuracy of the swarm, especially the $r_{step}(t)$ which leads to 100% of accurate decisions.

4.5 Robot swarm simulations

To validate our proposed strategies, we re-conducted the comparison of Section 4.4.2 through implementation on a simulated robot swarm of $S = 200$ Kilobots². We compared the different strategies in case of various decision problems with $n \in \{3, 6\}$ options, a difficulty $\kappa = 0.9$ ($v_H = 10$ and $v_L = 9$), and a quality estimation noise strength $\sigma^2 = 1$. Each experimental condition was tested through 100 simulation runs. For this, we employed the widely-used

²The robot control software is available online at <https://github.com/DiODEProject/Time-Varying-CDCI>

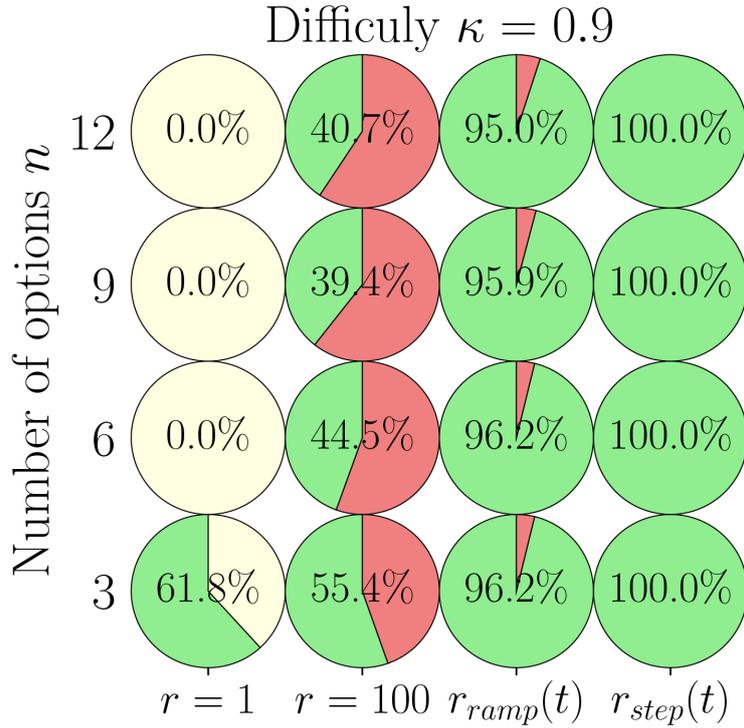


Fig. 4.7 Results of the SSA showing the effect of the time-variant behaviours on the decision outcome of a 200-agents swarm for various best-of- n options problems $n \in \{3, 6, 9, 12\}$ with difficulty $\kappa = 0.9$. Results of each condition are computed over 1,000 runs and represented with a pie-chart that indicates the percentage of runs terminating in a decision deadlock (yellow), a correct decision (green), or an incorrect decision (red). In each run, the swarm makes a correct/incorrect decision when it reaches the quorum threshold $Q = 80\%$ within $T_{max} = 10$ for the best option ($V_H = 10$)/lower-quality option ($V_L = \kappa \cdot V_H$). Otherwise the swarm is stuck in decision deadlock. The $r_{ramp}(t)$ and $r_{step}(t)$ from Eqs. (4.4)-(4.5) with parameters $\tau_0 = 5$ and $H_{max} = 100$ show a considerable improvement in accuracy (accuracy rate reported at the centre of each pie-chart)

physics-based simulator ARGoS [139, 138] that we previously introduced in Sections 3.4.1 and 3.4. We set a time limit of $T_{max} = 2$ hours for the swarm to decide for one option (*i.e.* at least $Q = 80\%$ of the swarm select the same option). Although we used the time limit T_{max} to assess the swarm's decision outcome, we run the simulations for 5 hours to better appreciate the decision speed of the tested strategies. The results of our comparison are reported in Figure 4.8, which shows the accuracy and the speed of the swarm's decision. In each tested decision problem and for different strategies. The results obtained via physics-based swarm robotics simulations are qualitatively similar to those obtained through the stochastic analysis of Section 4.4.2. As expected, using a low interaction rate $r = 1$, which results in slow dynamics, the swarm remains undecided and is unable to break the decision deadlock before within the

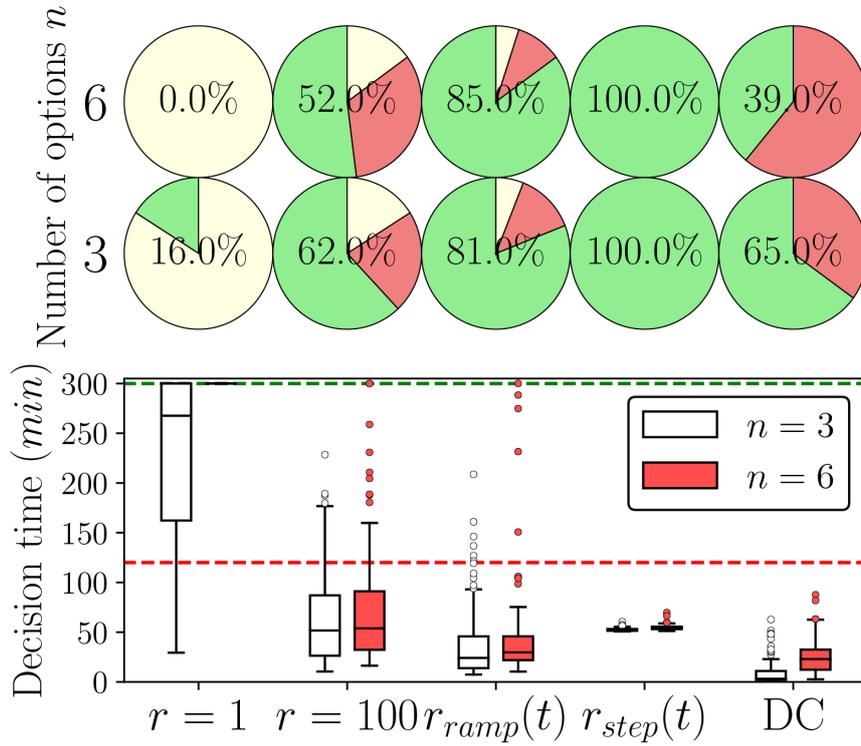


Fig. 4.8 200-Kilobot swarm results (100 simulations for each condition) for different decision strategies (in each column) in case of $n \in \{3, 6\}$ options with difficulty $\kappa = 0.9$ and noise strength $\sigma^2 = 1$. Top pie-charts show the decision accuracy (same colour code of Figs. 4.5-4.7). Bottom boxplots show the decision time; the horizontal red line (at 2 hours) is the cutoff time to compute the decision outcome (e.g. indecision vs decision). We let the simulation run a maximum of 5 hours to display the complete decision time dynamics. Low interaction rate ($r = 1$) shows low convergence rate and frequent deadlocks. High interaction rate ($r = 100$) shows low accuracy. Time-varying $r_{ramp}(t)$ shows an improvement in accuracy, which is further improved by $r_{step}(t)$ (both time-varying strategies use $\tau_0 = 50$ min). The DC (of Section 4.3) shows low accuracy due to the spreading of noisy estimates.

time limit. On the other hand, a high interaction rate $r = 100$ accelerates the dynamics but leads to wrong decisions. Using the first version of our proposed time-varying strategy $r_{ramp}(t)$ (with $\tau_0 = 50$ min), the swarm makes on average more accurate and faster decisions. The second version of our time-varying strategy $r_{step}(t)$ (with $\tau_0 = 50$ min) improves the decision accuracy of the swarm further. Intriguingly, the $r_{step}(t)$ strategy has also a stable and predictable decision time of about a few minutes after τ_0 . The DC strategy of Section 4.3 has the quickest decision times, but the lowest accuracy as individual misestimations quickly spread within the swarm leading to wrong decisions. The stochastic analysis of Section 4.4.2 gave a good prediction of the expected swarm behaviours, although not having identical dynamics as the swarm system.

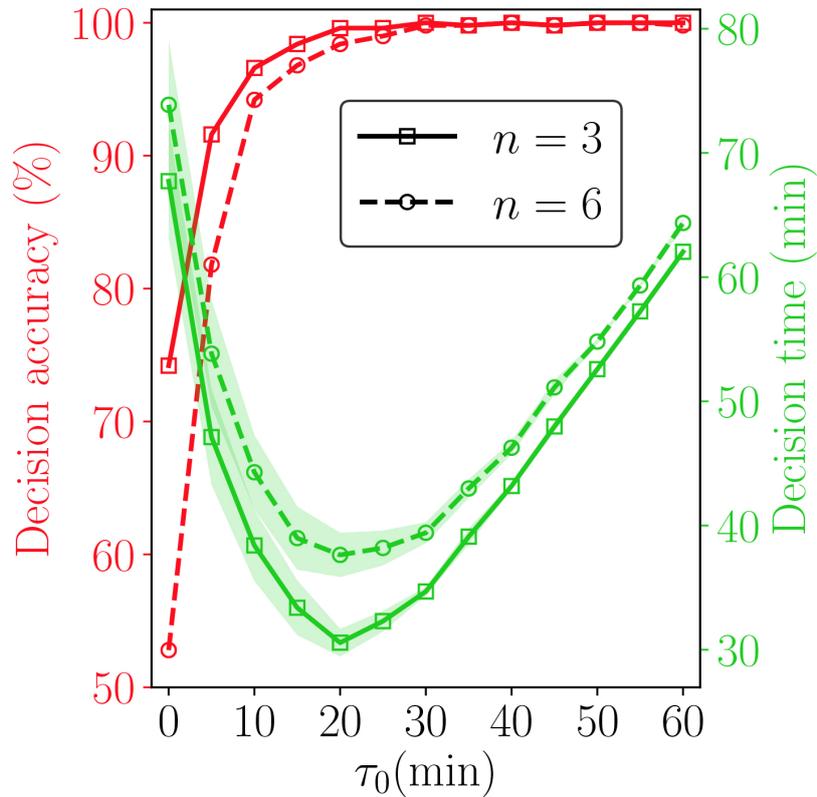


Fig. 4.9 Speed (green lines with 95% confidence shades and right y-axis) and accuracy (red lines and left y-axis) of the swarm robotics system for varying interaction speed $\tau_0 \in [0, 60]$ min for the $r_{step}(t)$ strategy. An inaccurate tuning of τ_0 may lead to sub-optimal performance.

The main differences between the swarm robotics system and the master equation model are the local communication between the agents and local encountering of the options. Robots perform discovery transitions only when they encounter the options which are localised in space. Moreover, the robots exchange information only with other robots within their communication range. Since the robots have slow motion, the local communication causes the robots to make correlated interactions (*i.e.* a robot can speak with the same robots for several times steps). In contrast, in the chemical reactions system analysed using the SSA, there is no notion of space and locality, each agent can unconditionally interact with any other agent and discover any of the options. For this reason, the interaction between the robots and between the robots and the environment are completely uncorrelated.

4.6 Discussion

In the study described in this chapter, we proposed behavioural rules through which the individual robots control their interactions with others to improve the performance of the swarm in making collective decisions. We tested these behavioural rules for solving the best-of- n decision problem in which the robots' task is to reach a consensus on the best option among several available alternatives. We focused our tests on best-of- n problems with a high number of options ($n > 2$) of nearly similar qualities as this type of decision problems are very challenging to solve [7, 164]. Our proposed individual behavioural rules consist of increasing the strength of interactions between the robots over time. This increase of interactions over time mitigates against the quick spreading of the first randomly discovered low-quality options that may be caused by strong interactions. Moreover, in the proposed individual behavioural rules, the speed at which robots increase the strength of their interactions depends on the quality of their options. The higher is the quality of a robot's option, the faster the robot increases its interaction strength leading to a quicker spreading of better options. To test the effect of the proposed individual behavioural rules on the performance of the swarm in the best-of- n problem, we integrated the behavioural rules into the Collective Decision through Cross-Inhibition decision strategy (CDCI) [181, 159]. Through both stochastic analysis and physics-based swarm robotics simulations, we demonstrated that our behavioural rules considerably improves the accuracy of the swarm in solving the best-of- n problem.

Most of the previous research on collective decision-making algorithms in swarm robotics studied binary decision problems ($n = 2$) [209]. Only very few works considered scenarios of more than two options. e.g. [133, 56]. However, a previous theoretical analysis showed that increasing the number of options can considerably change the swarm dynamics [164]. We, therefore, performed our analyses and experiments in a genuinely best-of- n setup. Our proposed individual behavioural rules demonstrated highly accurate collective decisions with an anticipated decision time for all the tested number of options.

Our best performing control rule divides the process of making a collective decision into two phases: an exploration phase and an exploitation phase. In the exploration phase, the robots individually explore the environment without interacting to allow the swarm to accumulate knowledge about the available options. In the exploitation phase, the robots interact with each other with a quality-proportional interaction strength to exchange their opinions about which option is the best and reach consensus. Modulating the strength of individuals' interaction as a function of environmental features have been previously demonstrated to improve collective behaviours such as collective motion systems [201, 183, 102] and collective foraging systems [131, 144]. While our step-based behavioural rule shows excellent decision performance, it is important to highlight that the best time to switch from the exploration phase

to the exploitation depends on the speed of the decision process which is a function of the decision problem. As depicted by Figure 4.9, the accuracy and the speed of the collective decision when using our two-phase behavioural rule $r_{step}(t)$ depend on the value of the time τ_0 . Figure 4.9 also shows that inaccurately tuning the minimum time for starting interaction τ_0 may diminish the swarm's performance. If the swarm switches to the exploitation phase too early, before any option is discovered, the benefit of the two-phases behavioural rule disappears. Conversely, if the switching happens too late, the swarm unnecessarily delays consensus and reduces the decision speed. Preliminary results indicated that the best value of τ_0 depends on environmental and robots' parameters including the environment size, the difficulty of the decision problem κ , the strength of the quality-estimation noise σ^2 , and the robots' field of view. These parameters are generally not known to the designer in advance, and hence the accurate tuning of the swarm system in advance is impossible. Therefore, to make our step-based behavioural rule adaptive to different decision problems, we envision to work on a decentralised strategy to allow individual robots to estimate the best moment to activate the interaction. Robots may be able to estimate the best switching time through some form of environmental sampling.

Chapter 5

Achieving adaptation in collective decision-making

In the previous chapter, we proposed individual behavioural rules that improve the collective decision-making of robot swarms in terms of accuracy and speed. In this chapter, we are interested in strategies that allow robot swarms to adapt their collective decision in case of environmental changes. Similarly to the previous chapter, we consider the best-of- n decision problem in which the swarm is required to reach a consensus on the best option among several options available in the environment. While in the previous chapter, the environment (*i.e.* the number and the qualities of the available options) was static throughout the experiment, in this chapter, we consider scenarios where the environment changes over time, requiring the swarm to adjust its decision accordingly. In this study, we propose individual behavioural rules that give robot swarms the ability to change their decision on the best available option in response to the current state of the environment. We test these behavioural rules using both multi-agent and swarm robotics simulations. Moreover, we analyse the performance of these behavioural rules for different values of the robot density in the environment and the robot's communication range. In section 5.1, we formalise the dynamic best-of- n options decision problem. In section 5.2, we present the types of environmental change considered in our study. Section 5.3 introduces the overall robot's behaviour. Section 5.4 introduces the decision-making model we employ and the individual behavioural rules we proposed in this study to achieve adaptation. Section 5.5 presents the experiments we conducted to analyse the studied individual behavioural rules. Finally, the results of our analysis are presented and discussed in sections 5.6 and 5.7, respectively.

5.1 The best-of- n problem in dynamic environments

Similarly to the previous chapter, the study described in this chapter addresses the best-of- n decision problem in which the swarm is required to reach a consensus on the best option (*i.e.* with the highest quality) out of n options available in the environment. In the previous chapter, the environment, *i.e.* the number and the qualities of the available options, remained fixed throughout the experiment. Hence, the swarm was only required to reach a consensus on the best available option once. This scenario is better suited to model collective decisions that must be followed by the implementation of what has been decided. For instance, a swarm of aerial robots have to decide about the best spot to land before performing the landing. In contrast, in this chapter, the environment changes over time and hence the swarm is required to maintain its collective decision about the best option up-to-date. This scenario corresponds to experimental cases where the primary task of the swarm is to keep track of the best option in the environment and possibly inform other systems that act accordingly. For example, a swarm of aerial robots monitors a disaster environment to keep track of the most urgent task to execute and informs another robotics system on the ground that executes the task.

In dynamic best-of- n decision problems, a swarm of S robots is required to flexibly vary its decision about the best option in response to the changes that occur in the environment. The environmental changes we consider in our study include variations in the number of the available options, that is, new options may appear in the environment, and existing options may disappear (see section 5.2). The number of options at time t is denoted by $n(t)$. We also consider environmental changes where the qualities of the available options vary over time. During an experiment, each of the available options has a unique and constant ID $i \in \{1, \dots, n_T\}$; where n_T is the number of the different options that can be found during an experiment. Each option i has a fixed position in the space χ_i and a time-dependent quality $v_i(t) \in [0, 1]$.

The desired behaviour of the swarm in dynamic best-of- n decision problems is to continually update its decision about the best option following the changes in the environment. This can be formalised as follows:

$$\arg \max_{\chi_i} v_i(t), \quad \text{with } i \in \{1, \dots, n_T\}. \quad (5.1)$$

As shown in the previous chapter and previous studies [164], the outcome of the swarm decision-making best-of- n problems is influenced by the difficulty of the decision problem. The difficulty of the best-of- n problem depends on how similar are the qualities of the available options, the more similar they are, the harder is for the swarm to decide which of the options is the best. In this study, we define the difficulty of the decision problem as the difference between the qualities of the two best options $\Delta v = (v_{\text{best}} - v_{\text{second-best}})$. The higher the value of Δv , the easier the decision problem and vice-versa.

In this study, the robots have no prior knowledge about the decision problem. The robots do not know in advance the number of the available options $n(t)$, their locations χ_i , their qualities $v_i(t)$ (with $i \in \{1, \dots, n_T\}$), or how they will vary over time. The robots only acquire this information by exploring the environment. Moreover, a robot can gather information about an existing option (*i.e.* the location and the quality of the option) only when the option is within its sensing range S_r .

In line with the concepts of swarm robotics [76], in this study, we consider robots that have minimal memory, communication, and sensory capabilities. In terms of sensory capabilities, robots are only able to make noisy estimates of the options' qualities $\hat{v}_i(t)$ (with $i \in \{1, \dots, n_T\}$). To simulate noisy quality estimations, each time a robot makes an estimate $\hat{v}_i(t)$ of the option i 's quality, the value $\hat{v}_i(t)$ is randomly drawn from a normal distribution $\mathcal{N}(v_i(t), \sigma^2)$. The mean of the distribution \mathcal{N} is the true quality of the option $v_i(t)$. The variance of the distribution \mathcal{N} is an arbitrary number that defines how noisy are the robots' estimates. When the quality estimate made by the robot lies outside its quality sensing range $[0, 1]$, we set the estimate to the nearest boundary value. In terms of memory capabilities, each robot can memorise only the quality and the location of its selected option (*i.e.* its opinion about the best option). In terms of communication capabilities, robots can share only a single piece of information that is the location of their selected option. The robots do not share their ID as in [210] not the quality of their selected option as in [159].

In previous collective decision-making studies [152, 180, 135, 27], as soon as the number of robots committed to the same option i reaches or surpasses a quorum threshold Q of the full population, the swarm is considered to have reached consensus for option i . In contrast, in this study, we consider that consensus is reached for option i when the average number of robots committed to an option i in the last T_w time steps is equal or higher than the quorum threshold $Q = 80\%$. This way of assessing consensus is more reliable because it ensures that the swarm has truly settled on one option rather than being in oscillation between multiple options.

5.2 Environmental changes

A large number of research works have previously addressed the best-of- n decision problem [209]. However, only a few of them looked at the problem in case of environmental changes [148, 187, 13]. Moreover, the few works that studied the best-of- n problem in dynamic environments [148, 187, 13] considered only one type of environmental change that is a sudden swap of the qualities of the available options. In the study presented here, we considered a wider range of environmental changes that are the appearance of a new option, the disappearance of the best option, and a swap of the qualities of the two best options.

5.2.1 The appearance of a new option

At the initial time $t = 0$, $n(0) = n_0$ options are available in the environment. At the time T_c , the swarm is fully settled on the best of the n_0 available options, and a new option appears in the environment (*i.e.* $n(t \geq T_c) = n_0 + 1$). If the new option has a higher quality than the previously available best option, the swarm is required to switch its consensus to the new option. In case the new option has an inferior quality than the previously available best option, the swarm should maintain its previous consensus state.

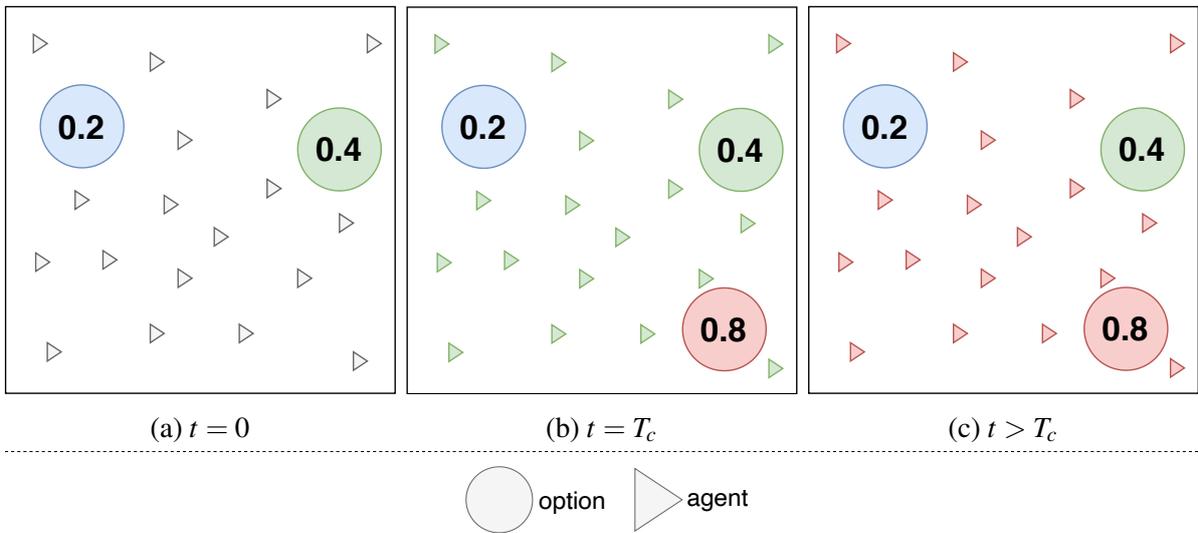


Fig. 5.1 A visual illustration of the *appearance* scenario in which a better option appears in the environment.

In figure 5.1, we illustrate the *appearance* scenario in case of $n_0 = 2$ options. The square represents the environment where the robots operate. The options are represented as colour-coded circles, and the number on top of each circle (*i.e.* option) represents the option's quality. The robots are illustrated by triangles, and the colour of each triangle matches the colour of the option the robot believes is the best. As shown by figure 5.1(a), at the start ($t = 0$), two options are available in the environment (the green option of quality 0.4 and the blue option of quality 0.2). Agents are initially uncommitted (grey-coloured). At the time T_c (figure 5.1(b)), the swarm is settled on the best of the two available options (the green option) when a better option appears (the red coloured option of quality 0.8). In this case, the swarm is required to switch its decision to the new best option (*i.e.* the red option), as shown in figure 5.1(c).

5.2.2 The disappearance of the best option

In this scenario, at the start $n(0) = n_0$ options are available in the environment. At the time T_c , the swarm is fully settled on the best of the n_0 options when this one suddenly disappears (*i.e.* $n(t \geq T_c) = n_0 - 1$). The swarm is required to switch its decision to the new best option (*i.e.* the previously second-best).

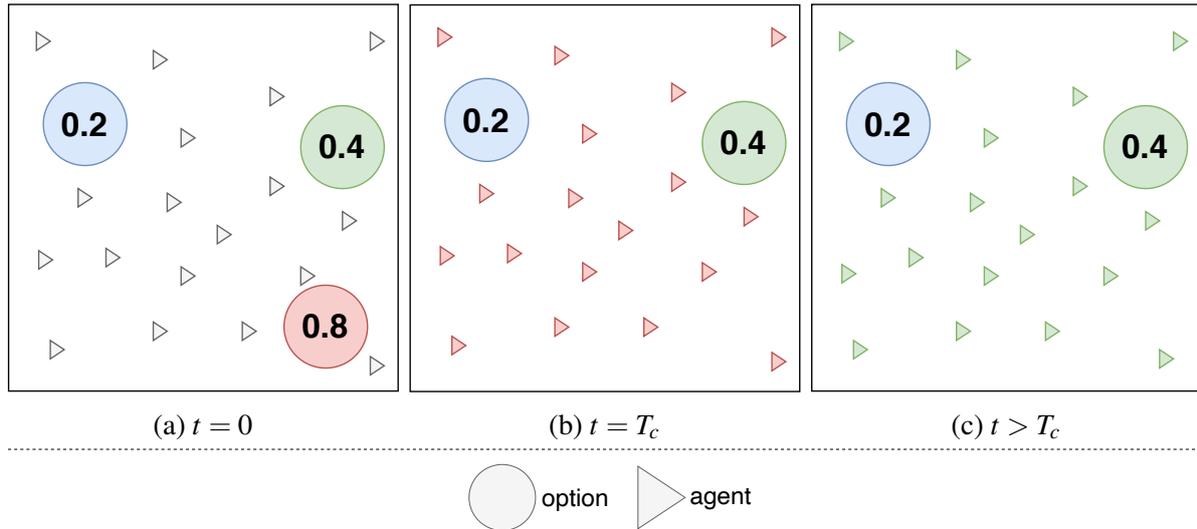


Fig. 5.2 A visual illustration of the *disappearance* scenario in which the best option disappears during the experiment.

In figure 5.2, we illustrate the *disappearance* scenario in case of $n_0 = 3$ options. As shown by figure 5.2(a), at the start ($t = 0$), three options are available in the environment (the red options of quality 0.8, the green option of quality 0.4 and the blue option of quality 0.2). Agents are initially uncommitted (grey-coloured). At the time T_c (figure 5.2(b)), the swarm is settled on the best of the three available options (the red option) when this one suddenly disappears from the environment. In this case, as shown by figure 5.2(c), the swarm is required to switch its decision to the new best option, *i.e.* the green option of quality 0.4 that was previously (*i.e.* $t < T_c$) the second-best option.

5.2.3 A swap of the qualities of the two best options

In this scenario, the number of options remains constant throughout the experiment (*i.e.* $n(t) = n_0 \forall t$). However, at time T_c , while the swarm is settled on the best of the n_0 available option, the quality of the two best options are swapped. The previously second-best option

becomes the new best option. In this case, the swarm needs to switch its decision to the new best option (*i.e.* the initially second-best).

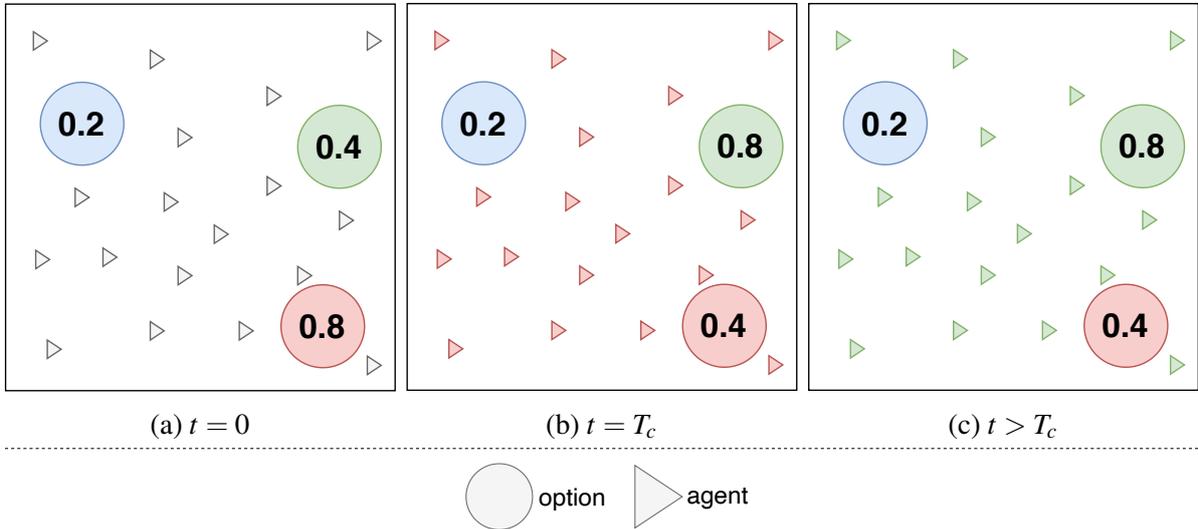


Fig. 5.3 A visual illustration of the *swap* scenario in which the qualities of the two best options are swapped.

