

The Information-Cost-Reward framework for understanding robot swarm foraging

Lenka Pitonakova¹  · Richard Crowder¹ · Seth Bullock²

Received: 7 April 2017 / Accepted: 9 November 2017
© The Author(s) 2017. This article is an open access publication

Abstract Demand for autonomous swarms, where robots can cooperate with each other without human intervention, is set to grow rapidly in the near future. Currently, one of the main challenges in swarm robotics is understanding how the behaviour of individual robots leads to an observed emergent collective performance. In this paper, a novel approach to understanding robot swarms that perform foraging is proposed in the form of the Information-Cost-Reward (ICR) framework. The framework relates the way in which robots obtain and share information (about where work needs to be done) to the swarm's ability to exploit that information in order to obtain reward efficiently in the context of a particular task and environment. The ICR framework can be applied to analyse underlying mechanisms that lead to observed swarm performance, as well as to inform hypotheses about the suitability of a particular robot control strategy for new swarm missions. Additionally, the information-centred understanding that the framework offers paves a way towards a new swarm design methodology where general principles of collective robot behaviour guide algorithm design.

Keywords Swarm robotics · Foraging · Modelling · Information flow

1 Introduction

Demand for autonomous multi-robot systems is set to grow rapidly in the near future. Considerable effort is currently being invested in the design of fleets of self-driving taxis (e.g. Griswold 2016), delivery robots (e.g. Amazon Prime Air 2016), autonomous agricultural robots (e.g. Cartade et al. 2012) and automated warehouses (e.g. Stiefelhagen et al. 2004). In

Electronic supplementary material The online version of this article (<https://doi.org/10.1007/s11721-017-0148-3>) contains supplementary material, which is available to authorized users.

✉ Lenka Pitonakova
contact@lenkaspaces.net

¹ Department of Electronics and Computer Science, University of Southampton, Southampton, UK

² Department of Computer Science, University of Bristol, Bristol, UK

order to become pervasive, these and other robots will need to coordinate and cooperate with each other while fulfilling on-demand tasks. While many swarm robotic experiments have been performed both in simulation and in the real world, a framework that organises results and explains how the behaviour of individual robots affects emergent swarm performance is still missing. It is therefore often unclear what type of robot control algorithm is suitable in a given swarm mission and which design decisions can be generalised to other missions. To this end, several authors have proposed that an information-based understanding of swarm behaviour can help us to generalise experimental results (e.g. Wang et al. 2012; Miller et al. 2014) and that paying attention to feedback loops between the swarm and the environment is important (e.g. Gardelli et al. 2007; Fernandez-Marquez et al. 2013).

In this paper, the *Information-Cost-Reward (ICR) framework* is developed, providing a high-level, information-based account of robot swarm behaviour during foraging. The framework defines several metrics that allow for a detailed analysis of mechanisms that lead to observed swarm performance. *Scouting efficiency* and *information gain rate* characterise information flow within the swarm, i.e. how well robots can acquire and share information and decrease the swarm's *uncertainty cost*. *Displacement* and *misinformation* costs are associated with the ability of the swarm to turn information into reward.

There are two main contributions of our research. First, we compare three robot control strategies, commonly used in the swarm robotics research, in a number of experimental scenarios and identify swarm missions in which each strategy leads to the best swarm performance. Second, we demonstrate how the ICR framework can be used to understand swarm performance and to identify aspects of robot behaviour that affect it negatively, and how the framework can be used to explain and form hypotheses regarding the relative performance of swarms that employ different control strategies.

This paper is organised as follows. Section 2 provides an account of the literature on swarm analysis and modelling methods. Section 3 introduces our simulation environment, robot control strategies, performance metrics and visualisation methods. The performance of robot swarms is explored initially in Sect. 4, where a simple foraging mission is considered. In this mission, worksites are “consumed” when encountered by robots and worksite locations remain constant during an experimental run. The information flow and cost analysis are presented in Sects. 5 and 6. The ICR framework is formulated in Sect. 7 and is employed to make sense of swarm behaviour in static central-place foraging and in dynamic foraging missions, where worksite locations change over time, in Sects. 8 and 9, respectively. Finally, the experimental results and the ICR framework are discussed more broadly in Sect. 10.

2 Background

Mathematical modelling approaches have been used in order to study the effect of system parameters on robot behaviour and swarm performance. These approaches include, for instance, “property-driven design” (Brambilla et al. 2014), “swarm calculus” (Hamann 2013), and other models based on probabilistic finite state machines (PFMSs) (e.g. Liu and Winfield 2010; Reina et al. 2015a) and differential equations (DEs) (e.g. Montes de Oca et al. 2011; Mather and Hsieh 2012; Reina et al. 2015b; Scheidler et al. 2016; Valentini et al. 2016). A comprehensive overview of these methods can be found in Brambilla et al. (2013). PFMSs are microscopic, time-discrete, models that describe how the robot population probabilistically transitions between possible “states” (e.g. “scouting” and “foraging”) in discrete time intervals. The state transition rates can be calculated in order to link a PFMS model to a

macroscopic model expressed in terms of DEs. The transition rates are often subject to various parameters. Parameters that the swarm designer has no control over, e.g. the effect of interference between robots, are estimated or measured by performing targeted experiments. Other parameters, such as recruitment time, can be controlled and their effect on swarm performance can be studied. A related modelling method is Turing Learning (Li et al. 2016), where the neural network architecture of agents can be inferred automatically through a co-evolution of models and their classifiers.

Swarm entropy (Sperati et al. 2011) has been used in order to quantify how much order there is in the observed behaviour of robots. The measure of entropy is inspired by Shannon's information theory (Shannon 1948) and characterises the probability of each robot being in one of N possible discrete states at a given point in time. For example, in the context of a transport task between two areas, robots needed to form chains when moving from one area to another (Sperati et al. 2011). The heading of robots was discretised into four possible directions. Entropy (i.e. "disorder") of the swarm was high when robots moved randomly, and it was low when robots synchronised their heading directions, i.e. when they formed chains. Similarly, *hierarchical social entropy* (Balch 2000) has been applied to measure the extent to which robots formed coherent clusters in a coordinated motion task (Ducatelle et al. 2014). Finally, *local transfer entropy (LTE)* has been used in order to measure information flow in robot swarms in the context of coordinated motion (Wang et al. 2012; Miller et al. 2014). LTE measures the correlation between a previous "state" (e.g. current velocity) of a source robot that holds some information, and the next "state" of a destination robot that reads the information. Positive LTE thus corresponds to coordinated motion in a swarm.

Despite providing valuable insights into swarm behaviour, these approaches have a number of drawbacks. Models based on PFMSs and DEs are useful when the selected robot behaviour needs to be parametrised for a specific experiment. However, because their role is to estimate swarm performance, given specific conditions or parameter values, they cannot *explain* swarm behaviour in a way that would allow us to learn something general about the algorithms used. On the other hand, measures of entropy can, to various degrees, describe more general properties of swarm behaviour, such as information flow between robots, or their tendency to coordinate their actions. However, to the best of our knowledge, no study has demonstrated so far how and whether entropy can be directly related to swarm performance in foraging-like missions, where the performance is not only dependent on the ability of robots to observe each other's actions and to coordinate their behaviour, but also on the structure of the environment and on interference between robots.

3 Methods

3.1 Simulation environment and swarm missions

All experiments are performed in the ARGoS simulation environment (Pinciroli et al. 2012).¹ The simulation takes place in continuous space and it is updated 10 times per second. The experimental arena contains a centrally located circular base surrounded by worksites (Fig. 1a, b). A similar setup has been previously used in, e.g. Balch and Arkin (1994), Pitonakova et al. (2016b), Gutiérrez et al. (2010).

The base has a radius of 3 m and is divided into two sections: an interior *recruitment area* and an *unloading area* around it (Fig. 1c). There is a light source placed above the middle of

¹ We discuss the advantages of using this simulation engine in Pitonakova et al. (2016a).

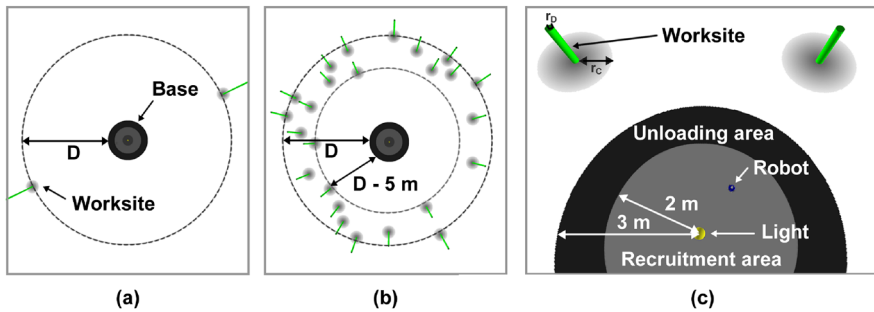


Fig. 1 a, b ARGoS simulation screenshot of the experimental arena containing a base in the centre and worksites in the a Heap2 and b Scatter25 scenarios with worksite distance $D = 13$ m. c A close-up screenshot of a base and nearby worksites

the base that the robots use as a reference for navigation towards and away from the centre of the base (as in, e.g. Krieger and Billeter 2000; Pini et al. 2013).