In figure 5.3, we illustrate the *swap* scenario in case of $n_0 = 3$ options. As shown by figure 5.3(a), at the start ($t = 0$), three options are available in the environment (the red options of quality 0.8, the green option of quality 0.4 and the blue option of quality 0.2). Agents are initially uncommitted (grey-coloured). At the time T_c (figure 5.3(b)), the swarm is settled on the initially best of the three available options (the red option) when suddenly the qualities of the two best options are swapped. In this case, as shown by figure 5.3(c), the swarm is required to switch its decision to the new best option, *i.e.* the green option of quality 0.8 that was previously (*i.e.* $t < T_c$) the second-best option.

5.3 The individual robot's behaviour

In this study, each robot contributes to the solving of the best-of- n decision problem by performing the same behaviour as the robot's behaviour described in section 4.2 of the previous chapter. In brief, the robot randomly explores the environment through a diffusive isotropic random walk to locate the available options and estimate their qualities. While exploring the environment, a robot interacts with other robots within its communication range to exchange opinions about the best available option. Finally, the robot uses the information gathered through environment exploration (also called *individual* information) and through interaction

with others (also called *social* information) to update its opinion about the best available option via a decision-making model (introduced in section 5.4). In contrast to the previous chapter, in this study, through random exploration, the robots continuously monitor the environment to detect changes. The robots can encounter new appearing options, re-assess the quality of the available options, and update the quality estimate of their options to detect quality changes and verify whether their option is still available or not. Every time a robot passes by the option to which it is committed, the robot re-estimates the option's quality. When a robot passes by its option's location but does not see it, the robot assumes that its option disappeared. When a robot detects that its option disappeared, the robot sets the estimate of its option's quality to zero rather than abandoning the option and becoming uncommitted. By setting the option's quality to zero, the robot stops advertising the option and at the same time avoids being recruited (see section 5.4.1) for the option again.

5.4 The decision-making models

In this study, we propose individual behavioural rules to give robot swarms the ability to adapt their collective decision in case of dynamic best-of- n decision problems. We integrate our individual behavioural rules into a personalised implementation of the existing weighted voter decision-making model [211]. The weighted voter decision-making model, in its original form, does not allow adaptation [148]. Our implementation of the weighted voter model is presented in section 5.4.1. The individual behavioural rules we add to the original weighted voter model to achieve adaptation are introduced in section 5.4.2.

5.4.1 The core of the decision-making models

The decision-making models employed in this study are based on the weighted voter model [211]. The weighted voter model, in its original form [211], has been already demonstrated to not allow adaptation in case of environmental changes [148]. Therefore, the weighted voter model is adequate for testing the effectiveness of the proposed behavioural rules in achieving adaptation.

Our implementation of the weighted voter model is slightly different from the one used in previous studies [211, 148]. In this study, we assume that robots have no prior knowledge about the decision problem. The robots start uncommitted and randomly explore the environment to find the available options. As demonstrated in chapter 4 and in [198], it is important to take into account the initial random exploration because it may be influential on the outcome of the decision process. For these reasons, in our version of the weighted voter model, we include a

quality-dependent discovery transition. An uncommitted robot that encounters an option while exploring the environment may become committed to that option with a probability that is proportional to the option's quality.

In the original implementation of the weighted voter model used in [211, 148], the robot's behaviour is divided into two phases, a dissemination phase where the robot advertises and updates its opinion, and an exploration phase where the robot assesses the quality of its option. In the dissemination phase, the robot advertises its option for a time proportional to its estimate of the option's quality then switches to the exploration phase. In contrast, in our version of the weighted voter model, each robot continuously communicates with the other robots. In each broadcast period, the robot probabilistically decides whether to share its commitment with others or not. The robot shares its commitment with a probability that is proportional to the quality-estimate of its option. For instance, if the quality estimate of the robot's option is equal to 50% of the maximum possible quality, every broadcast period, the robot shares its commitment with a 0.5 probability. Moreover, each robot continually explores the environment in search for new options. Every time the robot passes by its option, the robot re-assesses the quality of the option. When the robot commits to a new option through interaction with others, the robot navigates toward the option's location to self-estimate the option's quality.

Our version of the weighted voter model is implemented using the probabilistic finite state machine (PFSM) shown in Figure 5.4. The active state of the PFSM depends on the robot's commitment state. The state U is active when the robot is uncommitted. The state C_i is active when the robot is committed to option i (with $i \in \{1, \dots, n_T\}$). At each update time-step, the robot executes this model's PFSM to update its opinion based on the information in its possession. The information the robot uses to update its opinion can be either social, *i.e.* coming from other robots in its local neighbourhood, or individual, *i.e.* gathered through individual exploration efforts. Since the probability a robot shares information about its opinion is proportional to the quality of its option; the higher the quality of an option, the more likely a robot is exposed to information about it [167]. When an uncommitted robot holds individual information about option i , *i.e.* the robot has encountered the option i during its exploration of the environment satisfying the condition E_i , the robot commits to option i with a probability P_{D_i} . This transition is called a *discovery* of option i . When a robot is uncommitted or committed to option j and receives information about option $i \neq j$ from another robot (*i.e.* receives a recruitment message $R_{i \neq j}$), the robot commits to option i . We call this transition *recruitment* to option i . When the robot is recruited to option i , the robot navigates toward the option's location χ_i (received in the recruitment message R_i) to self-estimate the option's quality. Once the robot self-estimates the option's quality, it resumes random exploration. As demonstrated in chapter 4, self-estimating the option's quality allows avoiding the spread of individual misestimations.

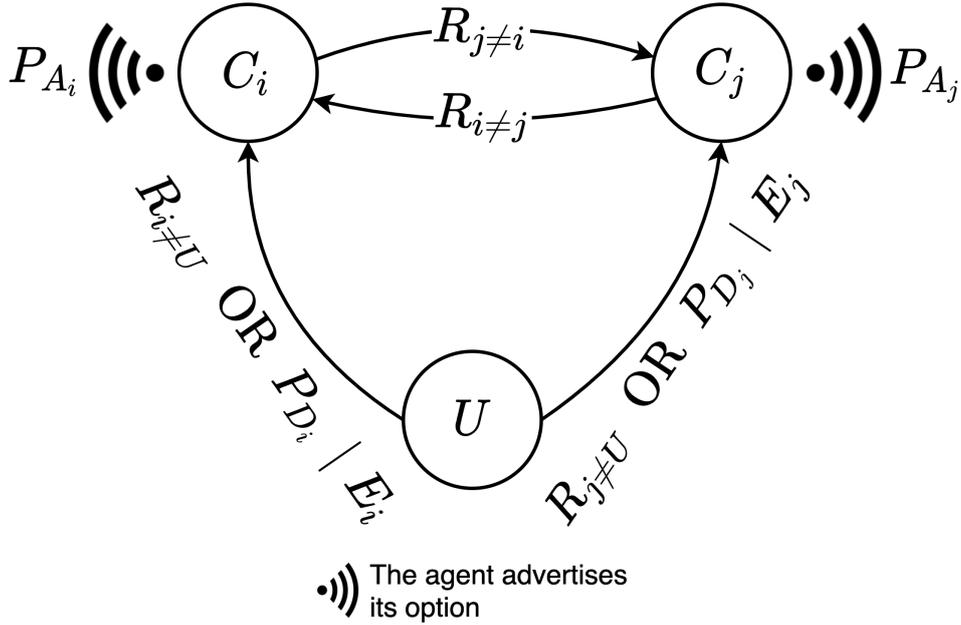


Fig. 5.4 The PFMSM used to implement our version of the weighted voter model [211]. This PFMSM is shown in case of $n = 2$ options but can scale to any number of options with two transitions linking each pair of commitment states. Each robot executes this PFMSM to update its opinion about the best option based on the information it acquires through environment exploration and interaction with peers. Upon the encountering E_i of option i (indicated by the symbol $|$ of conditional probability), an uncommitted robot U commits to the option i with probability P_{D_i} . We call this transition a *discovery* of option i . When a robot is uncommitted (state U) or committed to option j (state C_j) and receives a recruitment message $R_{i \neq j}$ from a robot that is committed to option i , the receiving robot commits to option i . At each broadcast time-step, robots that are committed to option i advertise the option i with probability P_{A_i} .

At each broadcast time-step, each committed robot advertises its option i with probability P_{A_i} . The robot advertises its option i by sending a recruitment message R_i to other robots in its surrounding. The recruitment message R_i contains the option's location χ_i . For the swarm to reach a consensus for the best available options, the discovery and the recruitment transitions must occur proportionally to the quality of the option [117, 167]. In other words, the higher is the quality v_i of option i , the more often a robot should discover or get recruited to option i . Therefore, in our implementation, the discovery and advertising probabilities are the following:

$$P_{D_i} = P_{A_i} = \hat{v}_i, i \in \{1, 2, \dots, n_T\}. \quad (5.2)$$

5.4.2 Local behavioural rules for achieving adaptation

As demonstrated in [148], the weighted voter model, as introduced in [211], does not allow swarms to adapt their collective decisions in case of environmental changes. The reason for this is that the weighted voter model has no mechanism that enables the individual robot to commit to a different option once all the robots reach a consensus on the same option. Here, we introduce two individual behavioural rules, each of which, when added to the weighted voter model, allows the swarm to adapt its decision to environmental changes. These individual behavioural rules make the individual robot always able to commit to a different option even after all of the robots commit to the same choice. The proposed individual behavioural rules are the *Compare* rule and the *Forget* rule.

5.4.2.1 The *Compare* rule

The *Compare* rule enables a committed robot to switch its commitment to an encountered option when it has a higher quality than its current choice. Comparing the quality of the encountered site to a threshold before committing to it has been previously used in decision-making models of social insects [29, 30, 170]. Figure 5.5 shows the PFSM of our version of the weighted voter decision-making model with the *Compare* rule (shown in red colour). Following the *Compare* rule, a robot that is committed to an option i and encounters another option $j \neq i$ (*i.e.* satisfies the condition $E_{j \neq i}$), switches its commitment from option i to j with a probability $P_{S_{ji}}$. The swarm is required to switch its consensus only when its current choice is no longer the best. Thus, the robot must switch its commitment from the option i to the encountered option j only when it estimates that the option j is better than its current option i (*i.e.* $\hat{v}_j > \hat{v}_i$). Besides, for the swarm to always favour options with the highest quality, the switching probability $P_{S_{ji}}$ should also be proportional to the estimated quality \hat{v}_j . Therefore, the probability $P_{S_{ji}}$ can be expressed as follows:

$$P_{S_{ji}} = \hat{v}_j H[v_j - (v_i + k)], \quad i \in \{1, 2, \dots, n_T\} \quad (5.3)$$

where $H()$ is the unit step function, and k is a parameter that sets the minimum required quality difference between the current robot's option i and the encountered option j for the robot to consider switching its opinion to option j . Using the parameter k , it is possible to set the threshold for which updating the opinion is worthwhile. In fact, in certain scenarios, when the qualities v_i and v_j are very similar, it is preferable for the swarm to not adapt its decision even if option j is better than option i (*i.e.* $v_j - v_i < k$).

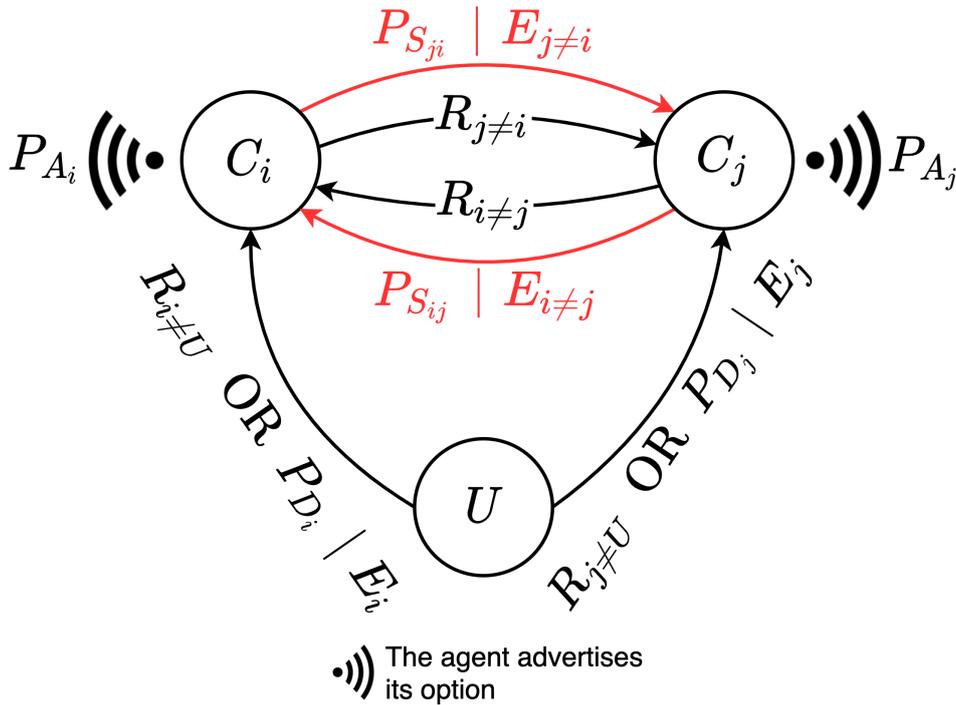


Fig. 5.5 The PFSM of the weighted voter decision-making model with the *Compare* rule (represented with red colour). This model extends our version of the original weighted voter model (see section 5.4.1 and figure 5.4). Following the *Compare* rule, when a robot that is committed to an option i encounters another option $j \neq i$ (i.e. satisfies the condition $E_{j \neq i}$), the robot switches its commitment from option i to j with probability $P_{S_{ji}}$ given by equation (5.3).

5.4.2.2 The *Forget* rule

The second individual behavioural rule that we propose to enable the swarm to adapt its decision in case of environmental changes is the *Forget* rule. Following the *Forget* rule, the robots spontaneously forget their current choice and become uncommitted. The *Forget* rule has previously been included in several collective decision-making and foraging models of social insects such as honeybees [132, 167, 164] and ants [194, 117, 182], and is sometimes referred to as *abandonment* or *leak*. Becoming uncommitted allows robots to re-assess and commit to the encountered options. As a result, the swarm can detect newly available options or changes in the qualities of the available options.

In figure 5.6, we show the PFSM of our version of the weighted voter decision-making model when including the *Forget* rule (shown in red colour). As depicted by figure 5.6, using the *Forget* rule, a robot that is committed to an option i spontaneously forgets its option and becomes uncommitted with probability P_{F_i} . Some previous studies [132, 167, 164] employed the *Forget* rule as an additional mechanism that drives the building of consensus for the best option. Thus, these studies selected the forgetting probability P_{F_i} to be inversely proportional to

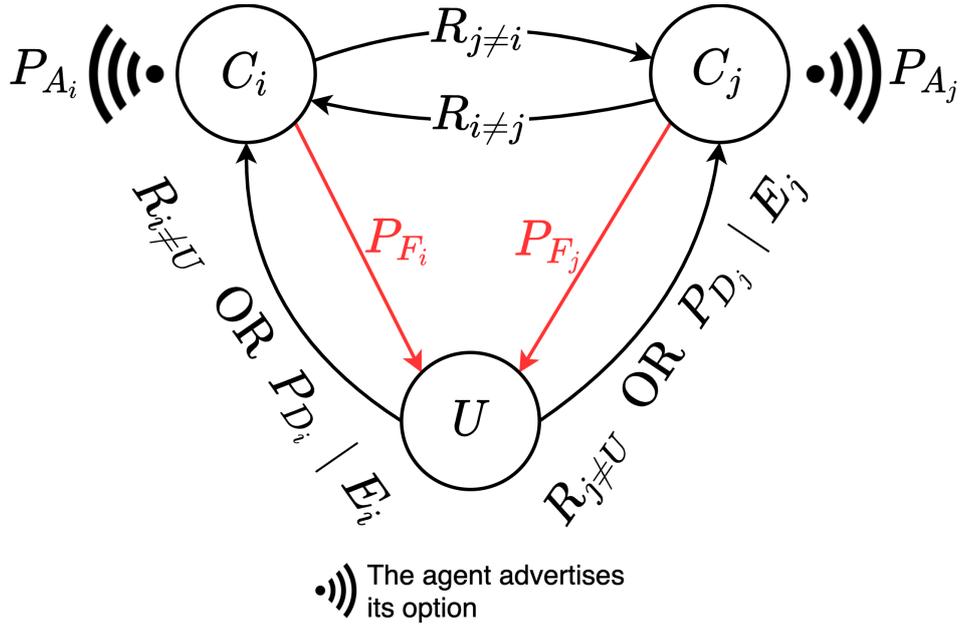


Fig. 5.6 The PFSM of the weighted voter decision-making model when including the *Forget* rule (represented with red colour). This model extends our version of the original weighted voter model (see section 5.4.1 and figure 5.4). Following the *Forget* rule, a robot that is committed to an option i spontaneously forgets its option and becomes uncommitted with probability P_{F_i} given by equation (5.4).

the quality of the robot's option, so the robots abandoned options of low quality more often than options of higher quality leading to more robots committing to better options. In this study, we are interested in how the *Forget* rule could enable the swarm to adapt its collective decisions in case of environmental changes. Therefore, for simplicity, we set the forgetting probability to a constant α :

$$P_{F_i} = \alpha, i \in \{1, 2, \dots, n_T\}. \quad (5.4)$$

5.5 Experimental setup

In this study, we introduce individual behavioural rules that allow the swarm to achieve adaptation in case of dynamic best-of- n problems described in section 5.1 and for the different types of environmental changes introduced in section 5.2. We employ the Decision-Making Multi-Agent Simulator (DeMaMAS) (described in section 3.2) to analyse the collective decisions resulting from the proposed behavioural rules. In section 5.5.1, we describe the experiments conducted in the DeMaMAS simulator. Moreover, to showcase the effectiveness of the analysed behavioural rules in more realistic setups, we tested them on a simulated Kilobot swarm using

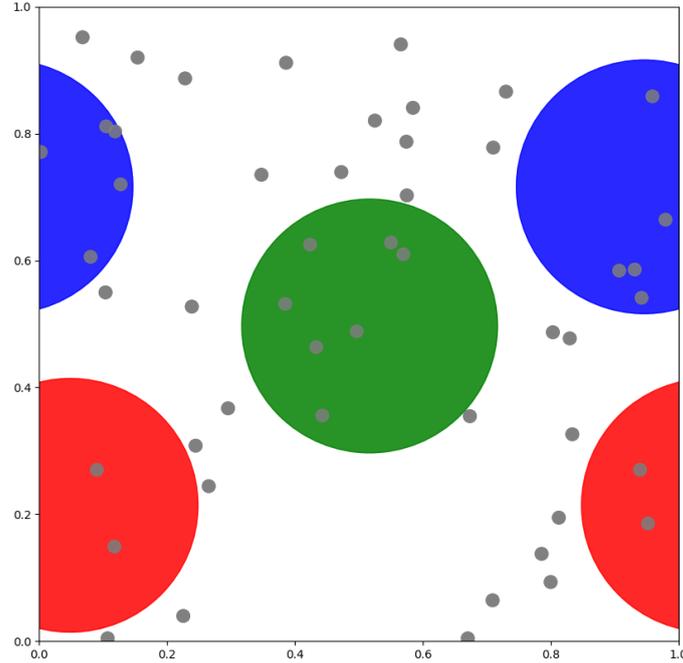


Fig. 5.7 A screenshot of the multi-agent simulations conducted in DeMaMAS (described in section 3.2) for $S = 50$ robots and $n = 3$ options. The frame shows the limits of the 1×1 environment with periodic boundary conditions. The coloured circles represent the areas (radius 0.2) in which the options can be perceived by robots (the small grey circles).

the physics-based swarm robotics simulator ARGoS [139, 138] of section 3.4.1. We describe the conducted swarm robotics experiments in section 5.5.2.

5.5.1 Multi-robot experiments

In this study, we employ multi-agent simulations to analyse the effectiveness of our proposed individual behavioural rules for achieving adaptation in case of the environmental changes described in section 5.2. Moreover, we use multi-agent simulations to investigate the effect of parameters such as the robot density in the environment, the robot's communication range, and the quality difference between the options, on the performance of our proposed strategies.

In our multi-agent experiments, a swarm of S robots is required to solve the best-of- n decision problems with $n = 3$ options. However, our proposed behavioural rules are intended to allow swarms to achieve adaptation in any best-of- n decision problem. To avoid the spatial correlations that may be caused by the positioning of the options, in each experiment, the options' locations χ_i (with $i \in 1, 2, 3$) are randomly and uniformly selected within the considered 1×1 environment. A robot is able to perceive options and estimate their qualities only when these are within its sensing range $S_r = 0.2$. The robot is able to exchange information only

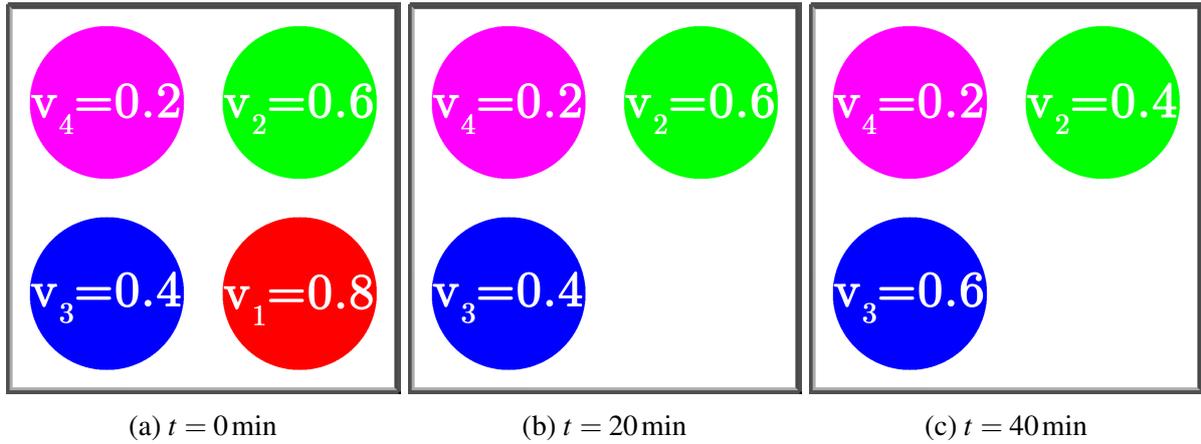


Fig. 5.8 Screenshots of the ARGoS simulator showing the state of the environment (*i.e.* the number n and the qualities v_i of the options) at different times of the swarm robotics experiments. The colour-coded circles (radius $S_r = 0.2$ m) represent the area where each option is perceivable by the robots. At $t = 0$ min (panel (a)), four options are available in the environment: option 1 (red) of quality $v_1 = 0.8$, option 2 (green) of quality $v_2 = 0.6$, option 3 (blue) of quality $v_3 = 0.4$, and option 4 (magenta) of quality $v_4 = 0.2$. At $t = 20$ min, option 1 disappears, and the other options remain unchanged. At $t = 40$ min, the qualities of options 2 and 3 are swapped, *i.e.* $v_2 = 0.4$ and $v_3 = 0.6$.

with robots within its communication range C_r , which we varied in our experiments in the range $C_r \in \{0.025, 0.05, 0.075, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5\}$. The experiment length is 60,000 time steps, and the environmental changes are applied at time step $T_c = 10,000$. We carefully selected T_c to give the swarm enough time to settle on a choice before the environmental change occurs. We say that the swarm has reached consensus for an option if the average number of robots committed to that option in the last $T_w = 5,000$ time steps is equal to or higher than $Q = 80\%$ of the population S .

5.5.2 Swarm robotics experiments

In this study, we validated the effectiveness of the proposed individual behavioural rules in achieving adaptation through their implementation on a simulated Kilobot swarm. Compared to multi-agent simulations, swarm robotics simulations represent a step closer to reality as they take into account collision between robots and noisy robot's motion that may influence the emergent collective behaviour of the swarm.

In the multi-agent simulations, each experiment focused on a single type of environmental change. In contrast, the swarm robotics experiments involved all the three types of environmental changes considered in this study (*i.e.* the appearance of a better option, the disappearance of the swarm's option, and a swap in the qualities of the two best options). The experiments

are one hour long and include up to $n = 4$ options. The environment state (*i.e.* the number and the qualities of the options) is changed every 20 min. At the start ($t = 0$ min), four options are available in the environment (figure 5.8(a)). All the robots are set to be committed to option 2 of quality $v_2 = 0.6$ which is the second-best available option. Starting with a fully committed swarm to the second-best option (the option 2 of quality $v_2 = 0.6$) corresponds to simulating the appearance of a better option, in this case, option 1 of quality $v_1 = 0.8$. At $t = 20$ min, option 1 disappears from the environment and option 2 becomes the best (figure 5.8(b)). At $t = 40$ min, the qualities of options 2 and 3 are swapped (*i.e.* $v_2 = 0.4$ and $v_3 = 0.6$) making option 3 the best (figure 5.8(c)).

To conduct the swarm robotics experiments, we employed the ARGoS Kilobots simulator described in Section 3.4.1. We used a swarm of $S = 50$ Kilobots [174] with a communication range $C_r = 0.1$ m. To match the multi-agent experiments, we consider a $1\text{ m} \times 1\text{ m}$ square environment and the robots' sensing range to be $S_r = 20$ cm. To allow the Kilobots to perceive the options, compute their locations, and estimate their qualities, we employed the simulated ARK system introduced in section 3.4.3. When an option i is within the robot's sensing range (20 cm), the option's location χ_i and noisy quality \hat{v}_i are sent to the robot via an ARK message. We also use ARK to inform the robots about their GPS location and their orientation. Thanks to the GPS location and orientation, the robots can move towards an option to self-estimate its quality. Besides, a robot uses its GPS location to detect whether its option is still available or not. If the robot is around its option (the distance between the robot's location and its option's location is smaller than the robot's sensing range), but the robot does not perceive the option, the robot assumes that its option disappeared. When the robot thinks its option has disappeared, the robot sets the quality of the option to 0. This allows the robot to avoid being recruited for that option again.

5.6 Results

In this section, we present the results of our analysis of the swarm's behaviour in the case of the environmental changes described in section 5.2 when using the decision-making models introduced in section 5.4. Section 5.6.1 shows the results of the experiments conducted using the DeMaMAS multi-agent simulator, while section 5.6.2 presents the results of the swarm robotics simulations.

5.6.1 Multi-robot simulations results

Here, we present the results of the multi-agent simulations we conducted to analyse the effectiveness of the behavioural rules (see section 5.4.2) in enabling the swarm to adapt its decision in case of the environmental changes detailed in section 5.2. By conducting the experiments described in section 5.5.1, we analysed the response of the swarm in best-of- n problems of different difficulty Δv . We also investigated the effect of the parameter of each behavioural rule on the swarm's adaption abilities. Finally, we analysed the effect of the robot density in the environment D and the robot's communication range C_r on the adaptation abilities achieved through the proposed behavioural rules.

5.6.1.1 Performance metrics

In the multi-agent simulation results presented in this section, each experimental condition is tested through 100 simulation runs. For each tested condition, we measure the probability of adaptation (indicated in the following plots using circular markers) as the proportion of runs where the swarm adapts its decision to the new best option, the probability of stagnation (indicated using square markers) as the portion of runs where the swam keeps its decision unchanged, and the indecision probability (indicated using triangular markers) as the proportion of runs where the swarm becomes undecided (*i.e.* the number of robots committed to any of the available options becomes less than the quorum threshold $Q = 80\%$ of the full population). We also measure the adaptation time (indicated using cross markers) as the average time it takes to the swarm to adapt its decision to the new best option. We consider that the swarm has adapted its decision to the new option when the average number of robots committed to the new option in the last $T_w = 5,000$ time steps is equal to or higher than the quorum threshold $Q = 80\%$ of the total number of robots S

5.6.1.2 Effect of problem difficulty on adaptation

To investigate the effect of the proposed behavioural rules on the decisions of the swarm in the case of environmental changes, we conducted the best-of-3 options experiments reported in section 5.5.1 for $S = 50$ robots and variable problem difficulty $\Delta v \in \{0, 0.1, \dots, 0.5\}$. To vary the difficulty $\Delta v = v_1 - v_2$, we fixed the quality of the best option to $v_1 = 0.8$ and the quality of the worst option to $v_3 = 0.1$, and varied the quality of the second-best option $v_2 \in \{0.3, 0.4, \dots, 0.8\}$. We fixed the robot's sensing range to $S_r = 0.2$, the robot's communication range to $C_r = 0.1$, and the level of the quality estimation noise to $\sigma = 0.01$.

The results of each tested condition are obtained via 100 simulation runs for each type of environmental change, and each proposed behavioural rule. We fixed the parameter k of

the *Compare* behavioural rule to $k = 0.1$, so a committed robot considers switching to an encountered option only when estimating that the quality of the option is at least 0.1 better than the quality of its current option (thus $\Delta v \geq 0.1$). We set the parameter α of the *Forget* behavioural rule to $\alpha = 0.01$. Thus, in each simulation time step, each committed robot may forget its current option and become uncommitted with a probability $P_F = 0.01$, that is, on average, once every 100 time-steps.