There are two types of scenario: (Fig. 1a, b):

- **Heap N_W** $N_W \in \{1, 2, 4\}$ high-volume worksites distributed evenly around the base at a distance D from the base edge
- **Scatter25** $N_W = 25$ worksites randomly distributed between distance D and $D - 5$ m from the base edge

The total amount of reward in each scenario is set to 100 and the amount of reward per worksite, $V = 100/N_W$. For example, each worksite in a Heap2 scenario has $V = 50$, while worksites in the Scatter25 scenario have $V = 4$. Each scenario is investigated with five different values for worksite distance, $D \in \{5, 9, 13, 17, 21\}$ m, from the base edge. The worksites are cylindrical and have a radius $r_D = 0.1$ m. In order to enable robots close to a worksite to move towards it, a colour gradient with radius $r_C = 1$ m is centred on the floor around each worksite (see Fig. 1c).

Two basic types of foraging mission are investigated in each scenario:

- **Consumption** worksites represent “tasks” or “jobs” that need to be completed by robots as soon as possible. When a robot is near a worksite, it gradually depletes the worksites’s volume, increasing the swarm’s total reward by a *reward gain rate* $\rho = 1/400$ units per second. This type of mission is analogous to the “consume” mission explored by Balch and Arkin (1994), the “task allocation” problem (e.g. Mataric et al. 2003; Lerman et al. 2006; Jevtic et al. 2012, or the “job completion” problem on a manufacturing floor (e.g. Gerkey and Mataric 2003; Dahl et al. 2009; Sarker and Dahl 2011).
- **Collection** worksites represent resource deposits that need to be exploited as soon as possible. A robot can collect a maximum of one unit of volume of resource at a time. A robot takes one second to load the resource and returns to the base in order to unload it. Reward is obtained in the base after one second of unloading, i.e. $\rho = 1$ unit per second. The robot then returns to the worksite in order to continue foraging from it, provided that the worksite has not been previously depleted. Similar missions were explored e.g. in Krieger and Billeter (2000), Lemmens et al. (2008), Ducatelle et al. (2011).

Worksites in both missions are depleted faster when more robots work on them at the same time. The loading rates for the two missions are set so that worksites take a similar average time to deplete during both Consumption and Collection, and when different experimental environments are considered.

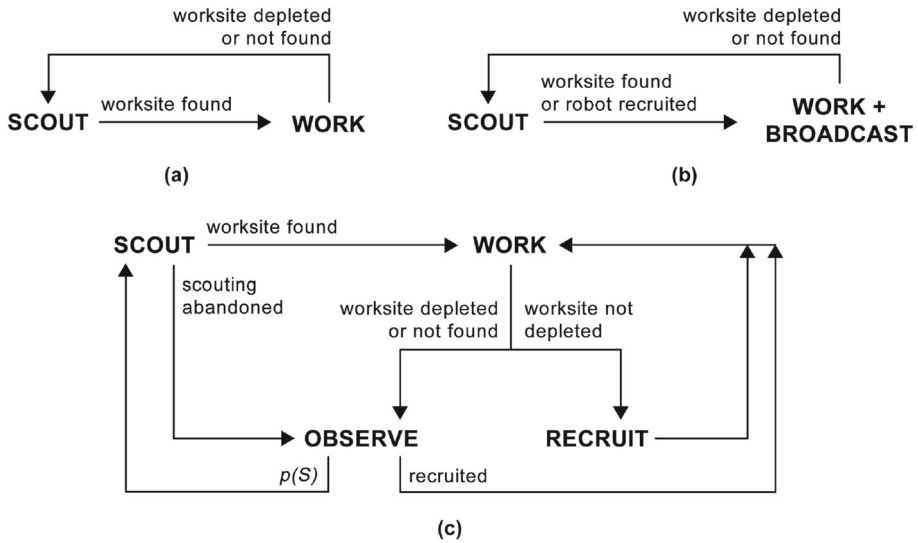


Fig. 2 Finite state machine representation of a **a** solitary, **b** local broadcaster, **c** bee robot

3.2 Robots and robot control strategies

The simulated MarXbots (Bonani et al. 2010) are circular robots with a radius of 8.5 cm.² Each experiment is performed with three types of homogeneous robot swarms that represent three different robot control strategies commonly used in the swarm robotics literature and used here to highlight the ICR framework. The control strategies, described below, are implemented as finite state machines and visualised in Fig. 2.