Figures 5.9 and 5.10 show the results obtained using the *Compare* and the *Forget* behavioural rules, respectively. Each panel in these figures presents the results obtained for a type of environmental change (described in section 5.2). In each tested condition, the outcome of the swarm's decision is reported on the left y-axis using the blue lines and markers. The circular markers report the proportion of runs where the swarm adapted its decision to the environmental change. The square markers show the proportion of runs where the swarm kept its decision unchanged (labelled as *stagnation*), and the triangular markers show the proportion of runs where the swarm became undecided. On the right y-axis and using the red x-shaped markers, we report the average adaptation time for the proportion of runs where the swarm adapted its decision. The adaptation time measures the number of time steps it takes for the swarm to reach a consensus for the new best option from when the environmental change happens.

As shown by figure 5.9, using the *Compare* behavioural rule (introduced in section 5.4.2.1), the swarm is able to adapt its decision to the different environmental changes. In the *appearance* scenario (figure 5.9(a)), for $\Delta v \in \{0.1, 0.2, \dots, 0.5\}$, the swarm switched its decision to the new better option in 100% of the runs. When the appeared option had the same quality as the swarm's current choice (*i.e.* $\Delta v = 0$), thanks to the parameter $k = 0.1$ that prevented the robots from switching to options with less than 0.1 improvement in quality, the swarm kept its decision unchanged. The swarm took on average between around 4,200 and 5,500 time steps to switch its decision to the appeared option. The higher the difference between the quality of the two best options Δv (*i.e.* the easier is the problem), the quicker the swarm adapted its decision to the new option. In the *disappearance* scenario (figure 5.9(b)), for $\Delta v \in \{0, 0.1, \dots, 0.5\}$, the swarm was able to switch its decision to the best of the remaining options (the previously second-best option) in 100% of the runs. In the *disappearance* scenario, the average adaptation time was between around 4,500 and 5,000 times steps. The average adaptation time decreased as the difference in quality Δv decreased, because for smaller Δv the quality v_2 of the remaining best option is higher, and thus robots commit to it more often and reach consensus faster. In the *swap* scenario (figure 5.9(c)), for $\Delta v \in \{0, 0.1, \dots, 0.5\}$, the swarm adapted its decision to the new best option (the previously second-best option) in 100% of the runs. The swarm took on average between around 4,700 and 6,100 time steps to reach a consensus for the new best option. The smaller was Δv , the more it took for the swarm to adapt its decision.

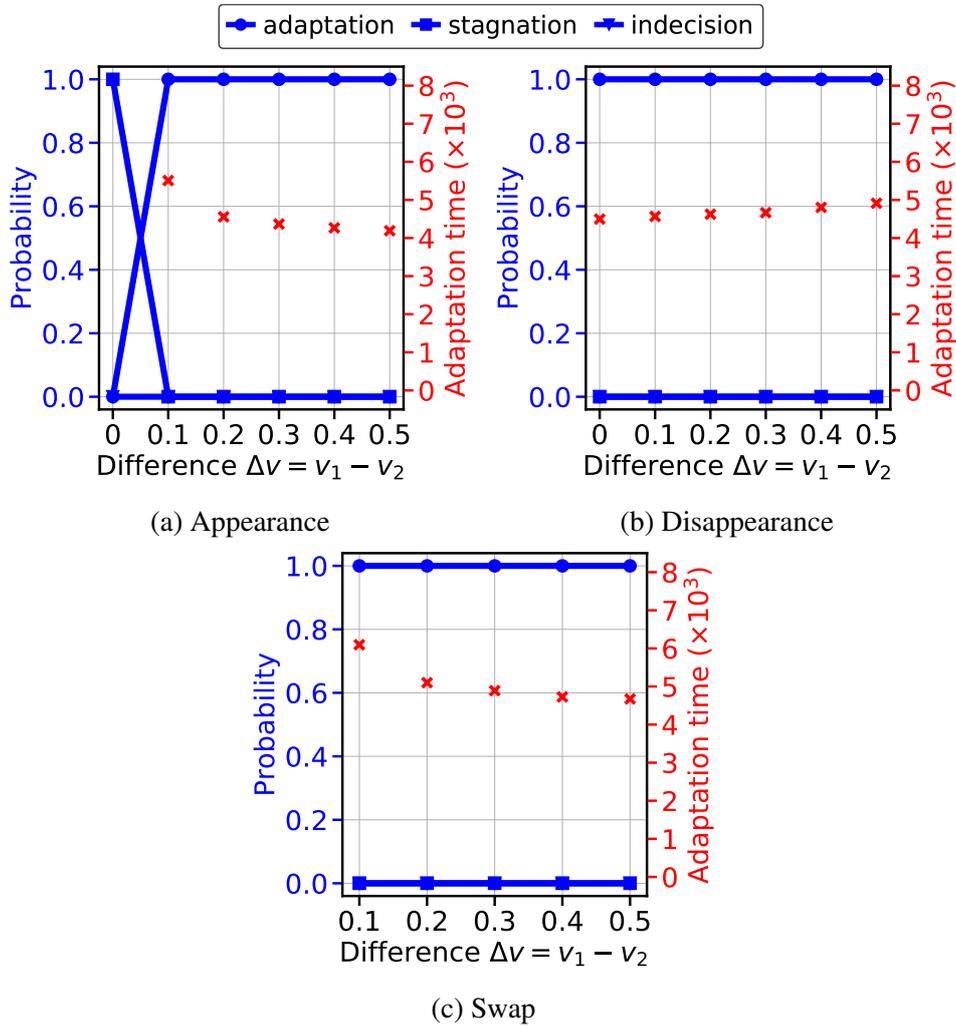


Fig. 5.9 Results of multi-agent simulation experiments described in section 5.5.1 for $S = 50$ -robots swarm in the case of the dynamic best-of-3 options problem described in section 5.2 when using the *Compare* behavioural rule introduced in section 5.4.2.1. Panels (a), (b), and (c) show the results in the *appearance*, the *disappearance*, and the *swap* scenarios (described in section 5.2.1), respectively. In each panel, we vary the difficulty $\Delta v \in \{0, 0.1, 0.2, \dots, 0.5\}$ by varying the quality of the second-best option $v_2 \in \{0.3, 0.4, \dots, 0.8\}$, and fixing the quality of the best option to $v_1 = 0.8$ and the quality of the worse option to $v_3 = 0.1$. We fixed the noise level of the robots' quality estimates to $\sigma = 0.01$ and the parameter of the *Compare* behavioural rule to $k = 0.1$. In each tested condition, the results are obtained through 100 simulation runs. The outcome of the swarm's decision is reported on the left y-axis using the blue lines and markers. The circular markers report the proportion of runs where the swarm adapted its decision to the environmental change. The square markers show the proportion of runs where the swarm kept its decision unchanged, and the triangular markers show the proportion of runs where the swarm became undecided. The adaptation time is reported on the right y-axis using the red x-shaped markers.

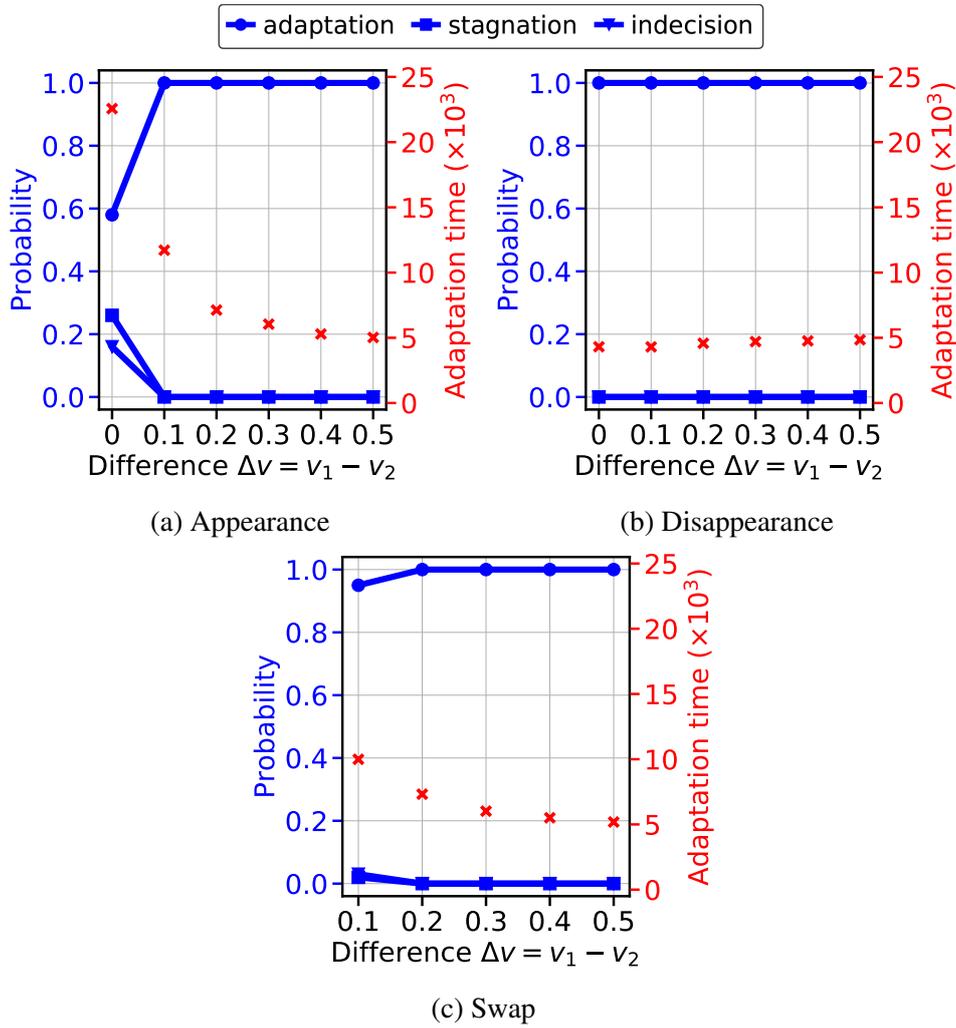


Fig. 5.10 Results of multi-agent simulation experiments described in section 5.5.1 for $S = 50$ -robots swarm in the case of the dynamic best-of-3 options problem described in section 5.2 when using the *Forget* behavioural rule introduced in section 5.4.2.2. Panels (a), (b), and (c) show the results in the *appearance*, the *disappearance*, and the *swap* scenarios (described in section 5.2.1), respectively. In each panel, we vary the difficulty $\Delta v \in \{0, 0.1, \dots, 0.5\}$ by varying the quality of the second-best option $v_2 \in \{0.3, 0.4, \dots, 0.8\}$, and fixing the quality of the best option to $v_1 = 0.8$ and the quality of the worse option to $v_3 = 0.1$. We fixed the noise level of the robots quality estimates to $\sigma = 0.01$ and the parameter of the *Forget* behavioural rule to $\alpha = 0.01$. In each tested condition, the results are obtained through 100 simulation runs. The outcome of the swarm's decision is reported on the left y-axis using the blue lines and markers. The circular markers report the proportion of runs where the swarm adapted its decision to the environmental change. The square markers show the proportion of runs where the swarm kept its decision unchanged, and the triangular markers show the proportion of runs where the swarm became undecided. The adaptation time is reported on the right y-axis using the red x-shaped markers.

As depicted by figure 5.10, the *Forget* rule (introduced in section 5.4.2.2) also enables the swarm to adapt its decision to the different environmental changes. In the *appearance* scenario (figure 5.10(a)), for $\Delta v \in \{0.1, 0.2, \dots, 0.5\}$, the swarm adapted its decision to the appeared option in 100% of the runs. For $\Delta v = 0$, the swarm switched its decision to the appeared option in around 60% of the runs, it kept the same decision in around 25% of the runs, and it became undecided in around 15% of the runs. The average adaptation time increased as the difference between the quality of the two options Δv decreased. The average adaptation time rose from around 5,000 time steps for $\Delta v = 0.5$ to around 12,000 time steps for $\Delta v = 0.1$. In the *disappearance* scenario (figure 5.10(b)), for $\Delta v \in \{0, 0.1, \dots, 0.5\}$, the swarm successfully detected the disappearance of its current option and switched its decision to the new best option (*i.e.* the previously second-best option) in 100% of the runs. The swarm took on average between around 4,300 and 4,800 time steps to adapt its decision to the new best option. The lower was Δv , the quicker the swarm adapted its decision as a lower Δv means a higher quality of the new best option v_2 and thus a faster consensus. In the *swap* scenario (figure 5.10(c)), for $\Delta v \in \{0.1, 0.2, \dots, 0.5\}$, the swarm adapted its decision to the new best option (the previously second-best option) in almost 100% of the runs. The average decision time increased as Δv decreased, going from around 5,200 time steps for $\Delta v = 0.5$ to around 10,000 time steps for $\Delta v = 0.1$.

The adaptation using the *Forget* rule (figure 5.10) is slower than when using the *Compare* rule (5.9), especially for lower values of Δv . For instance, in the *appearance* scenario and for $\Delta v = 0.1$, the average adaptation time was around 12,000 time steps using the *Forget* rule and around 5,500 time steps using the *Compare* rule, *i.e.* when using *Forget*, the swarm is approximatively two times slower than when using the *Compare* rule. When using the *Forget* rule, a robot only commits to the option i (proportionally to the quality estimate \hat{v}_i) if it is uncommitted. Robots that are already committed ignore the other options they encounter in the environment. Thus the probability that a robot commits to the new best option i using the *Forget* rule is proportional to $P_F \cdot P_E \cdot \hat{v}_i$ (where P_E is the probability of encountering the option). In contrast, using the *Compare* rule, all robots that encounter the better option i may commit to the option proportionally to their quality estimate \hat{v}_i of the option's quality. Hence, the probability that a robot commits to new best option i using the *Compare* rule is proportional to $P_E \cdot \hat{v}_i$. Therefore, the probability that a robot commits to the new best option using the *Compare* rule is approximatively $\frac{P_E \cdot \hat{v}_i}{P_{F_{j \neq i}} \cdot P_{E_i} \cdot \hat{v}_i} = \frac{1}{P_{F_{j \neq i}}} = \frac{1}{\alpha} = 100$ times higher than when using the *Forget* rule. For this reason, the adaptation using the *Compare* rule is faster than when using the *Forget* rule.

5.6.1.3 Effect of the behavioural rules' parameters on adaptation

Here, we investigate the effect of the parameter of each behavioural rule on the adaptation abilities of the swarm. We conducted the same experiments as section 5.6.1.2 for different values of the parameter of each behavioural rule. We varied the parameter k (introduced in (5.3)) of the *Compare* rule in $\{0.1, 0.2, 0.3\}$ and varied the parameter α (introduced in equation (5.4)) of the *Forget* rule in $\{0.0001, 0.001, 0.01, 0.1\}$. We tested each experimental condition through 100 simulation runs.

Figure 5.11 shows the results of the experiments when using the *Compare* rule. In each tested condition, we show the proportion of runs where the swarm adapted its decision to the environmental change. The red solid lines and markers show the results for $k = 0.1$, the green dashed lines and markers show the results for $k = 0.2$, and the blue dashed-dotted line and markers indicate the result for $k = 0.3$. As shown by figure 5.11(a), in the *appearance* scenario, the swarm adapted its decision to the new option only when $\Delta v \geq k$; *i.e.* only when the quality of the new option is better than the quality of the existing best option by at least k . As depicted by figure 5.11(b), in the *disappearance* scenario, the value of the parameter k did not influence the adaptation abilities of the swarm. This is due to the fact that when the robots detect that their option disappeared, they set the quality of their option to zero and hence are able to switch their commitment to any encountered option that has a quality of at least k (as expressed by the switching probability of equation (5.3)). As in the *appearance* scenario, in the *swap* scenario (figure 5.11(c)), the swarm adapted its decision to the new option only when $\Delta v \geq k$.

As demonstrated by the results shown in figure 5.11, the parameter k can be used to control the minimum improvement in quality that must occur for the swarm to consider adapting its decision. For example, it may be undesirable that the swarm takes time to adapt its decision for minor improvements in quality.

The results obtained when using the *Forget* rule are depicted in figure 5.12. In each tested case, we report the proportion of runs where the swarm adapted its decision to the environmental change. The red solid lines and markers show the results for $\alpha = 0.0001$, the green dashed lines and markers show the results for $\alpha = 0.001$, the blue dashed-dotted line and markers indicate the result for $\alpha = 0.01$ and the black dotted lines and markers show the results for $\alpha = 0.1$. In the *appearance* (figure 5.12(a)) and the *swap* (figure 5.12(c)) scenarios, for $\alpha = 0.1$ (the black dotted lines), the swarm is unable to adapt in most, when not all, the runs even when the appeared option is much better than the existing options. For $\Delta v = 0.5$ the swarm adapts its decision to the new option only in around 10% of the runs. The reason for this is that for $\alpha = 0.1$ the robots forget their choice so often that they are unable to reach a consensus. For $\alpha = 0.0001$ (the red solid lines), the swarm is also unable to adapt its decision in most of the runs even when the new option is much better than the existing options (*i.e.* $\Delta v = 0.5$). For

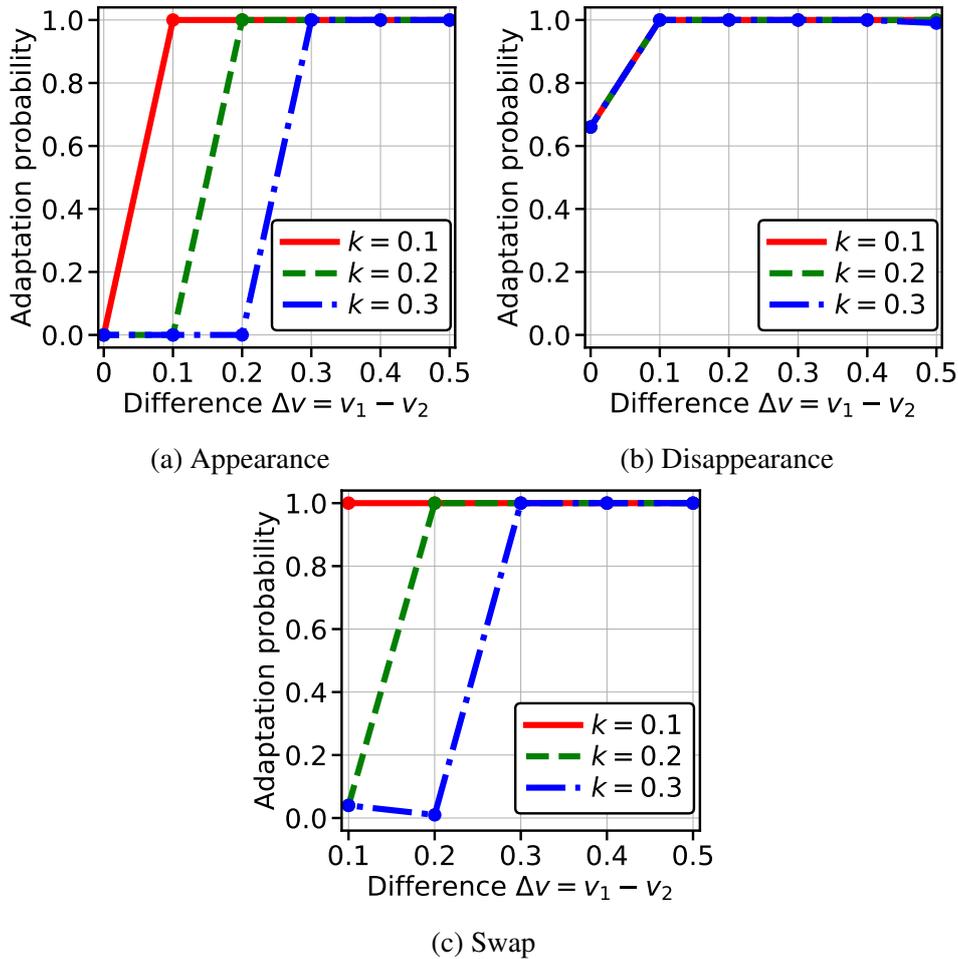


Fig. 5.11 Effect of the parameter k of equation (5.3) on adaptation when using the *Compare* rule.

$\Delta v = 0.5$ the swarm adapted its decision to the new option in only around 40% of the runs. This is due to the fact that for $\alpha = 0.0001$ the robots rarely forget their choice and hence are not able to consider the new option. For $\alpha = 0.01$ (the blue dashed-dotted line), the swarm was able to adapt its decision in nearly 100% of the runs for $\Delta v \in \{0.1, 0.2, \dots, 0.5\}$. For $\alpha = 0.001$ (the green dashed line), the adaptation decreased compared to $\alpha = 0.01$, especially for low values of Δv . For instance, in the *appearance* scenario, for $\Delta v = 0.1$, the adaptation decreased from 100% for $\alpha = 0.01$ to around 75% for $\alpha = 0.001$. As depicted by figure 5.12(b), in the *disappearance* scenario, the adaptation is not much influenced by the value of α , except for $\alpha = 0.1$ where the swarm is generally not able to adapt because the robots too frequently forget their choice, so they are not able to reach a consensus. The reason the value of alpha does not much influence adaptation in the *disappearance* scenario is that when the robots detect that their option disappeared they set its quality to zero and thus stop advertising the option. It is

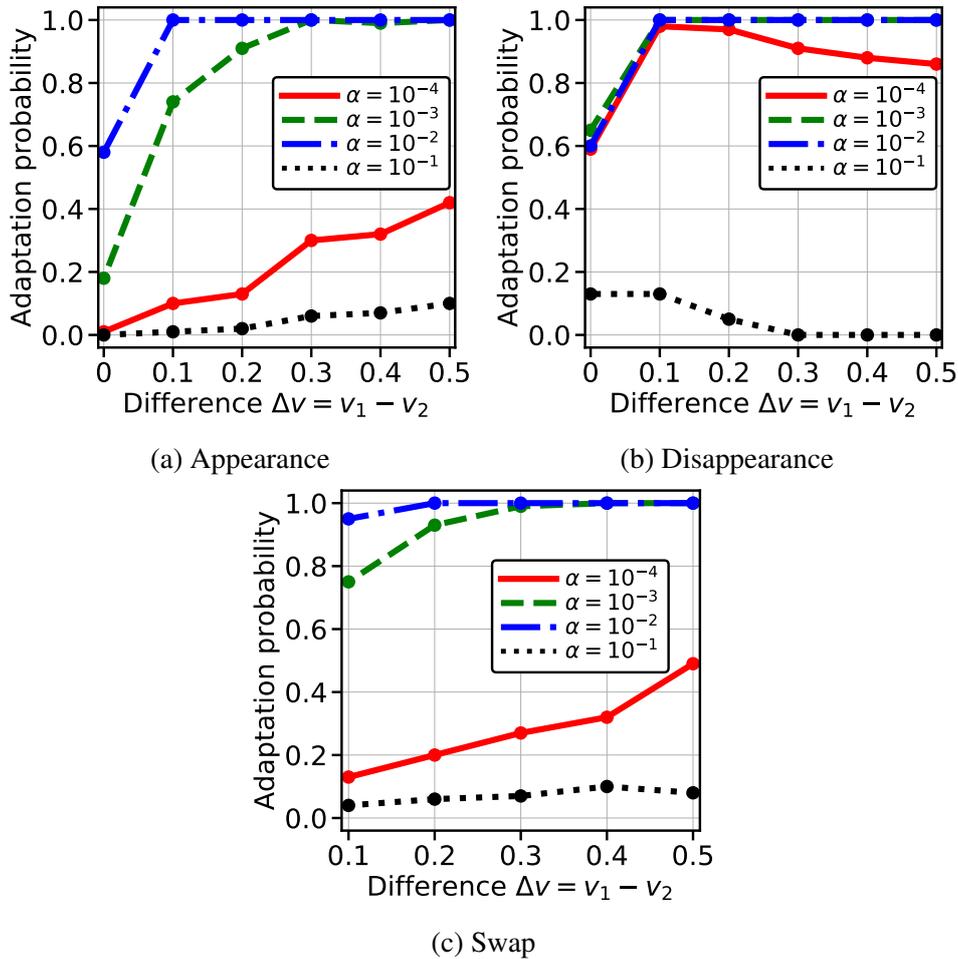


Fig. 5.12 Effect of the parameter α of equation (5.4) on the adaptation when using the *Forget* rule.

then enough that one robot forgets the disappeared choice and commits to the new best (the previously second-best option) that this choice quickly spreads within the population (*i.e.* the swarm adapts).

As demonstrated by the results shown in figure 5.12, the forgetting probability value α must be carefully tuned to achieve the desired adaptive decision. This result is in agreement with a previous result on signal detection in social insects where the value of the forgetting rate had to be tuned in order to detect signals in the presence of noise [97].

5.6.1.4 Effect of the robot's communication range on adaptation

Here, we analyse the effect of the robot's communication range on the adaptation abilities given to the swarm by the *Compare* and the *Forget* behavioural rules introduced in section 5.4.2. We performed our analysis in the *appearance* scenario (described in section 5.2.1) where

two options are initially available in the environment (*i.e.* $n(0) = 2$), the option 2 is of quality $v_2 = 0.7$ and the option 3 is of quality $v_3 = 0.1$. At time step $T_c = 10,000$, option 1 of quality $v_1 = 0.8$ appears (*i.e.* $n(t \geq T_c) = 3$). We fixed the number of robots to $S = 200$, the level of the quality estimation noise to $\sigma = 0.01$, and the robot's sensing range to $S_r = 0.2$. We fixed the parameter k of the *Compare* rule to 0.1 and the parameter α of the *Forget* rule to 0.01. We varied the robot's communication range $C_r \in \{0.025, 0.05, 0.075, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5\}$. The results of our analysis are obtained through 100 simulation runs for each tested condition, and are reported in figure 5.13. For each tested condition, we report the proportion of runs where the swarm switched its decision to the new option using the circular markers and the proportion of runs where the swarm kept the same decision using the square markers.

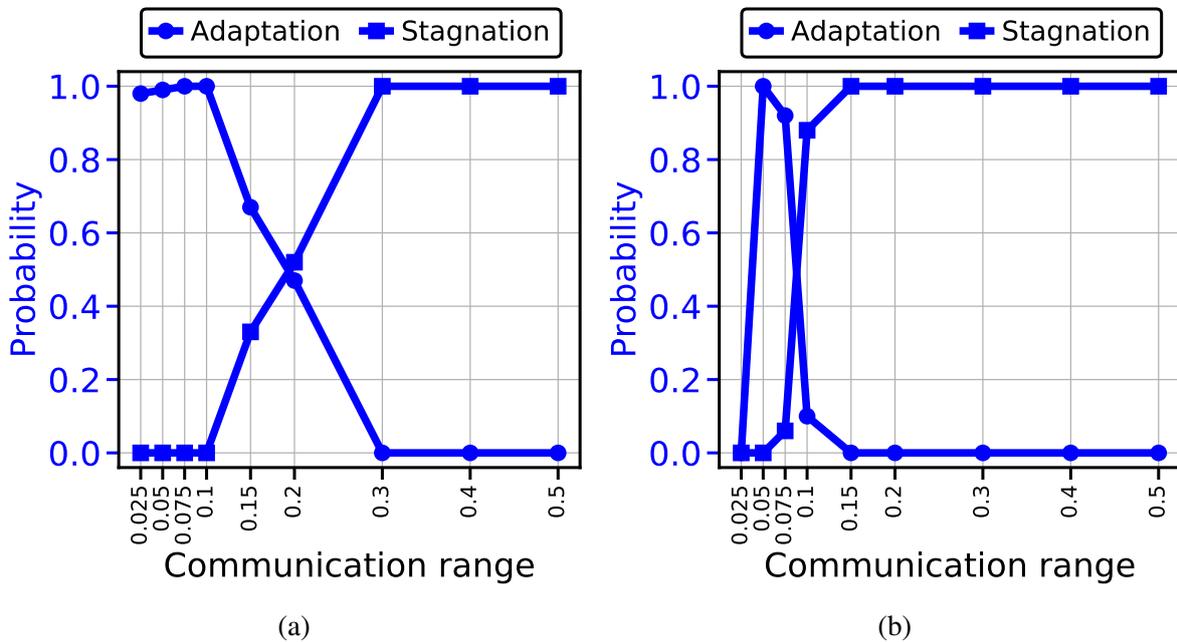


Fig. 5.13 Effect of the robot's communication range on adaptation in the *appearance* scenario described in section 5.2.1. Two options are initially available in the environment (option 2 of quality $v_2 = 0.7$ and option 3 of quality $v_3 = 0.1$). Option 1 of quality $v_1 = 0.8$ appears at time step $T_c = 10,000$. The swarm size is fixed to $S = 200$, and the quality estimation noise is set to $\sigma = 0.01$. The robot's sensing range is fixed to $S_r = 0.2$ and the robot's communication range C_r is varied in $\{0.025, 0.05, 0.075, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5\}$. Panel (a) shows the results when using the *Compare* rule with $k = 0.1$. Panel (b) shows the results when using the *Forget* rule with $\alpha = 0.01$. The results of each tested condition are obtained through 100 simulation runs. In each tested condition, the proportion of runs where the swarm adapted its decision to the new option is reported using the circular markers while the proportion of runs where the swarm kept the same decision is reported using the square markers.

Counter-intuitively, the adaption decreases as the robot's communication range increases, both when using the *Compare* rule (figure 5.13(a)) and when using the *Forget* rule (figure 5.13(b)). For instance, with a communication range of $C_r = 0.05$ the swarm adapts in 100% of the runs while with a communication range of $C_r = 0.5$ the swarm is not able to adapt at all. The reason for this is that longer communication ranges make the spread of new opinions within the swarm harder. The longer is the communication range, the higher the number of peers each robot communicates with at a time. Therefore, when a robot commits to the new option while the swarm is fully settled on another option, the robot gets quickly converted back to the previous choice, especially when the difference between the two options is small. This result is particularly interesting as it highlights the benefit of local communication that characterises both living and artificial swarms.

5.6.1.5 Effect of the robot density on adaptation

In this section, we investigate the effect of the robot density in the environment on the adaptation abilities achieved by the swarm using the proposed behavioural rules. The robot density D is computed as the ratio between the number of robots S and the area of the environment A_E . In this study, to vary the robot density, we kept the size of the environment $A_E = 1$ constant and varied the number of robots S . We tested various values of robot density $D = \frac{S}{A_E} \in \{50, 100, 200, 300, 400, 500, 1000\}$ in the *appearance* scenario of section 5.2.1. At the start, option 2 (of quality $v_2 = 0.7$) and option 3 (of quality $v_3 = 0.1$) are available in the environment (*i.e.* $n(0) = 2$). The option 1 of quality $v_1 = 0.8$ becomes available later at time step $T_c = 10,000$ (*i.e.* $n(t \geq T_c) = 3$). We fixed the robot's sensing range is fixed to $S_r = 0.2$, the robot's communication range to $C_r = 0.1$, and the level of the quality estimation noise to $\sigma = 0.01$. We set the parameter k of the *Compare* rule to 0.1 and the parameter α of the *Forget* rule to 0.01. We ran 100 simulations for each tested condition and each proposed behavioural rule. The results of our analysis are reported in figure 5.14. In each tested condition, the proportion of runs where the swarm adapted its decision to the appeared option is reported using circular markers, while the proportion of runs where the swarm kept its decision unchanged is reported using square markers.

As depicted by figure 5.14, adaption decreases as the robot density increases. For instance, when using the *Compare* rule (fig 5.14(a)), a swarm of $S = 50$ robots adapts in 100% of the runs while a swarm of $S = 500$ robots adapts in only 80% of the runs. When using the *Forget* rule (fig 5.14(b)), a swarm of $S = 50$ robots adapts in 100% of the runs while a swarm $S = 500$ robots does not adapt at all. The reason for this is that the higher the number of robots, the harder it is for a new opinion to spread. The higher the robot density; the higher the number of peers that each robot communicates with at a time. As a result, when a robot commits to a new

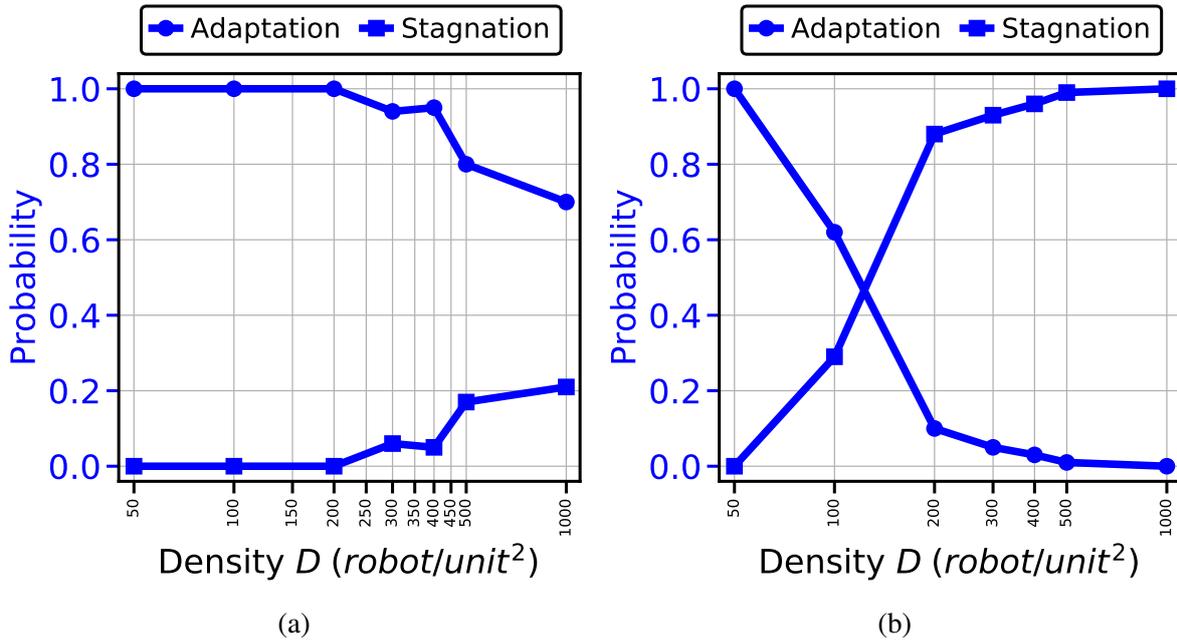


Fig. 5.14 Effect of the robot density on adaptation in the *appearance* scenario described in section 5.2.1. Two options are initially available in the environment (option 2 of quality $v_2 = 0.7$ and option 3 of quality $v_3 = 0.1$). Option 1 of quality $v_1 = 0.8$ appears at time step $T_c = 10,000$. The robot density D is varied in $\{50, 100, 200, 300, 400, 500, 1000\}$. The robot's sensing range is fixed to $S_r = 0.2$, the robot's communication range is set to $C_r = 0.1$, and the quality estimation noise is set to $\sigma = 0.01$. Panel (a) shows the results when using the *Compare* rule with $k = 0.1$. Panel (b) shows the results when using the *Forget* rule with $\alpha = 0.01$. The results of each tested condition are obtained through 100 simulation runs. In each tested condition, the proportion of runs where the swarm adapted its decision to the new option is reported using the circular markers while the proportion of runs where the swarm kept the same decision is reported using the square markers.

option while the swarm is already decided on another option, the robot gets quickly converted back to the previous choice by its surrounding peers. This situation happens especially when the difference between the new option's quality and the quality of the swarm's option is small. In this case, the probability P_A with which the robot advertises the new option is not superior enough to compete with its peers' advertisement for the current swarm's choice.

5.6.2 Swarm robotics simulations results

In this section, we present the simulation results of the swarm robotics experiment described in section 5.5.2 and conducted using the ARGoS swarm robotics simulator [139, 138] introduced in section 3.4. Figure 5.15 shows the time evolution of the fraction of robots committed to each option. The coloured lines show the average fraction of robots over 100 simulation runs.

The coloured shade around each line shows the 95% confidence interval. The colour of each line and shade matches the colour of the corresponding option. The vertical black dashed lines show the times at which environmental changes occur and mark the different parts of the experiment. The available options in each part of the experiment, their qualities, and their colours are indicated on the top of each figure. The horizontal black dashed line indicates the quorum threshold $Q = 80\%$. As shown by figure 5.15, the swarm was able to adapt to the different types of environmental change both using the *Compare* rule (figure 5.15(a)) and the *Forget* rule (figure 5.15(b)). Figures 5.16 and 5.17 show screenshots from a simulation of the swarm robotics experiment when using the *Compare* rule and the *Forget* rule, respectively.

5.7 Discussion

In this chapter, we proposed individual behavioural rules to give robot swarms the ability to adapt their decisions in response to environmental changes. We tested the proposed behavioural rules for solving the best-of- n decision problem in dynamic environments where the swarm is not only required to decide about the best available option in the environment but also needs to keep its decision up-to-date with changes that occur in the environment. In the context of the best-of- n decision problem, environmental changes include variations in the number of available options and their qualities. In this study, we considered the appearance of a better option, the disappearance of the swarm's option, and a sudden swap in the qualities of the two best options. To allow the swarm to adapt its decision to these environmental changes, we proposed two simple individual behavioural rules. The first is the *Compare* rule through which committed robots can commit to a different encountered option when estimating that this one has a better quality than their current option. The second is the *Forget* rule through which committed robots spontaneously forget their current option and hence are able to reconsider encountered options. We integrated these behavioural rules into the weighted voter model [211] for collective decision-making that has been shown to not allow adaption in its original form [148].

Using multi-agent simulations, we demonstrated the effectiveness of our proposed behavioural rules in enabling the swarm to adapt its decision to environmental changes. We assessed the adaptation performance of relatively small swarms ($S = 50$) in best-of- n problems of various difficulties Δv . Both behavioural rules enabled the swarm to adapt its decision when a better option became available in the environment. The swarm adapted its decision faster when using the *Compare* rule because the probability that a robot commits to a new option using the *Compare* rule is considerably higher than when using the *Forget* rule. Moreover, when using the *Compare* rule, the swarm kept its decision unchanged when a new option of similar quality

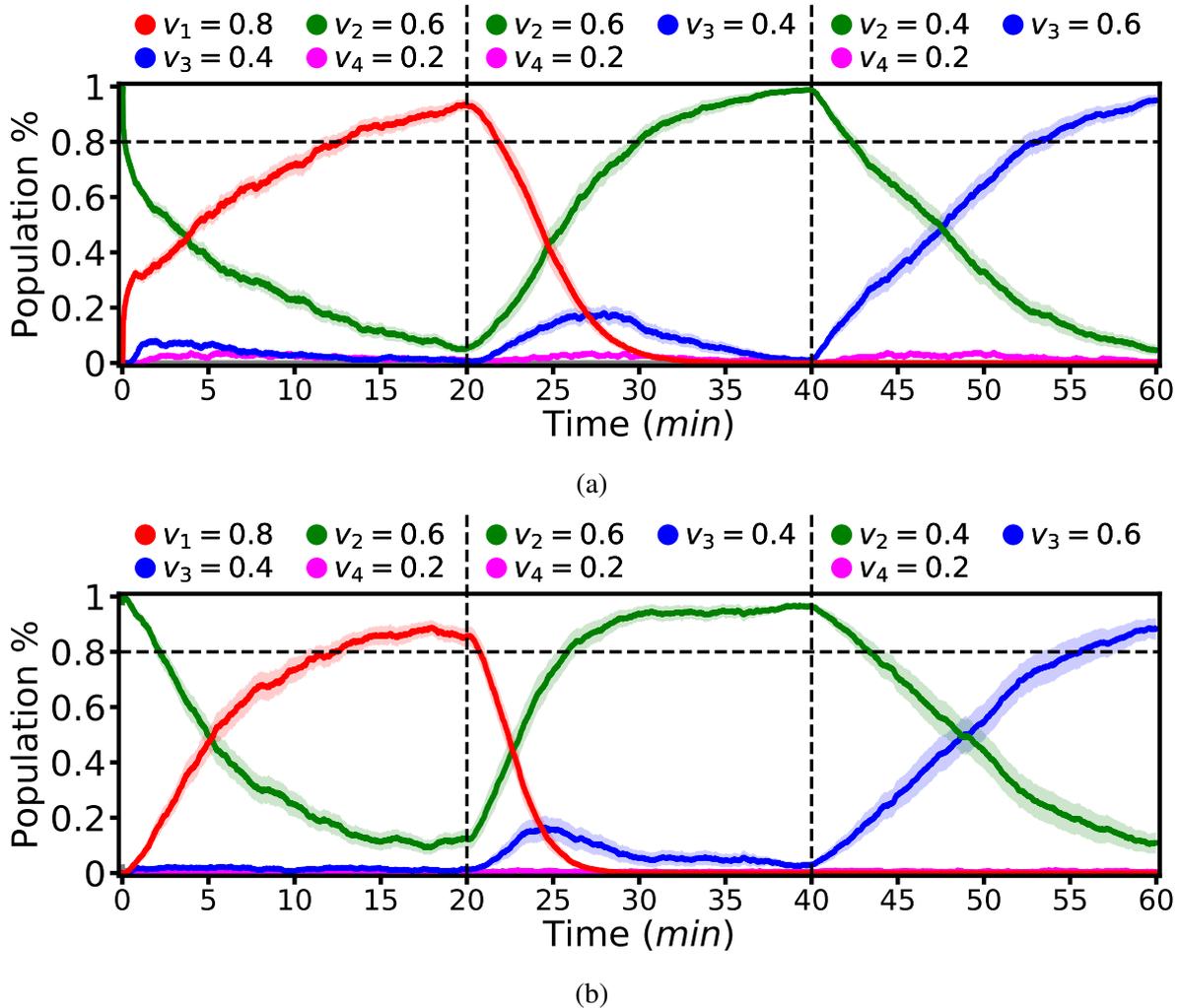


Fig. 5.15 Time evolution of the fraction of robots committed to each option in simulated swarm robotics experiments described in section 5.5.2. The coloured lines show the average fraction of robots over 100 simulation runs. The coloured shade around each line shows the 95% confidence interval. The colour of each line and shade matches the colour of the corresponding option. The vertical black dashed lines show the times at which environmental changes occur and mark the different part of the experiment. The available options in each part of the experiment, their qualities, and their colours are shown on the top of each plot. The horizontal black dashed line indicates the quorum threshold $Q = 80\%$. Panel (a) shows the results when each robot executes the *Compare*-based decision-making model of section 5.4.2.1. Panel (b) shows the results when each robot executes the *Forget*-based decision-making model of section 5.4.2.2.

appeared in the environment. However, when using the *Forget* rule, the swarm sometimes adapted its decision to the appeared option. The reason why the *Compare* rule allowed the swarm to keep its decision unchanged when the appeared option is of similar quality is the

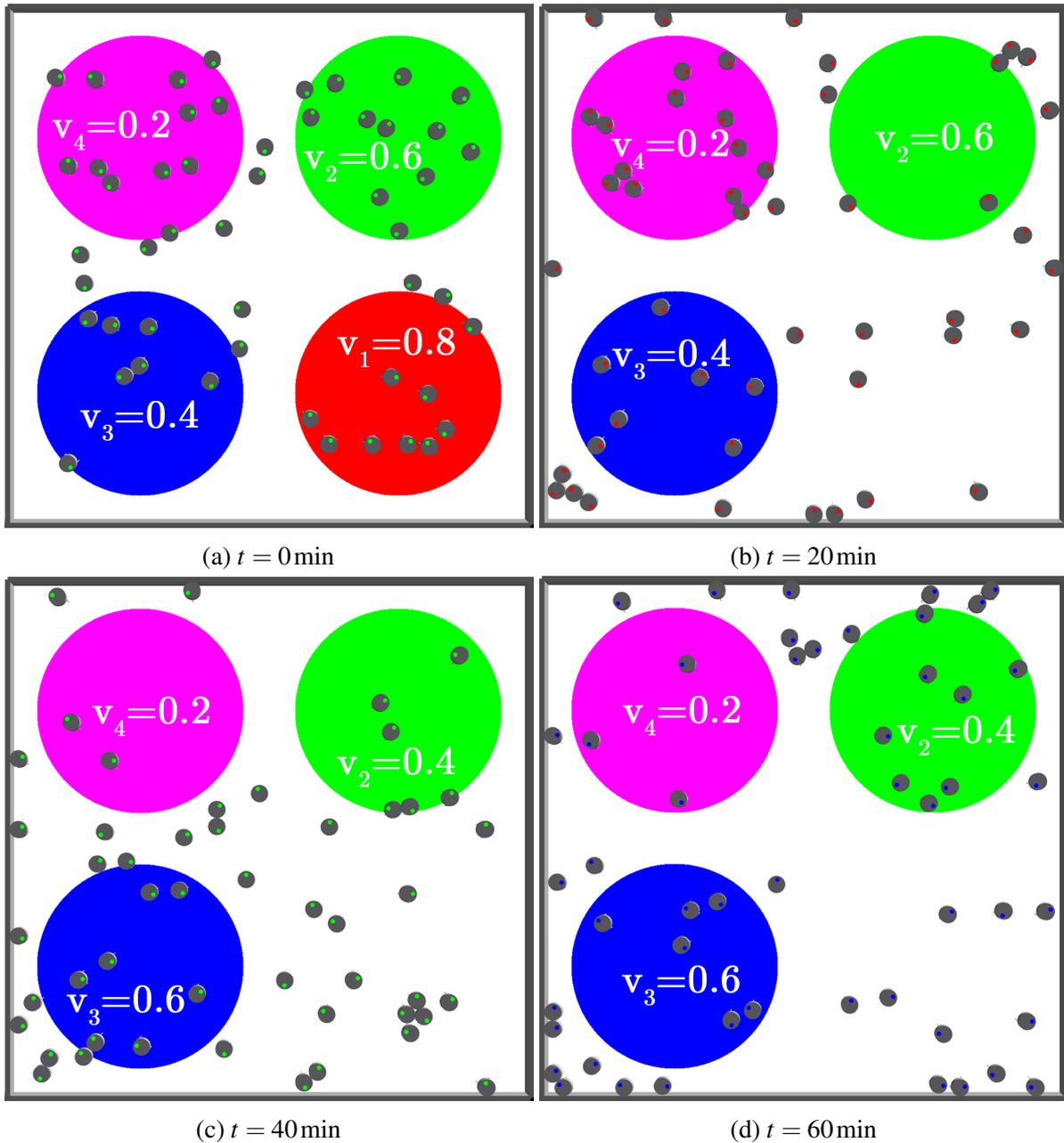


Fig. 5.16 Screenshots of the swarm robotics experiment described in section 5.5.2 conducted under ARGoS simulator. The screenshots are taken at different times throughout the experiment. Here, each robot executes the *Compare*-based decision-making model introduced in section 5.4.2.1. A video of the experiment is available at <https://youtu.be/nOU8XCe0J5Y>.

parameter k (of equation (5.3)) that prevents the robots from committing to other options with less than k quality improvement. The parameter k , when set to high enough values, prevents the robots from considering similar quality options to be better than their current option due

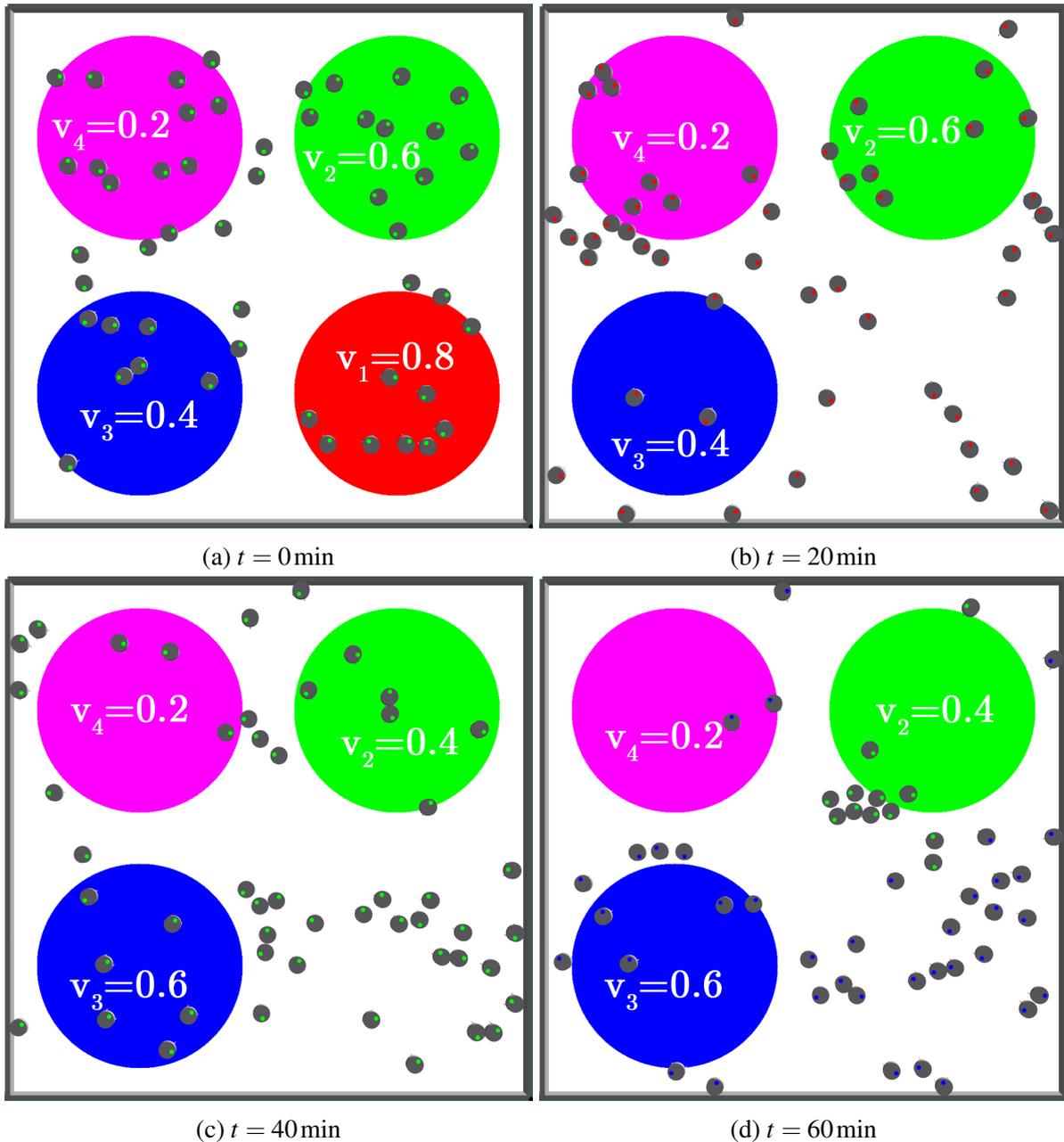


Fig. 5.17 Screenshots of the swarm robotics experiment described in section 5.5.2 conducted under ARGoS simulator. The screenshots are taken at different times throughout the experiment. Here, each robot executes the *Forget*-based decision-making model introduced in section 5.4.2.2. A video of the experiment is available at <https://youtu.be/zudCnoPRG3c>.

to estimation noise. Similar results can be achieved when using the *Forget* rule by correctly setting the parameter α (of equation (5.4)). However, controlling the response of the swarm through the parameter α is not straightforward. High values of α make the swarm unable to

a reach consensus as robots frequently forget their choice while low values make the swarm unable to adapt its decision since robots will not forget frequently enough their current choice to commit to the better option. It is important to highlight that the *Forget* rule is cognitively less demanding than the *Compare* rule as it only requires the robot to spontaneously abandon its choice [169]. In contrast, the *Compare* rule requires the robot to have the ability to memorise and compare two values.

Through multi-agent simulations, we evaluated the influence of the robot density in the environment and the robot's communication range on the adaptation abilities achieved by the swarm through the proposed behavioural rules. Counter-intuitively, the higher the density of the robots in the environment and the longer the robot's communication range, the less adaptive is the swarm especially when the difference between the quality of the new best options and the swarm's current choice is small. The reason for this is that increasing the density of the robots in the environment or the robot's communication range increases the number of peers each robot communicates with at a time. Consequently, when the swarm is settled on a choice, and a robot commits to a new option, the robot gets quickly reverted by its peers to the previous choice. A particularly interesting result is that lower communication ranges are better to achieve adaptation. This result highlights the importance of local communication that characterises swarms [76] and is in agreement with a recent study which suggests that limited cognitive and sensory capabilities are essential for the emergence of collective behaviours in animal groups [155].

Chapter 6

A simple individual behaviour for a tunable collective resource collection behaviour

In this chapter¹, we address the collective resource collection task where robots are required to find item sources in an unknown environment, collect items and transport them back to a central depot. Here, we propose a bio-inspired individual behaviour that allows robot swarms to perform the resource collection task in the case of objects of different quality. Similarly to some species of foraging ants, in the proposed individual behaviour robots coordinate their resource collection efforts using pheromone trails. Our proposed individual behaviour is highly simplified, as it is based on binary pheromone sensors. Despite being simple, the proposed individual behaviour is able to reproduce classical foraging experiments conducted with more capable real ants that sense pheromone concentration and follow its gradient. One key feature of our controllers is a control parameter which balances the trade-off between distance selectivity and quality selectivity of individual foragers. To assess the performance of the emergent collective behaviour, we employ an optimal foraging model [197] that explicitly takes account of crowding, and we compare its predictions against the results of simulations with swarms of varying sizes and experiments with up to 200 physical robots. In Section 6.1, we formalise the collective resource collection task, outline the required robot capabilities, and explain how the

¹This chapter is a modified form of a published manuscript: Talamali, M.S., Bose, T., Haire, M. et al. Sophisticated collective foraging with minimalist agents: a swarm robotics test. *Swarm Intelligence* 14, 25–56 (2020). The "Abstract" and "Introduction" sections of the manuscript has been renamed and modified to maintain consistency with other chapters. The section "Related works" of the manuscript has been modified and moved to Chapter 2. The "Kilobot robot" section of the manuscript is not included as it was already explained in Section 3.3.1. The section "An optimal resource collection model" of the manuscript has been modified to reflect that it is not part of the contributions made in this thesis. The "Appendix" section of the manuscript is not included as it is not part of the contributions made in this thesis.

ARK system (see Section 3.3.2) is used to equip the robots with the required capabilities. The proposed individual behaviour is introduced in Section 6.2. Section 6.3 presents the optimal foraging model used to assess the performance of the collective behaviour emerging from the proposed individual behaviour. Finally, the results of our tests are presented and discussed in Sections 6.4 and 6.5, respectively.

6.1 Resource collection in an unknown environment

In this section, we formally define the investigated problem and the required capabilities of the robot (Section 6.1.1), then we describe how the ARK system (see Section 3.3.2) is used to equip the robots with the necessary capabilities (Section 6.1.2).

6.1.1 The resource collection task

In this study, we investigate the problem of resource collection by a swarm composed of S robots. The environment has n circular source areas of radius 10cm, denoted by A_i with $i \in \{1, \dots, n\}$, which are scattered around a central depot. Each area A_i offers resource items of quality Q_i . The quality is a numerical indication of the importance of the resource with respect to the task that will be performed; this is similar to the nutritional value of food items in animal foraging. In this work we are interested in the foraging process at steady state, therefore, we assume sources which never deplete. If a robot enters a source area, it immediately collects one virtual item (or object) and returns it to the central circular depot (of radius 10cm). We do not take into account any handling time of the resource item. Also, we do not consider the time spent in the resource patch, as the robot immediately finds an object and returns to the depot (no exploration within the source area). The load carried back to the nest site is always one item at a time. Travelling takes place with the same speed independent of the load carried (*i.e.* either unloaded or loaded with one object). Keeping these aspects in abstract terms helps to focus the study on the collective motion aspect and allocation of robots to source areas. In fact, this study focuses on strategies to coordinate the robot motion between depot and source areas through decentralised self-organising mechanisms. In particular, we explore how indirect communication in the form of virtual pheromone trails can allow the robot swarm to balance the trade-off between the quality of resource items and the distance between the source area and the central depot.

The robots have limited computational and memory capabilities and need to operate in an unknown environment. Robots are incapable of memorising source areas' locations, instead rely on pheromone trails to find the previously discovered sources again. This form of indirect

communication requires the robots to be able to apply and read temporary marks in the environment. Additionally we assume that robots always know the direction to the depot (similarly to path integration in ants and other social insects [28, 17, 86]) and are able to detect walls in front of them. However, robots do not possess any form of direct communication amongst each other, and cannot perceive other robots in their surroundings.

6.1.2 Getting the required robot capabilities through the ARK system

In this study, we employ ARK to allow robots to apply and read virtual pheromone which evaporates and diffuses over time. We equip the Kilobots with five virtual sensors and one virtual actuator. In particular, each robot is equipped with:

- area sensor (either depot or source): the Kilobot is able to perceive if it is within the depot or a source area (this information is encoded in 2 bits);
- item quality sensor: the Kilobot is able to estimate the quality of the item it retrieves from the source area. Additionally, when the Kilobot enters in the depot, it can estimate the quality of the items that have been collected up to now (this information is encoded in 4 bits);
- depot direction sensor: the Kilobot has always knowledge about its relative direction to the depot (this information is encoded in 4 bits);
- wall sensor: the Kilobot can sense if there is a wall at a distance of ~ 5 cm in front of itself; note that this does not allow the Kilobot to sense the presence of other robots (this information is encoded in 4 bits);
- pheromone gland actuator: the Kilobot can deposit a drop of pheromone at its location (it expresses this behaviour by blinking its LED blue);
- pheromone antennae: the Kilobot can sense the presence of pheromone at a distance of ~ 3.5 cm from its centre in front of itself (this information is encoded in 4 bits, see Figure 6.1).

To store information about the pheromone, ARK models the environment as a discrete 2D matrix with cells of $6.7 \times 6.7 \text{ mm}^2$. Each time-step of length $\Delta t = 0.5$ s, ARK updates the pheromone matrix by adding pheromone deposited by the robots (each drop consists of an increment of $\phi = 250$ in the cell under the robot's centre) and computes evaporation and diffusion of the pheromone. Each matrix cell $m(i, j)$ is updated as

$$m(i, j) = m(i, j)[e^{\log(0.5)\varepsilon\Delta t} - 4\gamma\Delta t] + [m(i, j \pm 1) + m(i \pm 1, j)]\gamma\Delta t, \quad (6.1)$$

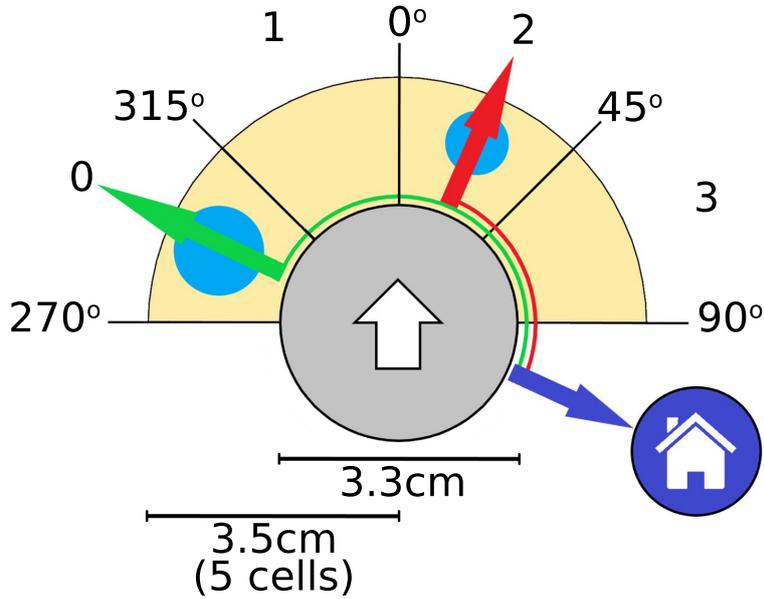


Fig. 6.1 Kilobots sense via ARK the presence of virtual pheromone in front of themselves at a distance of ~ 3.5 cm in four 45° -wide sectors. The virtual sensor indicates the presence or absence of pheromone as binary values, therefore, the Kilobot has no information about the pheromone quantity or concentration difference. In this illustration, pheromone is represented as blue circles, and thus the virtual sensor readings are $[1,0,1,0]$. When an exploring Kilobot detects pheromone, it interrupts random exploration and moves towards the detected pheromone. If more than one sector has pheromone (as in the illustration), to decide its motion direction the robot compares the sectors' direction with the depot direction (depot illustrated as a house and direction differences as red and green angles) and moves towards the largest angle (green arrow).

where the parameters $\varepsilon = 0.1$ and $\gamma = 0.02$ are the evaporation and diffusion rates, respectively. Equation (6.1) is a discrete realisation of Fick's law of diffusion [48], where we introduce the exponential term to take into account the pheromone evaporation consistently with studies from biology [56].

6.2 A simple individual behaviour for complex coordination

The individual robot behaviour is relatively simple and can be described by the Probabilistic Finite State Machine (PFSM) illustrated in Figure 6.2. The main structure of the behaviour is based on the control software designed by Font Llenas et al. [49]. The behaviour has been enriched by adding a new *Obstacle Avoidance* state (indicated as AO in Figure 6.2), by including an additional form of indirect communication that enables adaptability to different

quality scales (as described in Section 6.2.1), and by allowing for probabilistic transitions and tuneable pheromone functions (as described in Section 6.2.2).

The robots do not have previous knowledge about the number, location, and items' quality of the source areas. Therefore, a robot starts by exploring the environment to discover source areas (state RW in Figure 6.2). Due to the Kilobot's limited capabilities (see Section 3.3.1), the exploration is performed via an isotropic random walk which is a widely-used and simple method to search for targets in an unknown environment [36]. The random walk consists of alternate straight motion for 10 s and uniformly random rotation in $[-\pi, \pi]$. Upon encounter of a source area, the robot (virtually) picks up an item and transports it to the depot (state GD in Figure 6.2). As indicated in Section 6.1.1, we assume that the robots are limited in memory and only able to keep track of the direction towards a single location in the space, in our case the direction to the depot. This assumption is in line with the behaviour of several ants species which rely on path integration to return to the nest [28, 17, 86]. The robots follow the direction to depot to bring back collected items. Instead, to memorise the source locations, the robots rely on their stigmergic coordination which represents a form of collective memory. Therefore, on its way to the depot, the robot lays down virtual pheromone to allow itself, and other robots, to find the source area again. The robot, every four seconds, takes a probabilistic decision to deposit the next pheromone drop using the function $P_\phi(Q_i)$ which is a function of the collected item's quality Q_i ². The function $P_\phi(Q_i)$ is given by Equation (6.2) and described in detail in Section 6.2.2. On arriving at the depot, the robot unloads the item and probabilistically decides (according to Equation (6.3)) to turn back to follow the just-formed pheromone trail (state TB in Figure 6.2), or to interrupt its exploitation of this source area and to resume exploration through the random walk. When a robot senses virtual pheromone via the virtual antennae (composed by four sectors described in Section 6.1.2), the robot follows the trail by moving in the direction of the triggered antennae sector (state FP in Figure 6.2). If the robot senses pheromone in more than one direction, *e.g.* both left and right sectors as in the illustration of Figure 6.1, the robot compares the sensed-pheromone directions with the direction to the depot (red and green angles in Figure 6.1) and moves towards the direction with the largest difference (green arrow in Figure 6.1). This decision relies on the assumptions that robots only deposit pheromone in their straight path from a source area to the depot and that they always have access to the depot vector.

Compared with previous studies [49], the robot's behaviour has been enriched through the inclusion of obstacle avoidance (state AO in Fig 6.2). In fact, robots have been equipped with a virtual sensor to detect walls (see Section 6.1.2). The robot reacts to a wall only if sensed in

²*Lasius niger* ants follow a similar behaviour, laying pheromone trails on their way back to the nest while depositing a quantity of pheromone proportional to the quality of the foraged food [147, 31].

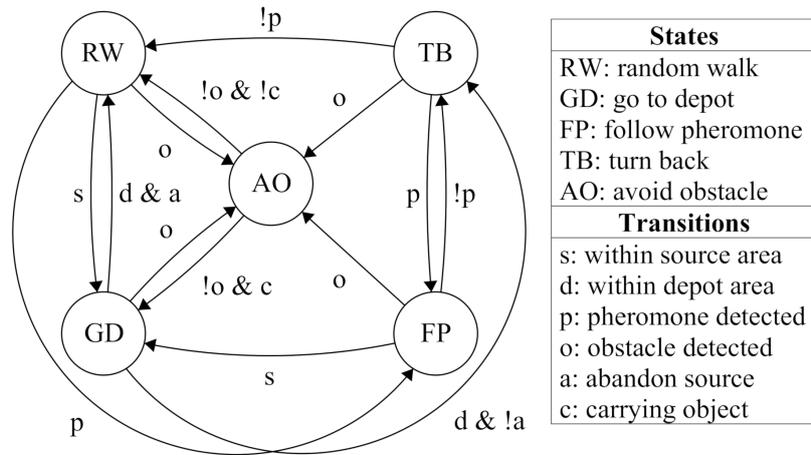


Fig. 6.2 Probabilistic Finite State Machine (PFSM) of the individual robot behaviour. Circles represent states and arrows are transitions. Robots start exploring the environment through a random walk (RW); when they find a source area, they collect an item and return to the depot (GD) laying pheromone according to Equation (6.2). Once arrived at the depot, they either turn back (TB) or resume exploration (RW). When explorer robots detect pheromone, they follow it (FP). When robots detect a wall, they avoid it (AO). Controlling individuals through this simple PFMS leads to sophisticated collective foraging dynamics.

a frontal position, *i.e.* the two central sectors in the range $[-45^\circ, 45^\circ]$ of the robot's heading (note that the virtual wall sensor is composed by four sectors equal to the virtual antennae of Figure 6.1). Upon wall detection, the robot turns left or right for about 22.5° in the opposite direction of the sensed obstacle, then moves straight for 2.5 s, and finally returns to either the random walk (RW) state or the go to depot (GD) state, depending on whether it carries an item or not. This behaviour may be triggered multiple times until no obstacle is sensed in the central sectors. In the case of symmetric sensing, *i.e.* both central sectors sense an obstacle, the robot uses as tie-breaker the lateral obstacle sectors to turn in the freest direction. In the case of complete symmetry, the direction is selected at random.

6.2.1 Adaptivity to relative quality differences

The robots do not have any prior information about the range of the sources' qualities that the unknown environment can offer. In order to allow the swarm to tune its behaviour to an unknown quality range, the individual robots update over time their knowledge on the best currently available quality Q_{max} . Initially, the robot has no prior knowledge about the quality range and thus ranks the first source it finds as the best available. Over time, the robot constantly compares its range (*i.e.* the best available quality Q_{max}) with other items collected by other swarm members. The communication between robots is indirect and takes place within the

depot. Each time a robot enters the depot, it can see the qualities of the items collected by the swarm until now; thus, the robot compares its information with the best quality and, if higher, updates its Q_{max} accordingly. This mechanism is consistent with animal behaviour where individuals can assess the nutrient quality of the swarm's reserves and compare against their own [44, 4].

In our study, we consider unlimited item sources to investigate the steady-state regime, however, in case of limited sources (*i.e.* with a limited number of items) the robots may update their quality range by only observing the latest collected items. In this way, we predict the swarm being able to flexibly adapt to appearance or depletion of sources.

6.2.2 Modulation of the individual rules to obtain a plastic behaviour

After collecting an item, the robot returns to the depot laying a pheromone trail. The pheromone trail acts as a form of indirect communication between robots which inform each other about paths connecting depot to discovered sources. Collective contribution to these trails leads to a form of swarm memory which allows the swarm to remember the location of sources in the environment. In fact, our simple robots cannot internally store sources' locations, although the swarm, as a whole, can remember locations through pheromone trails. A pheromone trail is formed by a sequence of drops that the robot deposits via its virtual pheromone gland (see Section 6.1.2). Similar to the approach of [49], a robot probabilistically decides every four seconds whether to lay the next drop or not. In the previous work, we implemented a simple linear function to map the quality Q_i into a pheromone deposition probability, *i.e.* $P_\phi(Q_i) = Q_i/Q_{max}$. Linking the pheromone deposition function to perceived source quality allowed the swarm to give priority to better quality sources over inferior sources.

In this study, we implement a tuneable function to allow the robot to regulate its selectivity on the quality through a single parameter $\alpha \geq 0$. The probability to deposit the next pheromone drop is given by

$$P_\phi(Q_i) = e^{\alpha(Q_i - Q_{max})Q_i^{-1}}. \quad (6.2)$$

The individual robots have access to α in a decentralised way and can alter this value to vary the global response. Using an $\alpha > 1$, the function has an exponential shape on Q_i resulting in high selectivity in favour of the highest quality sources. A value of $\alpha \approx 1$ leads to (approximately) linear response, therefore, approximating the function investigated in [49], thus having Equation (6.2) as a generalisation of the previous specific function. Finally, decreasing $\alpha < 1$ gradually flattens out the function to a constant value, that at the limit of $\alpha = 0$ becomes constant $P_\phi(Q_i) = 1$; this results in constant pheromone trails irrespective of the sources' qualities.

To further expand the individual robot capabilities to be able to balance the distance-quality trade-off, we introduce a decay function $P_d(t_i)$ that robots use, upon arrival in the depot with an item (event indicated with the letter ‘a’ in Figure 6.2), to decide whether to keep exploiting the same source or to start exploring for new sources. $P_d(t_i)$ is inspired by similar abandonment behaviours observed in social insects (*e.g.* foraging ants [182] and house-hunting honeybees [181]) and allows the robots to abandon exploiting source A_i that required a long travel time t_i (either because it is distant or has an overcrowded path). The travel time t_i is measured by the robots as the time spent between the item collection (from the source A_i) and the item deposition (in the depot). The function $P_d(t_i)$, similarly to $P_\phi(Q_i)$ of Equation (6.2), is modulated by the parameter α as

$$P_d(t_i) = (\alpha + 1)^{-2} e^{\frac{t_i - t_{max}}{(\alpha + 1)\sqrt{t_i}}} \quad (6.3)$$

where t_{max} is a parameter indicating the robot’s prior knowledge on the maximum acceptable time to return from a source area. The t_{max} could be adaptively tuned (similarly to Q_{max} in Section 6.2.1), although in this study we do not explore this aspect and we fix $t_{max} = 100$ s. Assuming a fixed t_{max} is reasonable because in both biological and artificial systems source areas may be accepted only if they are located within a certain maximum distance (or travel time t_i) from the depot that is decided a priori.

Equations (6.2) and (6.3) are linked by the parameter α which the robots can regulate to alter the swarm behaviour. Increasing $\alpha > 1$ has the combined effect of increasing discriminability on quality Q_i and flattening $P_d(t_i) \approx 0$ for any distance; therefore, the swarm ignores distance but selects the higher quality source. Conversely, small $\alpha < 1$ flattens out quality differences $P_\phi(Q_i) \approx 1$ and accentuates differences on travel time with an exponential abandonment $P_d(t_i)$ on high travel times; this leads to a system where the only discriminating factor on source selection is distance due to a combination of evaporation and abandonment on farther sources. Finally, intermediate values $\alpha \approx 1$ give a quasi-linear response of $P_\phi(Q_i)$ and sublinear $P_d(t_i) > 0$ which allow the swarm to balance the distance-quality trade-off (similarly to what has been reported in [49]).

6.3 Optimal foraging model

To assess the performance of the collective resource collection behaviour emerging from the individual behaviour introduced in Section 6.2, we employ the optimal foraging model proposed in [197]. This model gives a mathematical description of the optimal resource collection behaviour based on principles of optimal foraging theory [98, 93]. The model describes the utility gained by the collection of resource items discounted by the cost incurred

in transporting these items to the depot. The main components of our model are the items' qualities, the allocation of robots to various source areas, and the source-depot travel time. We model the robot allocation as ρ_j (with $j \in \{1, \dots, n\}$) which is the fraction of robots on the trail between the central depot and source area A_j . All robots that are actively involved in the transportation of items from the n sources are called *workers*; their fraction is denoted by $\rho_w = \sum_{j=1}^n \rho_j$. The remaining robots that explore the landscape are called *explorers*, their fraction is denoted by $\rho_e = 1 - \rho_w$. The travel time is a function of the source-depot distance and of the traffic congestion on the path. In fact, crowded paths lead to frequent collisions between robots and result in longer travel times. The full details of the model and its derivation are given in [197]; here we report the main quantity which is the swarm yield R , defined as

$$R = \sum_{j=1}^n \frac{q_j \beta_j \rho_j S}{\tilde{d}_j^2}, \quad \text{with} \quad \tilde{d}_j = d_j + v_o T_{C,j}(\rho_j S). \quad (6.4)$$

where S is the swarm size, $q_j = Q_j/Q_{\max}$ is the normalised quality of source area A_j , ρ_j is the fraction of robots on the trail between the central depot and source area A_j .

The parameter β_j in Equation (6.4) is a fitting parameter characterising the proportionality relationship between the number of collected items from source A_j and the number of robots on the trail to A_j at equilibrium given as

$$\Delta U_j = \varphi_j (T_2 - T_1) \beta_j \rho_j S, \quad (6.5)$$

where T_2 and T_1 (with $T_2 > T_1$) are times at which the swarm is already in a steady-state, and φ_j is a foraging rate, which is approximated by the inverse of the round-trip travel time between central depot and source A_j (for details see [197]).

The term d_j in Equation (6.4) is the distance between source area A_j and depot, $v_o = 1 \text{ cm/s}$ is the Kilobot's speed, and the function $T_{C,j}(\rho_j S)$ models the additional travel time arising from traffic congestion. The Equation (6.4) models traffic congestion as an increase of the travel distance d_j by accumulating the additional length of $v_o T_{C,j}(\rho_j S)$. The function $T_{C,j}(\rho_j S)$ is defined as

$$T_{C,j}(\rho_j S) = T_{0,j} \left(\exp \left[\kappa_j \frac{\rho_j S}{N_{\text{crit},j}} \right] - 1 \right), \quad (6.6)$$

where $T_{0,j}$ is a constant which sets the time scale of the additional travel time, κ_j is a constant included to fine-tune the nonlinear effect of overcrowding on the path to A_j , and $N_{\text{crit},j}$ is the critical number at which traffic congestion may have a significant effect. This means that $T_{C,j}(\rho_j S)$ is negligible if $\kappa_j \rho_j S \ll N_{\text{crit},j}$.

6.3.1 Estimation of model parameters from simulation data

As for the model proposed in [197] and introduced by Equations (6.4)-(6.6), three free parameters per source area ($T_{0,j}$, β_j , and κ_j , with $j \in \{1, \dots, n\}$) need to be estimated from data. To do so, we use the relationship between the number of robots on a path and the number of collected items given in Equation (6.5) (with $T_2 = 60$ min and $T_1 = 30$ min). For the case of two source areas, the results of fitting are depicted in Figure 6.3 and summarised in Table 6.1. As shown in Figure 6.3, for small-to-medium numbers of robots on a trail, the number of collected items per time interval increases linearly with the number of robots on a trail; whereas for medium-to-large numbers of robots on a trail, we observe a nonlinear decay. This type of curve is widespread in several natural and artificial systems and is often indicated as the *Universal Scalability Law* [70, 104, 78, 74, 75].

task condition	β_1	β_2	$T_{0,1}$ (s)	$T_{0,2}$ (s)	κ_1	κ_2	$R_{\text{GoF},1}^2$	$R_{\text{GoF},2}^2$
$q_1 = 1, q_2 = 0.5,$ $d_1 = d_2 = 0.6$ m	1.035 (0.004)	1.009 (0.004)	0.180 (0.019)	0.150 (0.015)	1.483 (0.027)	1.586 (0.026)	0.985	0.991
$q_1 = q_2 = 1,$ $d_1 = 1$ m, $d_2 = 0.5$ m	0.951 (0.004)	1.091 (0.004)	0.0004 (0.0001)	0.805 (0.046)	3.692 (0.110)	0.991 (0.012)	0.968	0.990
$q_1 = q_2 = 1,$ $d_1 = d_2 = 1$ m	0.961 (0.003)	0.968 (0.003)	0.026 (0.004)	0.032 (0.005)	2.327 (0.042)	2.315 (0.039)	0.984	0.985

Table 6.1 Overview of estimated model parameters. The goodness-of-fit is quantified by $R_{\text{GoF},j}^2 = 1 - \sum_i (y_i - y_i^{\text{fit}})^2 / \sum_i (y_i - \bar{y})^2$, where $y_i = \Delta U_i / \text{min}$, the y_i^{fit} correspond to the fitted values, and \bar{y} represents the mean value of all y_i . The index j corresponds to the trail. Mean model parameter values, including one standard deviation errors (values in brackets), are given.

6.4 Results

Through physics-based simulations, we systematically tested a variety of experimental conditions to study the performance of the proposed system. We validated some of the simulation results through experiments with up to 200 physical Kilobots. In Section 6.4.1 we present a set of simulation results that highlight the benefits of having introduced a virtual wall sensor, adaptability to unknown environmental scenarios, and behaviour modulation to balance the distance-quality trade-off. In Section 6.4.2 we compare the model predictions against robot swarm simulations for different swarm sizes.

The physics-based simulations were conducted with ARGoS [139, 138] which is a state-of-the-art swarm robotics simulator that accurately and efficiently simulates the Kilobots

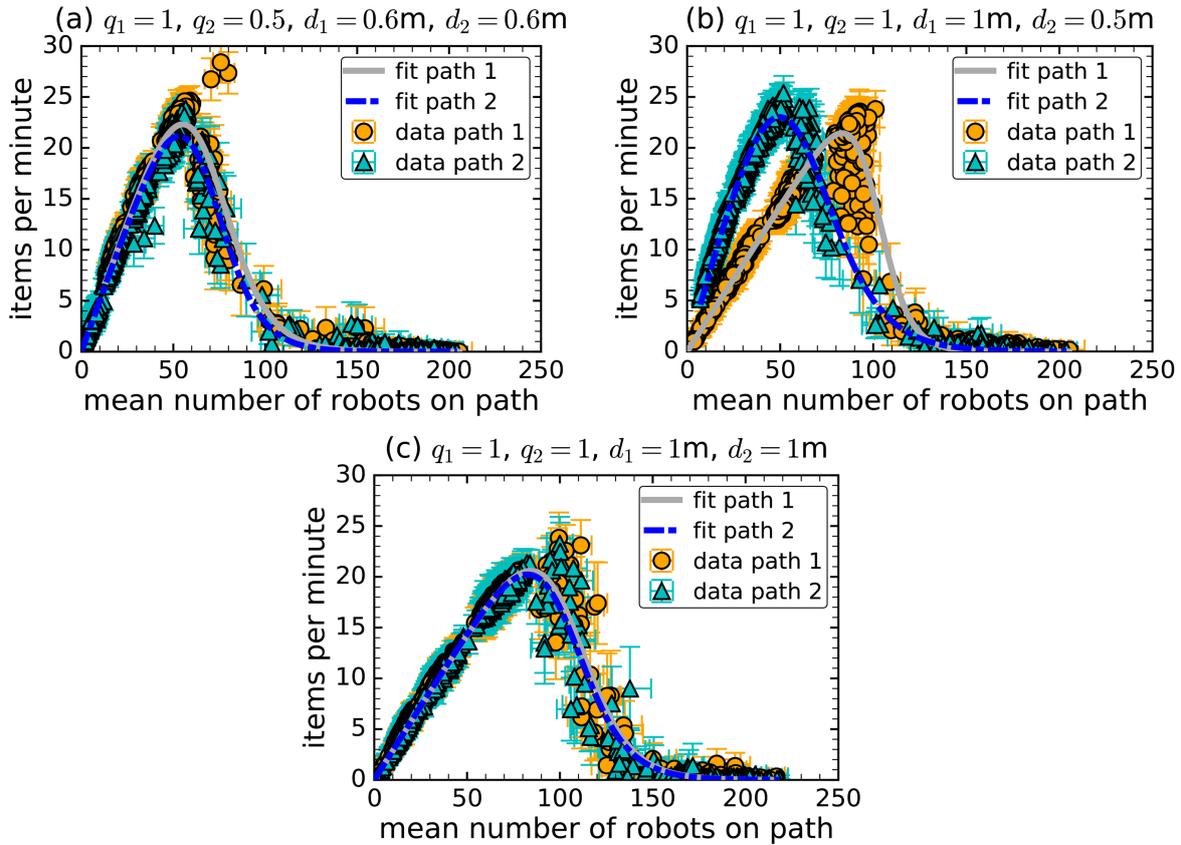


Fig. 6.3 Fits of Equation (6.5) to data generated by physics-based simulations in order to obtain the model parameters reported in Table 6.1. Fitting is performed in the case of $n = 2$ source areas with different quality and equal distance in panel (a), equal quality and different distance in panel (b), and equal quality and distance in panel (c). Data points are represented using symbols and fits are represented using lines (circles and solid grey lines show collection from source A_1 while triangles and dash-dotted blue lines show collection from source A_2). Error bars represent 95% confidence intervals. There is a linear growth for small-to-medium numbers of robots on a path, and a nonlinear decay for medium-to-large numbers of robots on a path. This type of growth-decay curve on population size is widespread in nature [104] as in engineering [70].

and the ARK system via a dedicated plug-in [138]. The physical robot experiments were run with fully charged Kilobots whose motors have been automatically calibrated through ARK [160]. The videos of these experiments are augmented by superimposing the virtual environment information (see a sample image in Figure 6.4) and available as online supplementary material (Online Resource 1-9) and at <https://www.youtube.com/playlist?list=PLCGKY9OHLZwMaGeB6cxVfxmHwhBFqKF7a>. The robot simulation code is open source and available online at <https://github.com/DiODEProject/PheromoneKilobotSwarmIntell>.

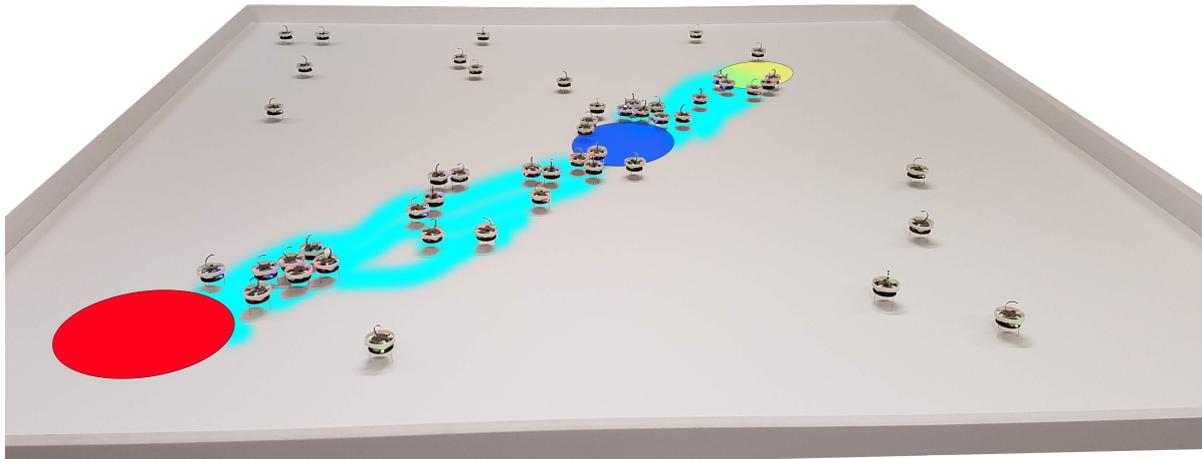


Fig. 6.4 (Colours online) A picture of a 50 real Kilobots experiment with the virtual environment superimposed on the image. The red (bottom-left) source area A_1 has quality $Q_1 = 10$, while the yellow (top-right) source area A_2 has quality $Q_2 = 4$. The sources are placed at $d_1 = 1$ m and $d_2 = 0.6$ m from the central (blue) depot. The (light-blue) shades represent the pheromone trails that the robots deposit and follow. Full videos are available at <https://www.youtube.com/playlist?list=PLCGKY9OHLZwMaGeB6cxVfxmHwhBFqKF7a>.

6.4.1 Results show tuneable and adaptive swarm responses

We report here the simulation and physical robot results to show evidence of the behaviours obtained through obstacle avoidance, adaptivity, and individual function modulation.

Obstacle avoidance Figure 6.5(b) shows a screenshot of an experiment inspired by the well-known study of Goss *et al.* [66] which showed that ants are able to exploit the shorter path in double bridge experiments with branching paths of different lengths. In our system, the individual robots have lower cognitive capabilities than the individual ants. In fact, the Kilobots cannot distinguish pheromone intensity, follow its gradient, nor make decisions with respect to differences in pheromone concentration. Nevertheless, the robot swarm was able to preferentially exploit the shorter path. This outcome was not limited to conditions where the pheromone evaporation was too high to exploit the longer path while sufficient to establish a path on the shorter, but it also applied to scenarios in which both paths were viable. In fact, we tested the swarm in an environment where we blocked the shorter path and only the longer path was active (see Figure 6.5(a)) and the robots exploited the longer path, as illustrated in the plot of Figure 6.5(c). Similar double-bridge experimental setups have been emulated and investigated in previous swarm robotics studies such as [122, 177], in which, however, the swarm's behaviour and goal were different.

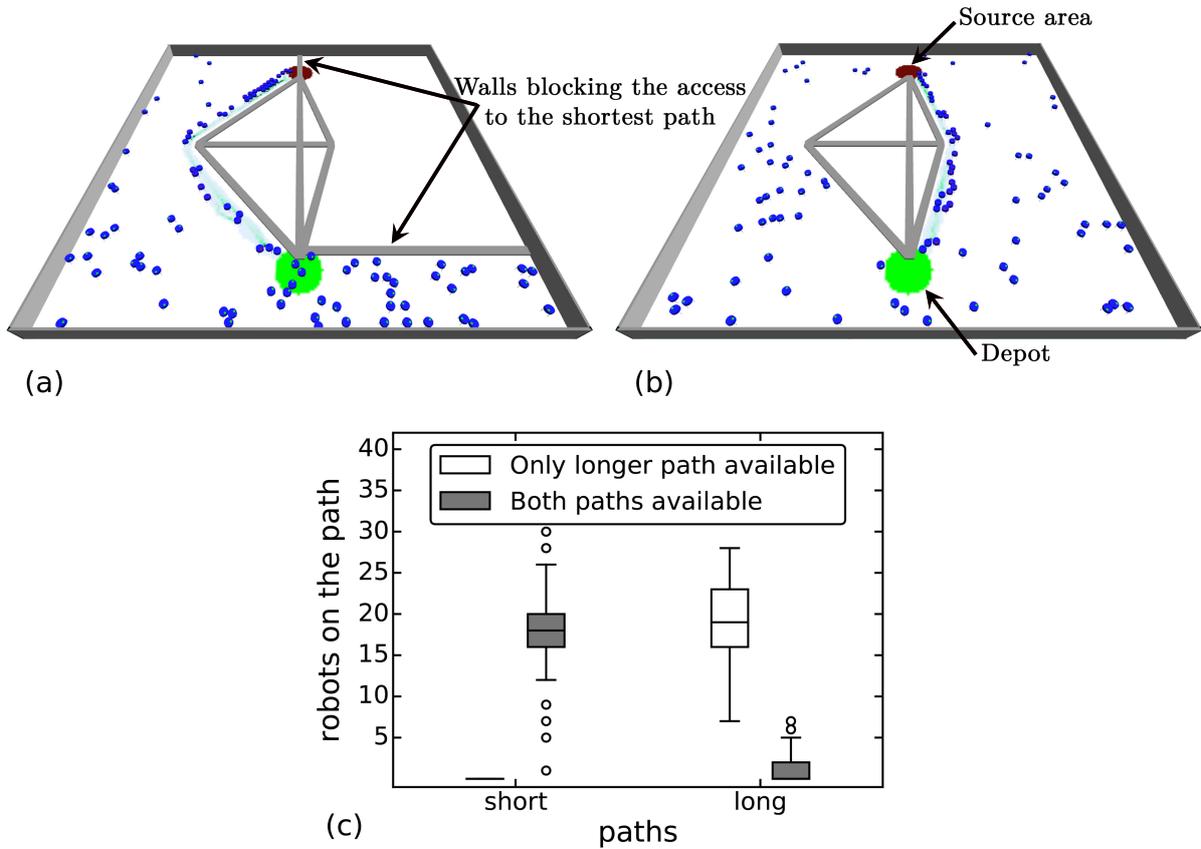


Fig. 6.5 A 50 simulated Kilobot swarm experiment inspired by the ants' double-bridge experiment by [66] in which two paths, a longer path (1.8 m long) and a shorter path (1.4 m long), connected source to depot. When the simulated swarm had access to only the longer path, panel (a), the Kilobots reinforced pheromone on that path and used it for their collections. Instead, when both paths were available, panel (b), the Kilobots disregarded the longer path and (almost exclusively) used the shorter for their collections. Panel (c) shows the number of robots on the two paths at the end of one simulated hour (boxes range from 1st to 3rd quartile of the data from 100 simulations and indicate the median with a horizontal line, the whiskers extends to 1.5 IQR). The individual Kilobots cannot follow a pheromone gradient nor detect any difference in pheromone concentration. Despite their limited individual capabilities, the robot swarm shows (in certain experimental conditions) behaviour similar to ants' colonies, which instead rely on much higher cognitive abilities at the individual level.

Our results indicate that, for certain types of experimental conditions, cognitively simpler individuals would suffice to reproduce the collective level behaviour observed in colonies of more complex ants. However, we believe that the ants, exploiting gradient sensing, are more flexible and can optimise path lengths in a larger range of environments than our robotic system. In fact our results may vary if we would increase the robots density and/or vary the paths' lengths. However, we cannot ascribe the observed behaviour to the manually tuned

maximum travel time $t_{max} = 100$ s of Equation (6.3) because our experiments were conducted with $\alpha = 10$ which flattens Equation (6.3) to zero for every path length. Therefore, the observed dynamics emerged from a more complex interplay between the Kilobots' behaviour and the virtual pheromone dynamics, and resulted in an efficient swarm selection of the shortest path.

Adaptivity As described in Section 6.2.1, the swarm is able to adapt to any quality range and have a response that only considers the ratio between qualities rather than the absolute quality values. Figure 6.6 shows the system's response to three scenarios with $n = 2$ sources with the same quality ratio (*i.e.* $Q_2/Q_1 = 0.4$) but different absolute quality values (*i.e.* $Q_1 = 15$, $Q_2 = 6$ on the left, $Q_1 = 10$, $Q_2 = 4$ in the centre, and $Q_1 = 5$, $Q_2 = 2$ on the right of the x-axes of Figs. 6.6(a)-(b)). The results show that the adaptive strategy (white boxplots) adapted to any condition and, as the quality ratio remained the same, also the swarm response remained the same. Instead, the constant range strategy (dark boxplots) reckoned with absolute quantities and led to the desired outcome only when the prior knowledge on the quality range matched the environment's range (central experimental scenario of Figure 6.6). The ability to respond to the relative quality of food sources, rather than to an absolute quality range, has been recently documented also in foraging ants [216].

Behaviour modulation Via Eqs. (6.2) and (6.3), the individual robots can modulate their behaviour to give priority to closer (low α) or better-quality (high α) source areas. This modulation at the individual level translates to different collective responses at the swarm level. We investigated such dynamics in swarms of $S = 50$ Kilobots operating in an $n = 2$ sources scenario environment with a superior source area A_1 at distance $d_1 = 1$ m with $Q_1 = 10$ and an inferior source area A_2 with $Q_2 = 4$ and varying distance $d_2 \in [0.5, 1]$ m. The relatively small swarm size was motivated by preliminary results that we reported in [49] which showed that large swarms do not discriminate between sources as there are enough robots to maximally exploit any area. Figure 6.7 shows the effect of the three tested values of $\alpha \in \{0, 0.85, 10\}$ on the swarm dynamics. Using $\alpha = 0$ promoted distance selectivity, in fact, the simulated swarm had the highest item collection per minute (panel (a)) from the closest source (A_2) to which the majority of the workers was deployed (panel (b)). Using $\alpha = 10$ promoted quality selectivity, in fact the simulated swarm had the highest item collection per minute from the highest-quality source (A_1) to which the majority of the robots was deployed. Finally, intermediate values of α , *e.g.* $\alpha = 0.85$, led to a distance-quality trade-off where the swarm exploited the nearest inferior-quality source only if it was much closer than the farther superior-quality source.

We ran three experiments with 50 physical robots for each of the two limit cases of quality-selective $\alpha = 10$ (solid black symbols) and of distance-selective $\alpha = 0$ (solid light-grey

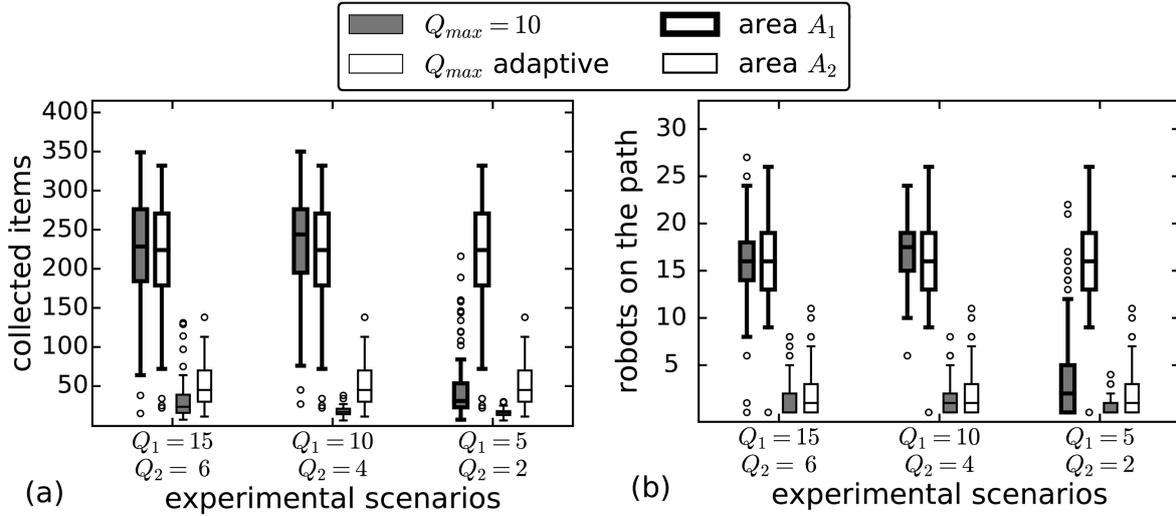


Fig. 6.6 Simulation results showing the adaptivity of the system. We measured the number of collected items in panel (a) and the number of robots on each path in panel (b) for the two source areas, the superior A_1 and inferior A_2 , both at equal distance $d_1 = d_2 = 1$ m. We kept the same quality ratio, *i.e.* $Q_2/Q_1 = 0.4$, but varied the absolute value of the objects (indicated on the x-axis). All experiments were conducted with swarms of $S = 50$ Kilobots and an intermediate value of $\alpha = 0.85$ in Equation (6.2) and Equation (6.3). Boxes range from 1st to 3rd quartile of the data from 100 simulations and indicate the median with a horizontal line; the whiskers extend to 1.5 IQR. Having a constant range (dark boxplots) shows good results only if the predefined range matches the actual range of the environment (central experiment). Instead, an adaptive strategy allows the swarm to exploit resources as a function of their relative qualities in a range adapted to the environment.

symbols). The videos of these six experiments are available at <https://www.youtube.com/playlist?list=PLCGKY9OHLZwMaGeB6cxVfxmHwhBFqKF7a>. Physical robots showed a resource collection less efficient than simulation; despite this, in both cases, the two strategies favoured either the best-quality or the nearest source area, as shown by the simulations. We explain the observed difference between reality and simulation (the reality gap) as a motion speed difference between robots and simulation. In fact, the simulation was accurately tuned on the movement of fully charged Kilobots [138], but did not take into account that the robot's speed was reduced over time due to the decrease of its battery level.

Figure 6.7(c) shows the rate per minute of collected items weighted by their normalised qualities ($q_1 = 1.0$ and $q_2 = 0.5$). We did not include any cost because in our experiments every robot moved constantly and continuously (either as worker or as explorer). Therefore the swarm incurred a constant cost independent of the collections (this would be different if, as ants do, some individuals would stop exploration to save energy [23], or to avoid overcrowding

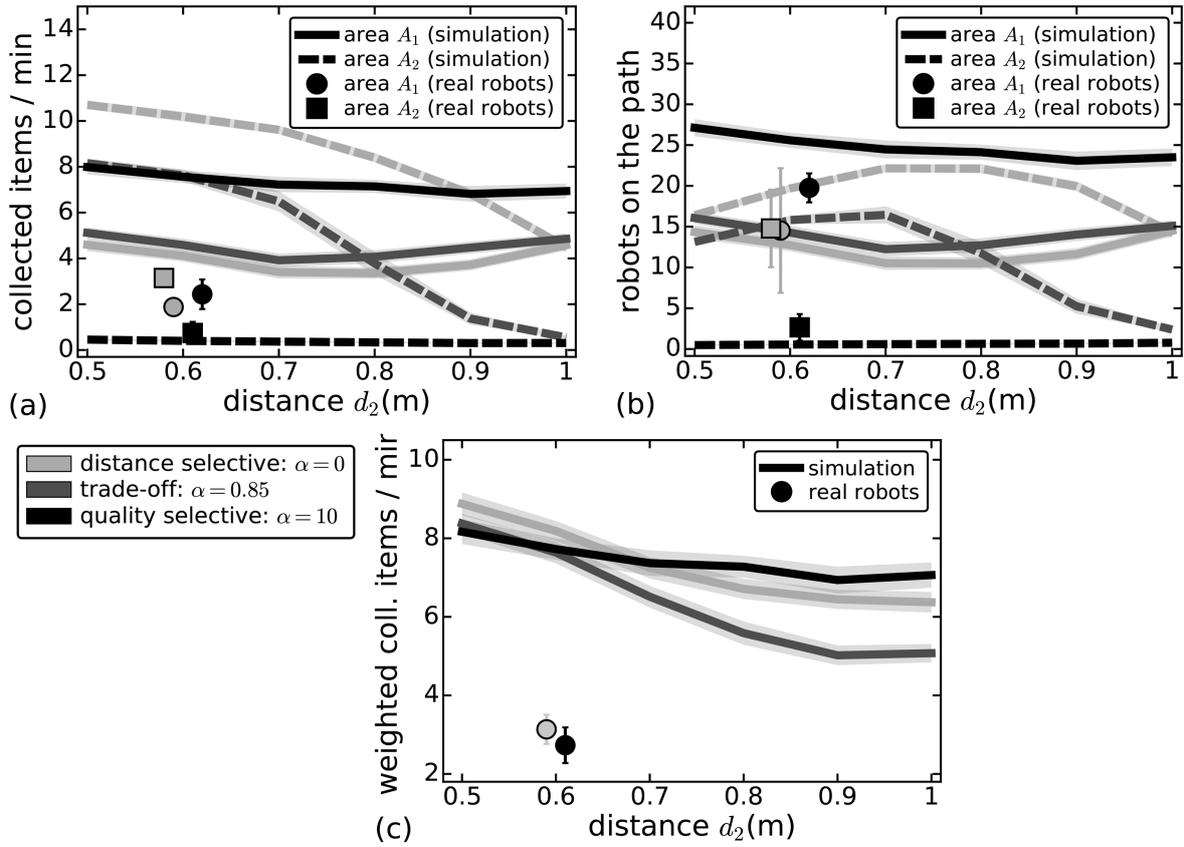


Fig. 6.7 Effect of the modulation of the parameter α from Eqs. (6.2) and (6.3) to favour nearer source areas ($\alpha = 0$), to favour the best-quality sources ($\alpha = 10$), or to balance the distance-quality trade-off ($0 < \alpha < 10$). Results of $\alpha = 0$ are shown in light-grey, $\alpha = 0.85$ in dark-grey, and $\alpha = 10$ in black. We report the results for simulations and physical robots experiments of one hour each in scenarios with $n = 2$ sources. We excluded the initial exploration phase and indicate mean values for the last 30 minutes. Physical robots results are indicated as solid symbols with vertical bars indicating the 95% confidence intervals of 3 runs for each condition (the symbols are slightly shifted to avoid bar overlaps but all represent results for $d_2 = 0.6$ m). Lines represent the mean of 100 simulations (shaded areas are 95% confidence intervals). Source A_1 had quality $Q_1 = 10$ and was located at distance $d_1 = 1$ m; source A_2 had quality $Q_2 = 4$ and varying distance $d_2 \in [0.5, 1.0]$ m. We report the rate of collected items per minute in panel (a), the mean number of robots on each path in panel (b), and the rate per minute of collected items weighted by the normalised quality $q_1 = 1.0$ and $q_2 = 0.5$ in panel (c). Individual robots can locally modulate the decentralised parameter α to lead the swarm to a range of different collective responses, *e.g.* selecting almost exclusively the best-quality source (high α) or balancing the distance-quality trade-off (low α). Physical robots are less efficient than simulations, however ordering between sources is preserved; this confirms the effects of α -modulation observed in simulation.

as discussed above). Interestingly, the results show that there was not one α -value that was better than all others; rather the best strategy varied in relation to the environment. For large distance difference, *i.e.* $d_2 \ll d_1$, the distance-selective strategy ($\alpha = 0$) displayed the highest weighted collection. Conversely, for similar distances, the best strategy consisted of favouring the best-quality source ($\alpha = 10$), analogously to what has been observed in some species of ants which focused their foraging efforts on the richer of two equally-distant sugar sources [9, 182].

6.4.2 Comparison of model and simulation data

Here we compare the performance of binary resource collections for varying swarm sizes S and varying α which regulates the swarm strategy (as from pheromone deposition in Equation (6.2) and trail abandonment in Equation (6.3)). The plot in Figure 6.8 shows the yield R as a function of the fraction of workers allocated to source A_1 (with $\rho_1 = \rho$) divided by the fraction of total workers involved in resource collection ρ_w , and of the number of worker robots $\rho_w S$ (*i.e.* involved in collecting resource items).

Best performing swarms have an intermediate size (*i.e.* $S = 200$). Relatively small swarms allocate robots more selectively depending on the implemented strategy. For instance, in Figure 6.8(a), the quality selective strategy ($\alpha = 10$ indicated as triangles) shows an allocation of workers predominantly to the best-quality source ($\rho/\rho_w > 0.8$) when $S \leq 200$. Instead, large swarms of $S = 500$ do not discriminate between sources and equally exploit both. The distance selective strategy ($\alpha = 0$ indicated as circles) in Figure 6.8(b) has a much smaller deviation and is visible only for the smallest swarm. Observing such a change in the swarm response is not an obvious result because robots cannot perceive each other. The observed change is an emergent property.

In general, simulations and the model show differences especially for swarms of size $S = 500$. In fact, for large swarms, the model predicts that the best strategy would be to allocate only a limited number of robots to the best path, in order to avoid overcrowding. We suggest that it would be possible to implement such a strategy by allowing the robots to sense and perceive peers (whilst they do not in this study). In the current strategy, we tried to overcome overcrowding by including the trail abandonment function of Equation (6.3), although this did not demonstrate sufficient ability to deviate from a symmetric exploitation for large swarms. The resulting dynamics for $S = 500$ are an equal split between the two paths (Figure 6.8(a)), which could be caused by physical ‘pushing’ between individuals, similarly to what is observed in some experiments of ants’ traffic organisation [43, 42, 50].

To investigate how collisions between individuals affect the collective dynamics, we reproduced the results of Figure 6.8 in the collision-free case in which we removed any effect of physical interactions between robots. Figure 6.9 reports the model results with null traffic

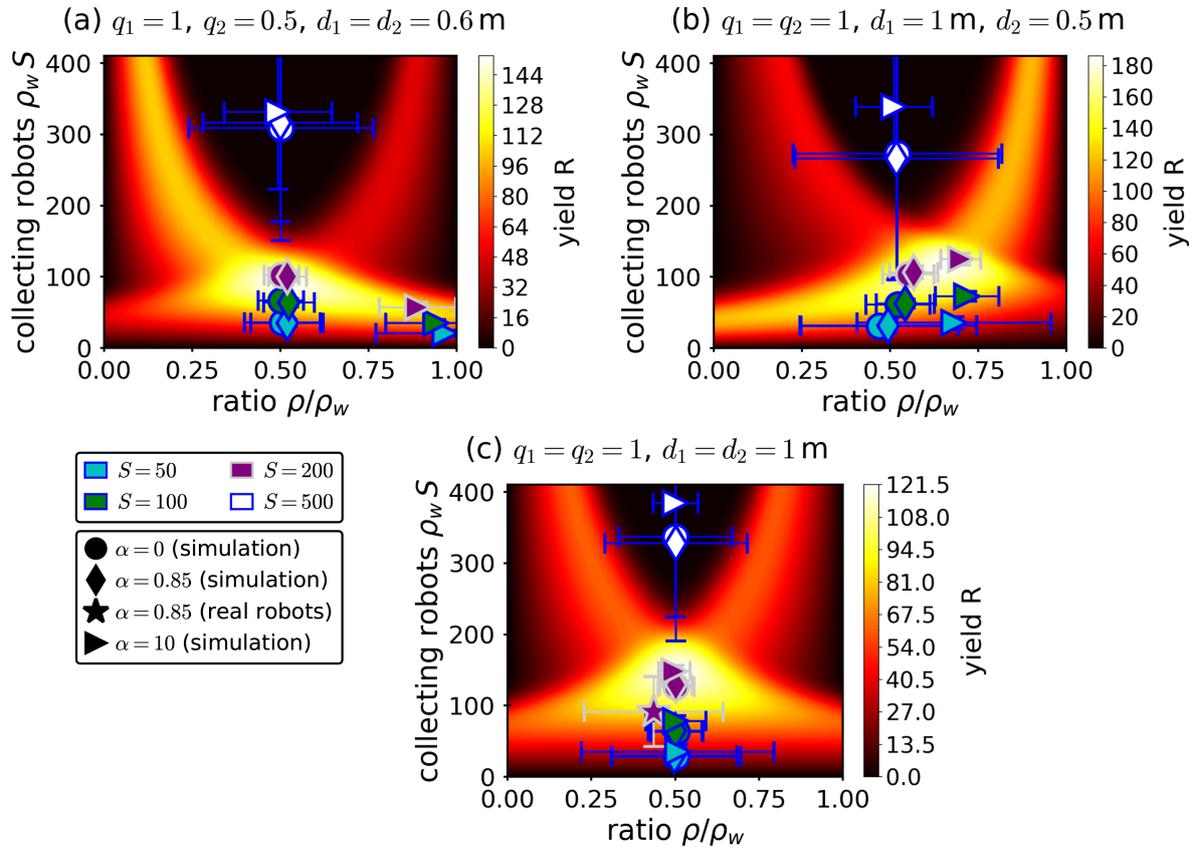


Fig. 6.8 (Colours online) Comparison of model with simulations and experiments: Total yield R as a function of the normalised swarm allocation ρ/ρ_w and the number of worker robots $\rho_w S$. We report the predicted yield R from the model of Equation (6.4) as a colour heatmap and we overlay robot simulations for three strategies: distance-selective $\alpha = 0$ (circles), distance-quality trade-off $\alpha = 0.85$ (diamonds), and quality-selective $\alpha = 10$ (triangles). We report simulations for swarm sizes $S = 50$ (cyan), $S = 100$ (green), $S = 200$ (purple) and $S = 500$ (white). Under the model’s assumptions, the simulated robot swarm performs best for $S = 200$ and $\alpha = 0.85$ ($R = 150.6 \text{ m}^{-2}$) in (a), $\alpha = 10$ ($R = 177.1 \text{ m}^{-2}$) in (b) and $\alpha = 10$ ($R = 120.4 \text{ m}^{-2}$) in (c). Swarms of large size ($S = 500$) do not achieve good performance as they equally exploit both sources and do not avoid overcrowding. The star symbol in (c) was obtained from three experiments with 200 Kilobots assuming $\alpha = 0.85$ (see online videos). Error bars represent 95% confidence intervals. Parameters: $\beta_j, T_{0,j}$ and κ_j are given in Table 6.1.

congestion contribution, *i.e.* Equation (6.6) becomes $T_{C,j}(\rho_j S) = 0$. We overlay the simulation results with deactivated collisions, *i.e.* the Kilobots’ physical body is not simulated and robots can move through each other.

As expected, the model predicts that for every workers size, $\rho_w S$ the best strategy is always to allocate all workers to the best quality source (Figure 6.9(a)), or to the closest source (Figure 6.9(b)). Some of the simulations approximate such an optimal behaviour. In the case

of asymmetric qualities (Figure 6.9(a)), the quality-selective strategy ($\alpha = 10$ represented as triangles) has high values of ρ . Similarly, the closer area in Figure 6.9(b) is largely exploited by distance-selective strategies ($\alpha = 0$ represented as circles and $\alpha = 0.85$ represented as diamonds).

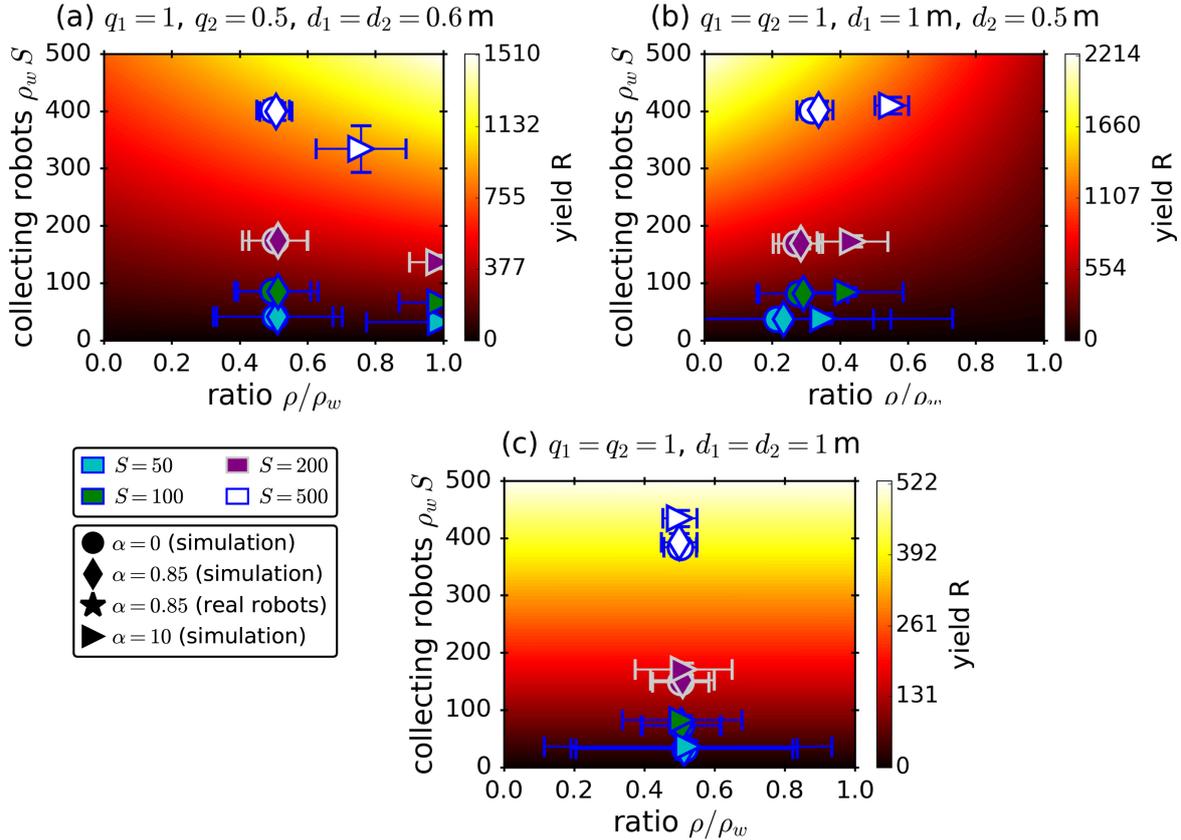


Fig. 6.9 (Colours online) Total yield R as a function of the normalised swarm allocation ρ/ρ_w and the number of worker robots $\rho_w S$ in the collision-free condition. We removed the effect of physical interactions (*i.e.* collisions between robots) that may cause traffic congestions and we report the predicted yield R from the model (6.4) as a colour heatmap and we overlay robot simulations for three strategies: distance-selective $\alpha = 0$ (circle), distance-quality trade-off $\alpha = 0.85$ (diamond), and quality-selective $\alpha = 10$ (triangle). We report simulations for swarm sizes $S = 50$ (cyan), $S = 100$ (green), $S = 200$ (purple) and $S = 500$ (white). Without collision, the predicted best strategy is allocation of all workers to the best-quality or closest source area. The collision-free simulations approximate such result when the corresponding strategy is activated, *e.g.* quality-selective $\alpha = 10$ (triangle) in panel (a) and the distance-selective $\alpha = 0$ (circle) in panel (b). Error bars represent 95% confidence intervals. Parameters: β_j , $T_{0,j}$ and κ_j are given in Table 6.1.

6.5 Discussion

Our results show how simple individual agents can collectively forage in a sophisticated manner. We assumed a minimal cognitive architecture including maintenance of a home vector (well evidenced in ants [28, 86]), and simple binary detection of pheromone trails and obstacles; our agents are thus much simpler than real ants. Combined with a simple pheromone deposition rule with a single tuneable parameter, however, we are able to qualitatively reproduce classical results such as the shortest path exploitation observed in lab ant colonies [66], and able to manage the classical distance-quality trade-off of foraging. To assess the performance of swarms using our proposed behaviour, we employed an optimality model accounting for congestion costs in foraging and examined the effect of resource distribution and colony size on the optimal distribution of foragers over forage patches. While others have previously considered the effect of colony size on recruitment strategy [145, 131, 119], our analysis instead assumes the recruitment strategy, and considers the optimal distribution. Our simple heuristic agent controllers are able to approximate the optimal distribution for relatively small swarm sizes, although large swarms depart from optimality. Large swarms cause crowded environments which require strategies to clear paths in order to reduce traffic congestion. We identify two possible strategies to limit traffic congestion: modifying the abandonment strategy or enriching the individual behaviour with collision-reactive states. In this work, after abandonment, the robots simply resumed exploration. The effects of this abandonment strategy are limited as robots quickly rediscover a path (which may be already congested). We believe that a better abandonment strategy (*e.g.* to stay at the depot for some time before resuming exploration, similar to ants [131]) could improve the results of the abandonment behaviour introduced in this work. Complementarily, traffic flow can be maintained undisrupted even in relatively crowded conditions by individual ants changing their behaviour as a function of collisions with other ants [43, 146]. Inspired by these results, the robot behaviour could be enriched with new collision-dependent states.

Our results are complementary to other approaches to minimal controllers necessary for collective behaviour in the swarm robotics field [61, 130]. Simple controllers increase the transferability to various robotics platforms thanks to their limited hardware requirements. Additionally, simple behaviours generally reduce the impact of the reality gap and preserve consistent dynamics in reality and simulations, as shown in our experiments where the same control software produced qualitatively similar results.

Our results illustrate the sophisticated collective dynamics that can be generated even by simple agents, which should be of interest to biologists and of practical utility to engineers. Similarly, our study of swarm size, and the scalability of foraging success, should interest both biologists and engineers, although it is worth noting that at least in some species of ants

congestion is much less of a problem compared to robots [92, 146]. In Section 6.4.2, we investigated a case closer to biology in which congestions did not impact the travel time; with model and simulations adapted accordingly. Nevertheless, we argue that taking a unifying perspective on the biology and engineering of collective foraging is illuminating, both through their similarities, and their differences.

Chapter 7

Conclusion and future work

Swarm robotics systems consist of a large number of simple and autonomous robots that locally interact with each other and with the environment via simple behavioural rules. The aggregate of these local interactions enable the swarm as a whole to exhibit interesting collective behaviours and to accomplish tasks that are not achievable by the single robots. A major challenge in designing robot swarms is to determine the behavioural rules through which the individual robots should interact to allow the swarm to perform specific tasks. The performance of the swarm in a specific task depends on the choice of the individual behavioural rules followed by the robots. To this end, the work presented in this thesis aimed at finding individual behavioural rules for improving the performance of robot swarms in two important collective behaviours.

The first collective behaviour we addressed in this thesis is known as the best-of- n decision problem where the swarm is required to reach a consensus for the best option among n available alternatives. Solving the best-of- n decision problem is considered to be an elementary ability that a swarm need to master to accomplish other collective tasks [209]. Previous research studies introduced several individual behaviours that allow robot swarms to solve the best-of- n problem [218, 134, 122, 123, 177, 211, 212, 210, 165, 167, 159, 209]. However, most of these behaviours were only tested in the case of $n = 2$ options. A recent theoretical work demonstrated a qualitative change in the dynamics of the decision process for $n > 2$ options [164]. That is, for $n > 2$ options, the swarm may be unable to reach a consensus, especially for challenging decision problems where the difference between options' qualities is small [7]. The study has also shown the existence of a dilemma. For the swarm to reach a consensus in case of $n > 2$ options, the individual robots must mainly rely on socially acquired information when updating their beliefs about the possible best option. However, the high usage of social information allows the initially randomly discovered options to quickly spread within the swarm even when having low quality and thus leads to a drop in the decision accuracy. On the other hand, if the individual robots rely mainly on individually acquired information for

updating their beliefs, the swarm makes accurate decisions when the difference between the qualities of the option is large. However, for small differences between the options' qualities, the swarm will be unable to reach a consensus due to insufficient interactions between the robots.

In our work, we confirmed the existence of the above dilemma through stochastic simulations and proposed individual behavioural rules to solve it. The individual behavioural rules require each robot to increase the strength of interaction with others progressively over time to limit the quick spread of first randomly discovered low-quality options that may occur when using initially strong interactions. Besides, the robots increase their interactions with others at a speed that is proportional to the quality of their preferred options. This results in a quicker spread of better options and hence a higher chance for the swarm to select the best option. Using stochastic analysis and swarm robotics simulations, we compared the decision accuracy of swarms implementing the Collective Decision through Cross-Inhibition decision strategy (CDCI) [181, 159] in best-of- n problem with $n > 2$, both with and without using the proposed individual behavioural rules. Using our individual behavioural rules resulted in a considerable improvement in the decision accuracy. The individual behavioural rule with the highest accuracy splits the decision-process into an exploration phase and exploitation phase. In the exploration phase, the robots survey the environment to accumulate information about the available options without interacting with others. This eliminates the premature spread of low-quality options. In the exploitation phase, the robots interact with others to exchange the information they acquired about the available options and reach a consensus, via a quality-proportional interaction strength. Moreover, in this two-phases behavioural rule, robots committed to high-quality options switch to the exploration phase (*i.e.* start interacting) earlier than those committed to low-quality options.

Although the two-phases behavioural rule allows reaching high decision accuracies, its performance is dependent on the correct tuning of the minimum time for switching to the exploration phase. Setting this time to very low values (*i.e.* not giving the swarm enough time to explore the environment) nullifies the benefit of the strategy while setting it to very high values leads to unnecessary delays and hence a drop in the decision speed. This is an important issue for future research. Preliminary results have shown that the optimal minimum time for switching to the exploration phase depends on environmental and robots' parameters such as the environment size, the robot's sensing and communication range, the decision problem difficulty, and the noise level of the robots' quality estimators. Therefore, tuning the robots' behaviour in advance is not possible as these parameters are not known to the designer in advance. However, it might be possible to allow the robots to autonomously decide about the best time to switch based on the information they gather during the exploration phase. As

future work, we aim at achieving this through the formulation of the switching decision as an optimal-stopping problem, which is a well-known class of mathematical problems [25, 11]. It is important to note that solving optimal stopping problems requires a minimum level of prior knowledge such as the distribution from where the qualities of the option are drawn from and the frequency of options in the operating environment.

In our first study, we demonstrated the benefit of splitting the collective decision process in swarm robotics into an exploration phase where individuals gather information and an exploitation phases where they interact to reach consensus. It will be interesting to investigate if such a strategy is adopted by biological organisms such as social insects and humans when making collective decisions. Reflecting on how we, humans, make a collective decision, it seems that we may be adopting such a strategy in a certain way. For instance, when deciding about a holiday destination for the family, each member spends some time searching for holiday packages alone (i.e. exploration phase). Then the most convinced and excited members start sharing their findings with the whole family (i.e. exploitation phase) to reach an agreement on the best holiday destination. The two-phases decision process might also be adopted by social insects such as honeybees. Scout honeybees go into an exploration trip to find the available nest sites before returning to disseminate their findings to other members in their colony. This process resembles the two-phases decision strategy studied in this. It might be interesting for biologists to investigate the extent to which the honeybees decision process can be considered a two-phase decision process and analyse the implication of this process on the colony's decisions.

In some scenarios of the best-of- n decision problem, making accurate decisions is not the only requirement. The swarm needs to be able to adapt its decision to environmental changes. For instance, in these scenarios, if a better option appears in the environment, the swarm should switch its decision to this option. In our second work on the best-of- n decision problem, we aimed at giving the swarm the ability to adapt its decision to environmental changes such as the appearance of a better option, the disappearance of the swarm's option, and a sudden swap in the qualities of the two best options. We introduced two simple individual behavioural rules that, each of them, allows robot swarms to adapt their decisions to environmental changes. The first rule requires the robot to continuously compare its current option to other encountered options and to switch its opinion to any option it estimates to be better than its current one. In the second rule, because only uncommitted robots consider committing to encountered options, committed robots forget their options at a constant rate to reconsider their opinions. In addition to using the above behavioural rules, to allow the swarm to adapt its decision in case of the considered environmental changes, each robot must continually monitor the quality and the presence of its option. We tested the effectiveness of the proposed individual behavioural rules

in allowing adaptation by integrating them into the weighted voter decision-making model [211] which does not allow adaptation in its original form [148]. Then, we tested the resulting decision-making models using both multi-agent and physics-based swarm robotics simulations. Our tests have shown that both individual behavioural rules enable adaptation, but the swarm's response to environmental changes is faster when using the compare-based rule. Moreover, the compare-based rule allows setting the minimum quality improvement that must be presented by the environmental change for the swarm to adapt its decision. This can be used to avoid unnecessary switching to similar or nearly similar options. On the other hand, when using the forget-based rule, controlling adaptation is less straightforward. If the forgetting rate is too high, the swarm will be unable to reach consensus, and if it is too low, the swarm will be unable to adapt to changes. However, it is important to note that the forget-based rule is cognitively less demanding than the compare-based rule as it only requires the robot to spontaneously abandon its choice. Instead, the compare-based rule requires the robot to be able to store and compare two values.

Our analysis of the proposed adaptation behavioural rules revealed a counter-intuitive result, that is, the higher is the density of the robots in the environment and the larger the robot's communication range, the less adaptive is the swarm, especially when the difference between options' qualities is small. The reason for this is that in dense environments and for a large communication ranges, the individual robot interacts with a high number of peers at the same time. As a result, when the swarm has already reached a consensus, it is difficult for a robot to spread a newly discovered opinion as the robot gets quickly reverted to the previous choice by its peers. Moreover, our analysis suggests that swarms better adapt when individuals rely on constrained communication. This result is particularly interesting as it highlights the importance of local communication in swarm robotics [76] and is in line with a recent biological study that revealed the importance of limited cognitive and sensory capabilities for the emergence of collective behaviours in animal groups [155]. It may also be interesting for biologists to investigate if this effect of communication range and density on adaptation is present in living organisms. One example can be to examine whether the increased communication between humans enabled by social media platforms is influencing the humans' ability to adapt their choices in collective matters such as fashion trends and political ideologies. In future studies, we plan to confirm the effect of density and communication range using real robots. Additionally, we aim at investigating additional behavioural rules that might enable adaptation in the case of high-density swarms. A possible rule can be that robots finding better options ignore the recruitment messages of others for a time that is proportional to the quality improvement presented by the new option. This might allow the robots to spread the new opinion within the population and hence drive the swarm to adapt its decision.

The second collective behaviour we addressed in this thesis is the collective resource collection task where the robots are asked to retrieve objects spread in an unknown environment. We addressed this task for its numerous prospective real-world applications including space exploration, search and rescue, and the collection of natural resources [220, 16, 221]. In line with the distribution of natural resources in the real world [168], we considered that the objects to collect are distributed in the environment in the form of clusters (*i.e.* source areas). To effectively collect clustered objects, robots need to rely on memory and communication [85]. Therefore, similarly to a considerable amount of literature [65, 217, 136, 129, 19, 39, 90, 153, 60, 83, 82, 85, 111, 112], we focused our attention on stigmergy-based resource collection where robots mark the environment as a way to remember the resources' locations (*i.e.* memory) and indicate them to other robots (*i.e.* communication). Previous research on stigmergy-based resource collection assumed that the objects to collect are all identical and thus aimed at finding individual behaviours that minimise the time to complete the collection of the available objects [83, 82, 111, 112]. In contrast, in our work, we considered objects of different values (*i.e.* quality). Value-based resource collection reflects a class of potential real-world applications where the objects to retrieve have different importance or priority. For instance, in human-performed search and rescue operations, firefighters are trained to rescue victims based on their risk level, starting with those at a higher risk [38]. Value-based resource collection has also been reported in biology where some species of foraging ants vary their collection behaviour based on the nutritional value of the food present in their environment [9, 147, 182].

In our work, we introduced an ant-inspired stigmergy-based individual behaviour that allows robot swarms to show suitable responses in scenarios where the objects have different values. Our behaviour requires robots with minimal cognitive abilities that include a limited memory for storing a home vector (similarly to some species of ants [28, 86]), binary sensors for detecting pheromone trails and obstacles, and a binary pheromone deposition actuator. Our proposed individual behaviour has a single parameter that controls the behavioural rules of pheromone deposition and path abandonment. The value of this parameter controls the collective response demonstrated by the swarm and allows to satisfy various objectives that may be encountered during real-world applications, such as focusing the collection on the highest-quality objects, the nearest objects, or balancing the distance-quality trade-off. We assessed the performance of robot swarms implementing our proposed behaviour using an optimality model that considers the costs of crowding. We employed this optimality model to investigate how swarm size and resource distribution affect the collective behaviour exhibited by the swarm. The results of our analysis have shown that for relatively small swarms, our simple individual behaviour results in the optimal robot distribution over the source areas suggested by

the optimality model. However, for large swarms, the resulting collective behaviour deviates from optimality due to crowded paths caused by the high number of robots.

Our proposed individual behaviour contains an abandonment strategy that was intended to limit the effect of crowding. Following this abandonment strategy, when robots abandon crowded paths, they directly resume random exploration. The reason why the proposed abandonment strategy failed to limit the effect of crowding in case of large swarms is that resuming random exploration after abandonment quickly leads to other paths that are already crowded nullifying the benefit of abandonment. As future work, we plan to implement an improved version of our abandonment strategy in which the robots abandoning crowded paths remain at the depot for some time before resuming exploration (similar to ants [131]). We believe that this new abandonment strategy can be tuned to regulate the number of worker robots and hence may allow achieving optimal resource collection in the case of large swarms. Moreover, in future investigations, we intend to compare the performance of our proposed resource collection behaviour to that of existing algorithms such as the *Central Place Foraging Algorithm (CPFA)* [85].

As part of our collective resource collection study, we reproduced the well-know double-bridge experiment that biologists used to demonstrate ants' ability to select the shorter of two branching paths of different lengths. Using the double bridge setup, we demonstrated that our proposed ant-inspired behaviour allows simple robots to exhibit similar collective behaviour as more complex real ants. This experiment highlights the potential of using robotics to investigate biological assumptions, especially with the ease of tuning the experimental conditions in robotics setups.

Although we addressed only the best-of- n decision problem and the collective resource collection task, the individual behavioural rules introduced in this thesis can inspire swarm robotics researchers working on other collective behaviours. Moreover, the results presented here can be of interest to biologists investigating the individual behavioural rules behind the collective behaviours observed in superorganisms.

References

- [1] Agassounon, W. and Martinoli, A. (2002). Efficiency and robustness of threshold-based distributed allocation algorithms in multi-agent systems. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 3, AAMAS '02*, page 1090–1097.
- [2] Aggarwal, A., Gupta, D., Vining, W. F., Fricke, G. M., and Moses, M. E. (2019). Ignorance is not bliss: An analysis of central-place foraging algorithms. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6510–6517.
- [3] Albani, D., IJsselmuiden, J., Haken, R., and Trianni, V. (2017). Monitoring and mapping with robot swarms for agricultural applications. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6.
- [4] Arganda, S., Nicolis, S. C., Perochain, A., Péchabadens, C., Latil, G., and Dussutour, A. (2014). Collective choice in ants: The role of protein and carbohydrates ratios. *Journal of Insect Physiology*, 69:19–26.
- [5] Arvin, F., Yue, S., and Xiong, C. (2015). Colias- ϕ : An autonomous micro robot for artificial pheromone communication. *International Journal of Mechanical Engineering and Robotics Research*, 4(4):349–353.
- [6] Athreya, A. P. and Tague, P. (2012). Survivable smart grid communication: Smart-meters meshes to the rescue. In *2012 International Conference on Computing, Networking and Communications (ICNC)*, pages 104–110.
- [7] Baronchelli, A. (2018). The emergence of consensus: a primer. *Royal Society Open Science*, 5(2).
- [8] Bayındır, L. (2016). A review of swarm robotics tasks. *Neurocomputing*, 172:292–321.
- [9] Beckers, R., Deneubourg, J.-L., and Goss, S. (1993). Modulation of trail laying in the ant *Lasius niger* (Hymenoptera: Formicidae) and its role in the collective selection of a food source. *Journal of Insect Behavior*, 6(6):751—759.
- [10] Beckers, R., Deneubourg, J.-L., Goss, S., and Pasteels, J. M. (1990). Collective decision making through food recruitment. *Insectes Sociaux*, 37(3):258–267.
- [11] Berezovsky, B. and Gnedin, A. (1984). *Problems of Best Selection (in Russian)*. Academia Nauk.

- [12] Berman, S., Halasz, A., Hsieh, M. A., and Kumar, V. (2009). Optimized Stochastic Policies for Task Allocation in Swarms of Robots. *IEEE Transactions on Robotics*, 25(4):927–937.
- [13] Bidari, S., Peleg, O., and Kilpatrick, Z. P. (2019). Social inhibition maintains adaptivity and consensus of honeybees foraging in dynamic environments. *Royal Society Open Science*, 6(12):191681.
- [14] Biro, D., Sumpter, D. J., Meade, J., and Guilford, T. (2006). From compromise to leadership in pigeon homing. *Current Biology*, 16(21):2123 – 2128.
- [15] Bosien, A., Turau, V., and Zambonelli, F. (2012). Approaches to fast sequential inventory and path following in RFID-enriched environments. *International Journal of Radio Frequency Identification Technology and Applications*, 4(1):28–48.
- [16] Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41.
- [17] Bregy, P., Sommer, S., and Wehner, R. (2008). Nest-mark orientation versus vector navigation in desert ants. *Journal of Experimental Biology*, 211(12):1868–1873.
- [18] Britton, N. F., Franks, N. R., Pratt, S. C., and Seeley, T. D. (2002). Deciding on a new home: how do honeybees agree? *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 269(1498):1383–1388.
- [19] Campo, A., Gutiérrez, Á., Nouyan, S., Pinciroli, C., Longchamp, V., Garnier, S., and Dorigo, M. (2010). Artificial pheromone for path selection by a foraging swarm of robots. *Biological Cybernetics*, 103(5):339–352.
- [20] Canciani, F., Talamali, M. S., Marshall, J. A. R., Bose, T., and Reina, A. (2019a). Keep calm and vote on: Swarm resiliency in collective decision making. In *Proceedings of Workshop Resilient Robot Teams of the 2019 IEEE International Conference on Robotics and Automation (ICRA 2019)*, page 4 pages.
- [21] Canciani, F., Talamali, M. S., and Reina, A. (2019b). DeMaMAS: Decision Making Multi-Agent Simulator. URL: <https://github.com/DiODEProject/DeMaMAS>. Accessed January 1st, 2020.
- [22] Carrillo-Zapata, D., Milner, E., Hird, J., Tzoumas, G., Vardanega, P. J., Sooriyabandara, M., Giuliani, M., Winfield, A. F. T., and Hauert, S. (2020). Mutual shaping in swarm robotics: User studies in fire and rescue, storage organization, and bridge inspection. *Frontiers in Robotics and AI*, 7:53.
- [23] Charbonneau, D., Hillis, N., and Dornhaus, A. (2015). ‘Lazy’ in nature: Ant colony time budgets show high ‘inactivity’ in the field as well as in the lab. *Insectes Sociaux*, 62(1):31–35.
- [24] Choe, D.-H., Villafuerte, D. B., and Tsutsui, N. D. (2012). Trail pheromone of the Argentine ant, *Linepithema humile* (Mayr) (Hymenoptera: Formicidae). *PLoS ONE*, 7(9):e45016.
- [25] Chow, Y., Robbins, H., and Siegmund, D. (1971). *Great Expectations: The Theory of Optimal Stopping*. Houghton Mifflin Company.

- [26] Christensen, A. L., O’Grady, R., and Dorigo, M. (2009). From fireflies to fault-tolerant swarms of robots. *IEEE Transactions on Evolutionary Computation*, 13(4):754–766.
- [27] Cody, J. R. and Adams, J. A. (2017). An evaluation of quorum sensing mechanisms in collective value-sensitive site selection. In *Proceedings of the 2017 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, pages 40–47. IEEE.
- [28] Collett, T. S. and Collett, M. (2002). Memory use in insect visual navigation. *Nature Reviews Neuroscience*, 3(7):542–552.
- [29] Cronin, A. L. (2018). An agent-based model of nest-site selection in a mass-recruiting ant. *Journal of Theoretical Biology*, 455:54–63.
- [30] Cronin, A. L. (2019). Fussy groups thwart the collective burden of choice: A theoretical study of house-hunting ants. *Journal of Theoretical Biology*, 483:110000.
- [31] Czaczkes, T. J., Grüter, C., Ellis, L., Wood, E., and Ratnieks, F. L. W. (2013). Ant foraging on complex trails: Route learning and the role of trail pheromones in *Lasius niger*. *The Journal of Experimental Biology*, 216(2):188–197.
- [32] Dames, P., Schwager, M., Kumar, V., and Rus, D. (2012). A decentralized control policy for adaptive information gathering in hazardous environments. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 2807–2813.
- [33] Deneubourg, J.-L., Aron, S., Goss, S., and Pasteels, J. M. (1990). The self-organizing exploratory pattern of the argentine ant. *Journal of Insect Behavior*, 3(2):159–168.
- [34] Detrain, C. and Deneubourg, J.-L. (2006). Self-organized structures in a superorganism: Do ants “behave” like molecules? *Physics of Life Reviews*, 3(3):162–187.
- [35] Devigne, C., Renon, A. J., and Detrain, C. (2004). Out of sight but not out of mind: modulation of recruitment according to home range marking in ants. *Animal Behaviour*, 67(6):1023–1029.
- [36] Dimidov, C., Oriolo, G., and Trianni, V. (2016). Random walks in swarm robotics: An experiment with Kilobots. In *Swarm Intelligence (ANTS 2016)*, volume 9882 of *LNCIS*, pages 185–196. Springer.
- [37] Dorigo, M., Birattari, M., and Brambilla, M. (2014). Swarm robotics. *Scholarpedia*, 9(1):1463.
- [38] Dornan, S. (2008). *Industrial fire brigade principles and practice*. Jones and Bartlett.
- [39] Ducatelle, F., Di Caro, G. A., Pinciroli, C., and Gambardella, L. M. (2011a). Self-organized cooperation between robotic swarms. *Swarm Intelligence*, 5(2):73–96.
- [40] Ducatelle, F., Di Caro, G. A., Pinciroli, C., Mondada, F., and Gambardella, L. M. (2011b). Communication assisted navigation in robotic swarms: self-organization and cooperation. In *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011)*, pages 4981–4988. IEEE.

- [41] Dussutour, A., Beekman, M., Nicolis, S. C., and Meyer, B. (2009). Noise improves collective decision-making by ants in dynamic environments. *Proceedings of the Royal Society of London B: Biological Sciences*, 276(1677):4353–4361.
- [42] Dussutour, A., Deneubourg, J.-L., and Fourcassié, V. (2005). Temporal organization of bi-directional traffic in the ant *Lasius niger* (L.). *Journal of Experimental Biology*, 208(15):2903–2912.
- [43] Dussutour, A., Fourcassié, V., Helbing, D., and Deneubourg, J.-L. (2004). Optimal traffic organization in ants under crowded conditions. *Nature*, 428(6978):70–73.
- [44] Dussutour, A. and Simpson, S. J. (2009). Communal nutrition in ants. *Current Biology*, 19(9):740–744.
- [45] Ferrante, E., Brambilla, M., Birattari, M., and Dorigo, M. (2013). Socially-Mediated Negotiation for Obstacle Avoidance in Collective Transport. In *Distributed Autonomous Robotic Systems*, volume 83 of *Springer Tracts in Advanced Robotics*, pages 571–583.
- [46] Ferrante, E., Turgut, A. E., Huepe, C., Stranieri, A., Pinciroli, C., and Dorigo, M. (2012). Self-organized flocking with a mobile robot swarm: a novel motion control method. *Adaptive Behavior*, 20(6):460–477.
- [47] Ferrante, E., Turgut, A. E., Stranieri, A., Pinciroli, C., Birattari, M., and Dorigo, M. (2014). A self-adaptive communication strategy for flocking in stationary and non-stationary environments. *Natural Computing*, 13(2):225–245.
- [48] Fick, A. (1855). Ueber diffusion. *Annalen der Physik*, 170(1):59–86.
- [49] Font Llenas, A., Talamali, M. S., Xu, X., Marshall, J. A. R., and Reina, A. (2018). Quality-sensitive foraging by a robot swarm through virtual pheromone trails. In *Swarm Intelligence (ANTS 2018)*, volume 11172 of *LNCS*, pages 135–149. Springer.
- [50] Fourcassié, V., Dussutour, A., and Deneubourg, J.-L. (2010). Ant traffic rules. *Journal of Experimental Biology*, 213(14):2357–2363.
- [51] Francesca, G. and Birattari, M. (2016). Automatic Design of Robot Swarms: Achievements and Challenges. *Frontiers in Robotics and AI*, 3:224–9.
- [52] Franks, N. R., Dornhaus, A., Best, C. S., and Jones, E. L. (2006). Decision making by small and large house-hunting ant colonies: one size fits all. *Animal Behaviour*, 72(3):611–616.
- [53] Fricke, G. M., Hecker, J. P., Griego, A. D., Tran, L. T., and Moses, M. E. (2016). A distributed deterministic spiral search algorithm for swarms. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4430–4436.
- [54] Fujisawa, R., Dobata, S., Kubota, D., Imamura, H., and Matsuno, F. (2008). Dependency by concentration of pheromone trail for multiple robots. In *SAnt Colony Optimization and Swarm Intelligence (ANTS 2008)*, volume 5217 of *LNCS*, pages 283–290. Springer.
- [55] Fujisawa, R., Dobata, S., Sugawara, K., and Matsuno, F. (2014). Designing pheromone communication in swarm robotics: Group foraging behavior mediated by chemical substance. *Swarm Intelligence*, 8(3):227–246.

- [56] Garnier, S., Combe, M., Jost, C., and Theraulaz, G. (2013a). Do ants need to estimate the geometrical properties of trail bifurcations to find an efficient route? a swarm robotics test bed. *PLoS Computational Biology*, 9(3):e1002903.
- [57] Garnier, S., Gautrais, J., Asadpour, M., Jost, C., and Theraulaz, G. (2009). Self-organized aggregation triggers collective decision making in a group of cockroach-like robots. *Adaptive Behavior*, 17(2):109–133.
- [58] Garnier, S., Jost, C., Jeanson, R., Gautrais, J., Asadpour, M., Caprari, G., and Theraulaz, G. (2005). Aggregation behaviour as a source of collective decision in a group of cockroach-like-robots. In *Advances in Artificial Life*, pages 169–178.
- [59] Garnier, S., Murphy, T., Lutz, M., Hurme, E., and Leblanc, S. (2013b). Stability and Responsiveness in a Self-Organized Living Architecture. *PLOS Computational Biology*, 9(3):1002984.
- [60] Garnier, S., Tâche, F., Combe, M., Grimal, A., and Theraulaz, G. (2007). Alice in pheromone land: An experimental setup for the study of ant-like robots. In *Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007)*, pages 37–44. IEEE.
- [61] Gauci, M., Chen, J., Li, W., Dodd, T. J., and Groß, R. (2014). Self-organized aggregation without computation. *The International Journal of Robotics Research*, 33(8):1145–1161.
- [62] Ghassemi, A., Bavarian, S., and Lampe, L. (2010). Cognitive radio for smart grid communications. In *2010 First IEEE International Conference on Smart Grid Communications*, pages 297–302.
- [63] Giancola, S., Valenti, M., and Sala, R. (2018). *A Survey on 3D Cameras: Metrological Comparison of Time-of-Flight, Structured-Light and Active Stereoscopy Technologies*. Springer.
- [64] Gillespie, D. T. (1976). A general method for numerically simulation the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22(4):403–434.
- [65] Goss, S., Deneubourg, J.-L., Bourguine, P., and Varela, E. (1992). Harvesting by a group of robots. In *1st European Conference on Artificial Life*, pages 195–204. MIT Press.
- [66] Goss, S., Deneubourg, J.-L., and Pasteels, J. M. (1989). Self-organized shortcuts in the Argentine ant. *Naturwissenschaften*, 76(12):579–581.
- [67] Graham, J. M., Kao, A. B., Wilhelm, D. A., and Garnier, S. (2017). Optimal construction of army ant living bridges. *Journal of Theoretical Biology*, 435:184 – 198.
- [68] Gray, R., Franci, A., Srivastava, V., and Leonard, N. E. (2018). Multiagent decision-making dynamics inspired by honeybees. *IEEE Transactions on Control of Network Systems*, 5(2):793–806.
- [69] Groß, R. and Dorigo, M. (2009). Towards group transport by swarms of robots. *Int. J. Bio-Inspired Comput.*, 1(1/2):1–13.
- [70] Gunther, N. J. (2000). *The practical performance analyst*. Authors Choice Press.

- [71] Gutiérrez, Á., Campo, A., Monasterio-Huelin, F., Magdalena, L., and Dorigo, M. (2010). Collective decision-making based on social odometry. *Neural Computation*, 19(6):807–823.
- [72] Halme, A. (2012). Kilobot app—a kilobot simulator and swarm pattern designer. URL: <https://github.com/ajhalme/kbsim>. Accessed April 20th, 2018.
- [73] Hamann, H. (2012). Towards swarm calculus: Universal properties of swarm performance and collective decisions. In *Swarm Intelligence (ANTS 2012)*, volume 7461 of *LNCS*, pages 168–179. Springer.
- [74] Hamann, H. (2013). Towards swarm calculus: Urn models of collective decisions and universal properties of swarm performance. *Swarm Intelligence*, 7(2-3):145–172.
- [75] Hamann, H. (2018a). Superlinear scalability in parallel computing and multi-robot systems: shared resources, collaboration, and network topology. In *International Conference on Architecture of Computing Systems (ARCS 2018)*, volume 10793 of *LNCS*, pages 31–42. Springer.
- [76] Hamann, H. (2018b). *Swarm Robotics: A Formal Approach*. Springer.
- [77] Hamann, H. and Reina, A. (2020). Scalability in computing and robotics. *arXiv*, 2006.04969.
- [78] Hamann, H., Schmickl, T., Wörn, H., and Crailsheim, K. (2012). Analysis of emergent symmetry breaking in collective decision making. *Neural Computing and Applications*, 21(2):207–218.
- [79] Hamann, H. and Wörn, H. (2008). A framework of space–time continuous models for algorithm design in swarm robotics. *Swarm Intelligence*, 2(2-4):209–239.
- [80] Hammersley, J. M. and Handscomb, D. C. (1964). *Monte Carlo methods*. Methuen London.
- [81] Hangartner, W. (1969). Orientierung von *Lasius fuliginosus* Latr. an einer Gabelung der Geruchsspur. *Insectes Sociaux*, 16(1):55–60.
- [82] Hecker, J. P., Carmichael, J. C., and Moses, M. E. (2015). Exploiting clusters for complete resource collection in biologically-inspired robot swarms. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 434–440.
- [83] Hecker, J. P., Letendre, K., Stolleis, K., Washington, D., and Moses, M. E. (2012). Formica ex Machina: Ant swarm foraging from physical to virtual and back again. In *Swarm Intelligence (ANTS 2012)*, volume 7461 of *LNCS*, pages 252–259. Springer.
- [84] Hecker, J. P. and Moses, M. E. (2013). An evolutionary approach for robust adaptation of robot behavior to sensor error. In *GECCO 2013 - Proceedings of the 2013 Genetic and Evolutionary Computation Conference Companion*, pages 1437–1444.
- [85] Hecker, J. P. and Moses, M. E. (2015). Beyond pheromones: evolving error-tolerant, flexible, and scalable ant-inspired robot swarms. *Swarm Intelligence*, 9(1):43–70.
- [86] Heinze, S., Narendra, A., and Cheung, A. (2018). Principles of insect path integration. *Current Biology*, 28(17):R1043–R1058.

- [87] Herianto and Kurabayashi, D. (2009). Realization of an artificial pheromone system in random data carriers using RFID tags for autonomous navigation. In *Proceedings of the 2009 IEEE/RSJ International Conference on Robotics and Automation (ICRA 2009)*, pages 2288–2293. IEEE.
- [88] Herianto, Sakakibara, T., and Kurabayashi, D. (2007). Artificial pheromone system using RFID for navigation of autonomous robots. *Journal of Bionic Engineering*, 4(4):245–253.
- [89] Heylighen, F. (2016). Stigmergy as a universal coordination mechanism ii: Varieties and evolution. *Cognitive Systems Research*, 38:50–59.
- [90] Hoff, N., Wood, R., and Nagpal, R. (2012). Distributed colony-level algorithm switching for robot swarm foraging. In *Distributed Autonomous Robotic Systems (DARS 2010)*, volume 83 of *STAR*, pages 417–430. Springer.
- [91] Hölldobler, B. and Wilson, E. O. (1990). *The Ants*. Harvard University Press.
- [92] Hönicke, C., Bliss, P., and Moritz, R. F. A. (2015). Effect of density on traffic and velocity on trunk trails of *Formica pratensis*. *The Science of Nature*, 102(3-4):17.
- [93] Houston, A. I. and McNamara, J. M. (2014). Foraging currencies, metabolism and behavioural routines. *Journal of Animal Ecology*, 83(1):30–40.
- [94] Hsieh, M. A., Halász, A., Berman, S., and Kumar, V. (2008). Biologically inspired redistribution of a swarm of robots among multiple sites. *Swarm Intelligence*, 2(2-4):121–141.
- [95] Jaffe, J. S., Franks, P. J. S., Roberts, P. L. D., Mirza, D., Schurgers, C., Kastner, R., and Boch, A. (2017). A swarm of autonomous miniature underwater robot drifters for exploring submesoscale ocean dynamics. *Nature Communications*, 8(1):14189.
- [96] Jansson, F., Hartley, M., Hinsch, M., Slavkov, I., Carranza, N., Olsson, T. S. G., Dries, R. M., Grönqvist, J. H., Marée, A. F. M., Sharpe, J., Kaandorp, J. A., and Grieneisen, V. A. (2015). Kilombo: a Kilobot simulator to enable effective research in swarm robotics. *arXiv.org*, 1511.04285.
- [97] Johansson, A., Ramsch, K., Middendorf, M., and Sumpter", D. J. (2012). Tuning positive feedback for signal detection in noisy dynamic environments. *Journal of Theoretical Biology*, 309:88–95.
- [98] Kacelnik, A. (1984). Central place foraging in Starlings (*Sturnus vulgaris*). I. patch residence time. *The Journal of Animal Ecology*, 53(1):283.
- [99] Kaur, R., Anoop, K., and Sumana, A. (2012). Leaders follow leaders to reunite the colony: Relocation dynamics of an Indian queenless ant in its natural habitat. *Animal Behaviour*, 83(6):1345–1353.
- [100] Kernbach, S., Thenius, R., Kernbach, O., and Schmickl, T. (2009). Re-embodiment of honeybee aggregation behavior in an artificial micro-robotic system. *Adaptive Behavior*, 17(3):237–259.

- [101] Khaliq, A. A., Di Rocco, M., and Saffiotti, A. (2014). Stigmergic algorithms for multiple minimalistic robots on an RFID floor. *Swarm Intelligence*, 8(3):199–225.
- [102] Khaluf, Y., Rausch, I., and Simoens, P. (2018). The impact of interaction models on the coherence of collective decision-making: a case study with simulated locusts. In *Swarm Intelligence (ANTS2018)*, volume 11172 of *LNCS*, pages 252–263. Springer.
- [103] Krapivsky, P. L. and Redner, S. (2003). Dynamics of majority rule in two-state interacting spin systems. *Physical Review Letters*, 90:238701.
- [104] Krause, J., Ruxton, G. D., and Ruxton, G. D. (2002). *Living in groups*. Oxford University Press.
- [105] Kümmerli, R. (2009). Social insects – superorganisms or just superb organisms? *Current Biology*, 19(3):R105 – R107.
- [106] Lerman, K., Martinoli, A., and Galstyan, A. (2005). A review of probabilistic macroscopic models for swarm robotic systems. In *Swarm Robotics*, volume 3342 of *LNCS*, pages 143–152. Springer.
- [107] Letendre, K. and Moses, M. E. (2013). Synergy in ant foraging strategies: Memory and communication alone and in combination. In *GECCO 2013 - Proceedings of the 2013 Genetic and Evolutionary Computation Conference*, pages 41–48.
- [108] Li, J., Esteban-Fernández de Ávila, B., Gao, W., Zhang, L., and Wang, J. (2017). Micro/nanorobots for biomedicine: Delivery, surgery, sensing, and detoxification. *Science Robotics*, 2(4).
- [109] Liggett, T. M. (1999). *Voter Models*, pages 139–208. Springer.
- [110] Lu, Q., Griego, A. D., Fricke, G. M., and Moses, M. E. (2019). Comparing physical and simulated performance of a deterministic and a bio-inspired stochastic foraging strategy for robot swarms. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9285–9291.
- [111] Lu, Q., Hecker, J. P., and Moses, M. E. (2016). The MPFA: A multiple-place foraging algorithm for biologically-inspired robot swarms. In *IEEE International Conference on Intelligent Robots and Systems*, pages 3815–3821.
- [112] Lu, Q., Hecker, J. P., and Moses, M. E. (2018). Multiple-place swarm foraging with dynamic depots. *Autonomous Robots*, 42(4):909–926.
- [113] Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., and Balan, G. (2005). MASON: A multiagent simulation environment. *Simulation*, 81(7):517–527.
- [114] Mallon, E., Pratt, S., and Franks, N. (2001). Individual and collective decision-making during nest site selection by the ant *leptothorax albigipennis*. *Behavioral Ecology and Sociobiology*, 50(4):352–359.
- [115] Mamei, M. and Zambonelli, F. (2005). Physical deployment of digital pheromones through RFID technology. In *Proceedings of the 2005 IEEE Swarm Intelligence Symposium (SIS 2005)*, pages 281–288. IEEE.

- [116] Mamei, M. and Zambonelli, F. (2007). Pervasive pheromone-based interaction with RFID tags. *ACM Transactions on Autonomous and Adaptive Systems*, 2(2):4.
- [117] Marshall, J. A. R., Bogacz, R., Dornhaus, A., Planqué, R., Kovacs, T., and Franks, N. R. (2009). On optimal decision-making in brains and social insect colonies. *Journal of The Royal Society Interface*, 6(40):1065–1074.
- [118] Mayet, R., Roberz, J., Schmickl, T., and Crailsheim, K. (2010). Antbots: A feasible visual emulation of pheromone trails for swarm robots. In *Swarm Intelligence (ANTS 2010)*, volume 6234 of *LNCS*, pages 84–94. Springer.
- [119] Mayya, S., Pierpaoli, P., and Egerstedt, M. (2019). Voluntary retreat for decentralized interference reduction in robot swarms. In *Proceedings of the 2019 IEEE/RSJ International Conference on Robotics and Automation (ICRA 2019)*, pages 9667–9673. IEEE.
- [120] Miller, N., Garnier, S., Hartnett, A. T., and Couzin, I. D. (2013). Both information and social cohesion determine collective decisions in animal groups. *Proceedings of the National Academy of Sciences*, 110(13):5263–5268.
- [121] Mondada, F., Pettinaro, G. C., Guignard, A., Kwee, I. W., Floreano, D., Deneubourg, J.-L., Nolfi, S., Gambardella, L. M., and Dorigo, M. (2004). SWARM-BOT: A new distributed robotic concept. *Autonomous Robots*, 17(2):193–221.
- [122] Montes de Oca, M., Ferrante, E., Scheidler, A., Pinciroli, C., Birattari, M., and Dorigo, M. (2011). Majority-rule opinion dynamics with differential latency: A mechanism for self-organized collective decision-making. *Swarm Intelligence*, 5(3-4):305–327.
- [123] Montes de Oca, M. A., Ferrante, E., Scheidler, A., and Rossi, L. F. (2013). Binary consensus via exponential smoothing. In *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, volume 126, pages 244–255.
- [124] Murphy, R. (2019). *Introduction to AI robotics*. MIT Press, 2 edition.
- [125] Na, S., Qiu, Y., Turgut, A. E., Ulrich, J., Krajník, T., Yue, S., Lennox, B., and Arvin, F. (2020). Bio-inspired artificial pheromone system for swarm robotics applications. *Adaptive Behavior*, page 1059712320918936.
- [126] Nicolis, S. C., Zabzina, N., Latty, T., and Sumpter, D. J. T. (2011). Collective irrationality and positive feedback. *PLoS ONE*, 6(4):e18901.
- [127] Nonacs, P. and Dill, L. M. (1990). Mortality risk vs. food quality trade-offs in a common currency: Ant patch preferences. *Ecology*, 71(5):1886–1892.
- [128] Nouyan, S., Campo, A., and Dorigo, M. (2008). Path formation in a robot swarm. *Swarm Intelligence*, 2(1):1–23.
- [129] Nouyan, S., Groß, R., Bonani, M., Mondada, F., and Dorigo, M. (2009). Teamwork in self-organized robot colonies. *IEEE Transactions on Evolutionary Computation*, 13(4):695–711.

- [130] Özdemir, A., Gauci, M., Bonnet, S., and Groß, R. (2018). Finding consensus without computation. *IEEE Robotics and Automation Letters*, 3(3):1346–1353.
- [131] Pagliara, R., Gordon, D. M., and Leonard, N. E. (2018). Regulation of harvester ant foraging as a closed-loop excitable system. *PLOS Computational Biology*, 14(12):e1006200.
- [132] Pais, D., Hogan, P. M., Schlegel, T., Franks, N. R., Leonard, N. E., and Marshall, J. A. R. (2013). A mechanism for value-sensitive decision-making. *PLoS ONE*, 8(9):e73216.
- [133] Parker, C. A. C. and Zhang, H. (2005). Active versus passive expression of preference in the control of multiple-robot decision-making. In *IEEE International Conference on Intelligent Robots and Systems (IROS2005)*, pages 3706–3711. IEEE.
- [134] Parker, C. A. C. and Zhang, H. (2009). Cooperative decision-making in decentralized multiple-robot systems: the best-of-n problem. *IEEE Transactions on Mechatronics*, 14(2):240–251.
- [135] Parker, C. A. C. and Zhang, H. (2010). Collective unary decision-making by decentralized multiple-robot systems applied to the task-sequencing problem. *Swarm Intelligence*, 4(4):199–220.
- [136] Payton, D. W., Daily, M., Estowski, R., Howard, M., and Lee, C. (2001). Pheromone robotics. *Autonomous Robots*, 11(3):319–324.
- [137] Paz Flanagan, T., Letendre, K., Burnside, W., Fricke, G. M., and Moses, M. (2011). How ants turn information into food. In *IEEE SSCI 2011 - Symposium Series on Computational Intelligence - IEEE ALIFE 2011: 2011 IEEE Symposium on Artificial Life*, pages 178–185.
- [138] Pinciroli, C., Talamali, M. S., Reina, A., Marshall, J. A. R., and Trianni, V. (2018). Simulating Kilobots within ARGoS: models and experimental validation. In *Swarm Intelligence (ANTS 2018)*, volume 11172 of *LNCS*, pages 176–187. Springer.
- [139] Pinciroli, C., Trianni, V., O’Grady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., Di Caro, G., Ducatelle, F., Birattari, M., Gambardella, L. M., and Dorigo, M. (2012). ARGoS: A modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, 6(4):271–295.
- [140] Pini, G., Brutschy, A., Birattari, M., and Dorigo, M. (2009). Interference reduction through task partitioning in a robotic swarm. In *Sixth International Conference on Informatics in Control, Automation and Robotics (ICINCO 2009)*, pages 52–59.
- [141] Pini, G., Brutschy, A., Francesca, G., Dorigo, M., and Birattari, M. (2012). Multi-armed bandit formulation of the task partitioning problem in swarm robotics. In *Swarm Intelligence (ANTS 2012)*, *LNCS*, pages 109–120.
- [142] Pirrone, A., Stafford, T., and Marshall, J. A. R. (2014). When natural selection should optimize speed-accuracy trade-offs. *Frontiers in Neuroscience*, 08(73):1–5.
- [143] Pitonakova, L., Crowder, R., and Bullock, S. (2016). Information flow principles for plasticity in foraging robot swarms. *Swarm Intelligence*, 10(1):33–63.

- [144] Pitonakova, L., Crowder, R., and Bullock, S. (2018). The Information-Cost-Reward framework for understanding robot swarm foraging. *Swarm Intelligence*, 12(1):71–96.
- [145] Planqué, R., Van Den Berg, J. B., and Franks, N. R. (2010). Recruitment strategies and colony size in ants. *PLoS One*, 5(8):e11664.
- [146] Poissonnier, L. A., Motsch, S., Gautrais, J., Buhl, J., and Dussutour, A. (2019). Still flowing, experimental investigation of ant traffic under crowded conditions. *eLife*, in press.
- [147] Portha, S., Deneubourg, J.-L., and Detrain, C. (2004). How food type and brood influence foraging decisions of *Lasius niger* scouts. *Animal Behaviour*, 68(1):115–122.
- [148] Prasetyo, J., De Masi, G., and Ferrante, E. (2019). Collective decision making in dynamic environments. *Swarm Intelligence*, 13(3):217–243.
- [149] Prasetyo, J., De Masi, G., Ranjan, P., and Ferrante, E. (2018). The best-of-n problem with dynamic site qualities: Achieving adaptability with stubborn individuals. In *Swarm Intelligence (ANTS 2018)*, volume 11172 of *LNCS*, pages 239–251. Springer.
- [150] Pratisoli, F., Reina, A., Kaszubowski Lopes, Y., Sabattini, L., and Groß, R. (2019). A soft-bodied modular reconfigurable robotic system composed of interconnected Kilobots. In *Proceedings of the 2019 IEEE International Symposium on Multi-Robot and Multi-Agent Systems (MRS 2019)*, page in press.
- [151] Pratt, S. C. (2005). Quorum sensing by encounter rates in the ant *Temnothorax albipennis*. *Behavioral Ecology*, 16(2):488–496.
- [152] Pratt, S. C., Mallon, E., Sumpter, D. J. T., and Franks, N. (2002). Quorum sensing, recruitment, and collective decision-making during colony emigration by the ant *Leptothorax albipennis*. *Behavioral Ecology and Sociobiology*, 52(2):117–127.
- [153] Purnamadajaja, A. H. and Russell, R. A. (2007). Guiding robots’ behaviors using pheromone communication. *Autonomous Robots*, 23(2):113–130.
- [154] Purtan, R. R. P. and Udrea, A. (2013). A modified stochastic simulation algorithm for time-dependent intensity rates. In *Proceedings - 19th International Conference on Control Systems and Computer Science, CSCS 2013*, pages 365–369. IEEE.
- [155] Rahmani, P., Peruani, F., and Romanczuk, P. (2020). Flocking in complex environments—attention trade-offs in collective information processing. *PLOS Computational Biology*, 16(4):1–18.
- [156] Rapaport, D. C. (2004). *The Art of Molecular Dynamics Simulation*. Cambridge University Press.
- [157] Rausch, I., Reina, A., Simoens, P., and Khaluf, Y. (2019). Coherent collective behaviour emerging from decentralised balancing of social feedback and noise. *Swarm Intelligence*, 13(3–4):321–345.
- [158] Reid, C. R., Latty, T., and Beekman, M. (2012). Making a trail: Informed Argentine ants lead colony to the best food by U-turning coupled with enhanced pheromone laying. *Animal Behaviour*, 84(6):1579–1587.

- [159] Reina, A., Bose, T., Trianni, V., and Marshall, J. A. R. (2018). Effects of spatiality on value-sensitive decisions made by robot swarms. In *Proceedings of the 13th International Symposium on Distributed Autonomous Robotic Systems (DARS2016)*, volume 6 of *STAR*, pages 461–473. Springer.
- [160] Reina, A., Cope, A. J., Nikolaidis, E., Marshall, J. A. R., and Sabo, C. (2017a). ARK: Augmented Reality for Kilobots. *IEEE Robotics and Automation Letters*, 2(3):1755–1761.
- [161] Reina, A., Dorigo, M., and Trianni, V. (2014a). Collective decision making in distributed systems inspired by honeybees behaviour. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems, AAMAS '14*, page 1421–1422.
- [162] Reina, A., Dorigo, M., and Trianni, V. (2014b). Towards a cognitive design pattern for collective decision-making. In *Swarm Intelligence (ANTS 2014)*, LNCS, pages 194–205.
- [163] Reina, A., Marshall, J. A. R., and Bose, T. (2016). Swarm deadlock for symmetric choices over three or more options. In *The 14th International Conference on the Simulation of Adaptive Behavior (SAB)*, pages 1–2.
- [164] Reina, A., Marshall, J. A. R., Trianni, V., and Bose, T. (2017b). Model of the best-of-N nest-site selection process in honeybees. *Physical Review E*, 95(5):052411.
- [165] Reina, A., Miletitch, R., Dorigo, M., and Trianni, V. (2015a). A quantitative micro-macro link for collective decisions: The shortest path discovery/selection example. *Swarm Intelligence*, 9(2–3):75–102.
- [166] Reina, A., Salvaro, M., Francesca, G., Garattoni, L., Pinciroli, C., Dorigo, M., and Birattari, M. (2015b). Augmented reality for robots: Virtual sensing technology applied to a swarm of e-pucks. In *Proceedings of the 2015 NASA/ESA Conference on Adaptive Hardware and Systems (AHS 2015)*, pages 1–6. IEEE.
- [167] Reina, A., Valentini, G., Fernández-Oto, C., Dorigo, M., and Trianni, V. (2015c). A design pattern for decentralised decision making. *PLoS ONE*, 10(10):e0140950.
- [168] Ritchie, M. (2009). *Scale, Heterogeneity, and the Structure and Diversity of Ecological Communities*. Princeton University Press.
- [169] Robinson, E. J., Smith, F. D., Sullivan, K. M., and Franks, N. R. (2009). Do ants make direct comparisons? *Proceedings of the Royal Society B: Biological Sciences*, 276(1667):2635–2641.
- [170] Robinson, E. J. H., Franks, N. R., Ellis, S., Okuda, S., and Marshall, J. A. R. (2011). A simple threshold rule is sufficient to explain sophisticated collective decision-making. *PLoS ONE*, 6(5):e19981.
- [171] Robinson, E. J. H., Jackson, D. E., Holcombe, M., and Ratnieks, F. L. W. (2005). ‘no entry’ signal in ant foraging. *Nature*, 438(7067):442–442.
- [172] Robinson, E. J. H., Ratnieks, F. L., and Holcombe, M. (2008). An agent-based model to investigate the roles of attractive and repellent pheromones in ant decision making during foraging. *Journal of Theoretical Biology*, 255(2):250–258.

- [173] Rohmer, E., Singh, S. P. N., and Freese, M. (2013). V-REP: A versatile and scalable robot simulation framework. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1321–1326.
- [174] Rubenstein, M., Ahler, C., Hoff, N., Cabrera, A., and Nagpal, R. (2014). Kilobot: A low cost robot with scalable operations designed for collective behaviors. *Robotics and Autonomous Systems*, 62(7):966–975.
- [175] Rubenstein, M., Cabrera, A., Werfel, J., Habibi, G., McLurkin, J., and Nagpal, R. (2013). Collective transport of complex objects by simple robots: Theory and experiments. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013)*, pages 47–54. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC.
- [176] Schall, J. D. and Hanes, D. P. (1993). Neural basis of saccade target selection in frontal eye field during visual search. *Nature*, 366(6454):467–469.
- [177] Scheidler, A., Brutschy, A., Ferrante, E., and Dorigo, M. (2016). The k-unanimity rule for self-organized decision-making in swarms of robots. *IEEE Transactions on Cybernetics*, 46(5):1175–1188.
- [178] Schlick, T. (2010). *Molecular Modeling and Simulation: An Interdisciplinary Guide*. Springer.
- [179] Seeley, T. D. and Buhrman, S. C. (2001). Nest-site selection in honey bees: how well do swarms implement the “best-of-n” decision rule? *Behavioral Ecology and Sociobiology*, 49(5):416–427.
- [180] Seeley, T. D. and Visscher, P. K. (2004). Quorum sensing during nest-site selection by honeybee swarms. *Behavioral Ecology and Sociobiology*, 56(6):594–601.
- [181] Seeley, T. D., Visscher, P. K., Schlegel, T., Hogan, P. M., Franks, N. R., and Marshall, J. A. R. (2012). Stop signals provide cross inhibition in collective decision-making by honeybee swarms. *Science*, 335(6064):108–111.
- [182] Shaffer, Z., Sasaki, T., and Pratt, S. C. (2013). Linear recruitment leads to allocation and flexibility in collective foraging by ants. *Animal Behaviour*, 86(5):967–975.
- [183] Shklarsh, A., Ariel, G., Schneidman, E., and Ben-Jacob, E. (2011). Smart swarms of bacteria-inspired agents with performance adaptable interactions. *PLoS Computational Biology*, 7(9).
- [184] Shucker, B. and Bennett, J. K. (2007). Scalable Control of Distributed Robotic Macrosensors. In *Distributed Autonomous Robotic Systems 6*, pages 379–388.
- [185] Sitti, M. (2004). *Mobile Microrobotics*. The MIT Press.
- [186] Soleymani, T., Trianni, V., Bonani, M., Mondada, F., and Dorigo, M. (2015). Bio-inspired construction with mobile robots and compliant pockets. *Robotics and Autonomous Systems*, 74:340 – 350.

- [187] Soorati, M. D., Krome, M., Mora-Mendoza, M., Ghofrani, J., and Hamann, H. (2019). Plasticity in collective decision-making for robots: Creating global reference frames, detecting dynamic environments, and preventing lock-ins. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4100–4105.
- [188] Soysal, O. and Sahin, E. (2005). Probabilistic aggregation strategies in swarm robotic systems. In *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.*, pages 325–332.
- [189] Soysal, O. and Şahin, E. (2007). A macroscopic model for self-organized aggregation in swarm robotic systems. In *Swarm Robotics*, pages 27–42.
- [190] Spears, W. M., Spears, D. F., Hamann, J. C., and Heil, R. (2004). Distributed, physics-based control of swarms of vehicles. *Autonomous Robots*, 17(2):137–162.
- [191] Sperati, V., Trianni, V., and Nolfi, S. (2011). Self-organised path formation in a swarm of robots. *Swarm Intelligence*, 5(2):97–119.
- [192] Stickland, T. R., Britton, N. F., and Franks, N. R. (1999). Models of information flow in ant foraging: The benefits of both attractive and repulsive signals. In *Information Processing in Social Insects*, pages 83–100. Birkhäuser Basel.
- [193] Sugawara, K., Kazama, T., and Watanabe, T. (2004). Foraging behavior of interacting robots with virtual pheromone. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, volume 3, pages 3074–3079. IEEE.
- [194] Sumpter, T. and Pratt, C. (2003). A modelling framework for understanding social insect foraging. *Behavioral Ecology and Sociobiology*, 53(3):131–144.
- [195] Svennebring, J. and Koenig, S. (2004). Building terrain-covering ant robots: A feasibility study. *Autonomous Robots*, 16(3):313–332.
- [196] Tabone, M., Ermentrout, B., and Doiron, B. (2010). Balancing organization and flexibility in foraging dynamics. *Journal of Theoretical Biology*, 266(3):391–400.
- [197] Talamali, M. S., Bose, T., Haire, M., Xu, X., Marshall, J. A. R., and Reina, A. (2020). Sophisticated collective foraging with minimalist agents: a swarm robotics test. *Swarm Intelligence*, 14(1):25–56.
- [198] Talamali, M. S., Marshall, J. A. R., Bose, T., and Reina, A. (2019). Improving collective decision accuracy via time-varying cross-inhibition. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9652–9659.
- [199] Thienen, W. V., Metzler, D., Choe, D. H., and Witte, V. (2014). Pheromone communication in ants: A detailed analysis of concentration-dependent decisions in three species. *Behavioral Ecology and Sociobiology*, 68(10):1611–1627.
- [200] Thomas D. Seeley and P. Kirk Visscher (2004). Group decision making in nest-site selection by honey bees. *Apidologie*, 35(2):101–116.

- [201] Torney, C. J., Neufeld, Z., and Couzin, I. D. (2009). Context-dependent interaction leads to emergent search behavior in social aggregates. *Proceedings of the National Academy of Sciences*, 106(52):22055–22060.
- [202] Trianni, V., Groß, R., Labella, T. H., Şahin, E., and Dorigo, M. (2003). Evolving aggregation behaviors in a swarm of robots. In *Advances in Artificial Life*, pages 865–874.
- [203] Truszkowski, W. F., Hinchey, M. G., Rash, J. L., and Rouff, C. A. (2006). Autonomous and autonomic systems: a paradigm for future space exploration missions. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 36(3):279–291.
- [204] Tsimring, L. S. (2014). Noise in biology. *Reports on Progress in Physics*, 77(2):026601.
- [205] Turgut, A. E., Çelikkanat, H., Gökçe, F., and Şahin, E. (2008). Self-organized flocking in mobile robot swarms. *Swarm Intelligence*, 2(2):97–120.
- [206] Valentini, G. (2017). *Achieving Consensus in Robot Swarms*, volume 706 of *Studies in Computational Intelligence*. Springer International Publishing.
- [207] Valentini, G., Antoun, A., Trabatttoni, M., Wiandt, B., Tamura, Y., Hocquard, E., Trianni, V., and Dorigo, M. (2018). Kilogrid: A novel experimental environment for the Kilobot robot. *Swarm Intelligence*, 12(3):245—266.
- [208] Valentini, G., Brambilla, D., Hamann, H., and Dorigo, M. (2016a). Collective perception of environmental features in a robot swarm. In *Swarm Intelligence (ANTS2016)*, LNCS, pages 65–76. Springer.
- [209] Valentini, G., Ferrante, E., and Dorigo, M. (2017). The Best-of-n Problem in Robot Swarms: Formalization, State of the Art, and Novel Perspectives. *Frontiers in Robotics and AI*, 4:9.
- [210] Valentini, G., Ferrante, E., Hamann, H., and Dorigo, M. (2016b). Collective decision with 100 Kilobots: speed versus accuracy in binary discrimination problems. *Autonomous Agents and Multi-Agent Systems*, 30(3):553–580.
- [211] Valentini, G., Hamann, H., and Dorigo, M. (2014). Self-organized collective decision making: The weighted voter model. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems, AAMAS '14*, page 45–52. International Foundation for Autonomous Agents and Multiagent Systems.
- [212] Valentini, G., Hamann, H., and Dorigo, M. (2015). Efficient decision-making in a self-organizing robot swarm: On the speed versus accuracy trade-off. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, volume 2, pages 1305–1314.
- [213] Van Vorhis Key, S. E. and Baker, T. C. (1982). Trail-following responses of the Argentine ant, *Iridomyrmex humilis* (Mayr), to a synthetic trail pheromone component and analogs. *Journal of Chemical Ecology*, 8(1):3–14.
- [214] Vigelius, M., Meyer, B., and Pascoe, G. (2014). Multiscale modelling and analysis of collective decision making in swarm robotics. *PLoS ONE*, 9(11):e111542.

- [215] Visscher, P. K. and Camazine, S. (1999). Collective decisions and cognition in bees. *Nature*, 397(6718):400–400.
- [216] Wendt, S., Strunk, K. S., Heinze, J., Roider, A., and Czaczkes, T. J. (2018). Relative value perception in an insect: Positive and negative incentive contrasts in ants. *bioRxiv*, page 330241.
- [217] Werger, B. B. and Matarić, M. J. (1996). Robotic "food" chains: Externalization of state and program for minimal-agent foraging. In *From Animals to Animats 4. Proceedings of the 4th international conference on Simulation of Adaptive Behavior (SAB 96)*, pages 625–634. MIT Press.
- [218] Wessnitzer, J. and Melhuish, C. (2003). Collective decision-making and behaviour transitions in distributed Ad Hoc wireless networks of mobile robots: Target-hunting. In *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, volume 2801, pages 893–902.
- [219] Wilson, E. O. (1962). Chemical communication among workers of the fire ant *Solenopsis saevissima* (Fr. Smith) 1. The Organization of Mass-Foraging. *Animal Behaviour*, 10(1-2):134–147.
- [220] Winfield, A. F. T. (2009). Foraging robots. In *Encyclopedia of Complexity and System Science*, pages 3682–3700. Springer.
- [221] Yang, G.-Z., Bellingham, J., Dupont, P. E., Fischer, P., Floridi, L., Full, R., Jacobstein, N., Kumar, V., McNutt, M., Merrifield, R., Nelson, B. J., Scassellati, B., Taddeo, M., Taylor, R., Veloso, M., Wang, Z. L., and Wood, R. (2018). The grand challenges of science robotics. *Science Robotics*, 3(14).
- [222] Yang, J., Zhang, C., Wang, X., Wang, W., Xi, N., and Liu, L. (2019). Development of micro- and nanorobotics: A review. *Science China Technological Sciences*, 62(1):1–20.
- [223] Yun, S.-k., Schwager, M., and Rus, D. (2011). Coordinating Construction of Truss Structures Using Distributed Equal-Mass Partitioning. In *Robotics Research*, volume 70 of *STAR*, pages 607–623.
- [224] Zhang, S. (2013). *Handbook of 3D machine vision : optical metrology and imaging*. CRC Press.
- [225] Zhang, S. (2016). *High-Speed 3D Imaging with Digital Fringe Projection Techniques*. CRC Press.