- **Solitary** (also in, e.g. Labella et al. 2006; Yang et al. 2009): robots search the environment for worksites as “scouts” and become “workers” when they discover a worksite. They do not communicate worksite locations to each other.
- **Local broadcaster** (also in, e.g. Gutiérrez et al. 2010; Wawerla and Vaughan 2010; Ducatelle et al. 2014): workers broadcast information about a worksite that they are currently working on to scouts that are nearby in order to recruit them.
- **Bee** (also in, e.g. Krieger and Billeter 2000; Pitonakova et al. 2014; Hecker and Moses 2015; Reina et al. 2015a): robots meet in the base in order to exchange information about worksites. A robot located in the base can be in one of two states: “recruiter” or “observer”. A recruiter knows a worksite’s location and spends a certain amount of time in the base in order to recruit observers to its worksite. Observers that are not recruited have a small probability, $p(S)$, to leave the base in order to become scouts. Scouts that are unable to find worksites after a certain period of time abandon scouting and return to the base to become observers.

The swarms are fully decentralised, and any communication between the robots, if applicable to their control strategy, happens locally, using the range and bearing module with a signal range of approximately 5 m. Each robot utilises odometry in order to keep track of the location of its current worksite. Odometry errors may occur as a result of minor differential steering sensor noise and wheel slippage. See online supplementary material, Section S1, for additional details about the robot control strategies and their parameters.

² We have previously described the robots in detail in Pitonakova et al. (2016a).

3.3 Performance metrics and visualisation

When swarm performance is considered, a control strategy that performs significantly better than others is referred to as the *winning strategy*. In static environments, a strategy that makes the swarm deplete all worksites the quickest, and thus has the *lowest* mission completion time, is considered a winning strategy. In dynamic environments, where worksites spontaneously disappear, a winning strategy has the *highest* total reward. Statistical significance is determined by using Tukey's honest significant difference (HSD) test (Tukey 1949) in conjunction with ANOVA, with statistical significance level $p = 0.01$. Each performance metric is based on 50 independent simulation runs with different random seeds.

Winning strategies are visualised in matrix plots (e.g. Fig. 3), where each grid cell represents a mission scenario as a combination of a particular number of worksites and worksite distance from the base. The grid cell colour represents a winning strategy in that scenario. On some occasions, there may be more than one winning strategy, provided that the differences between them are not statistically significant but at least one of them is significantly better than the third remaining strategy. When there are no statistically significant differences between any control strategies in a given experiment, all are considered to be winning strategies. Multiple winning strategies are represented as multiple coloured boxes in a single matrix plot cell.

Additionally, box plots (e.g. Fig. 4) are utilised for comparing characteristics of multiple control strategies. A middle horizontal line of a box plot represents a median value of a set of results. The line is surrounded by a box, representing the inter-quartile range or "middle fifty" of the result set, and whiskers representing data in the range of 1.5 times the inter-quartile range, with outliers outside this range shown as plus signs (Matplotlib: Box plots 2017).

4 Consumption in static environments

The relative performances of the control strategies explored here are dependent on the properties of the experimental scenario being simulated (Fig. 3). Solitary swarms complete the Consumption mission faster than the other swarms in the least difficult scenarios, where worksites are numerous and thus easy to find, i.e. in the Scatter25 scenarios with worksites that are close to the base. In these environments, recruitment, utilised by both local broadcasters and bee swarms leads to a strong commitment of multiple robots to a single worksite, causing *physical interference* between robots and preventing them from accessing worksites and from moving around the environment. Additionally, bee swarms also experience *exploitational interference*, where robots in the base recruit observers to worksites that have already been depleted by others.³ Local broadcasters rarely suffer from this type of interference, since they recruit near worksites and the recruiters thus have more up-to-date information about whether their worksites still have some reward in them. The disadvantage of local broadcasters and bee swarms in less difficult scenarios is greater when there are more robots in the swarm, i.e. when the interference between robots is stronger.

On the other hand, Heap1 scenarios represent the most difficult environments, where only a single worksite exists and reward is thus difficult to obtain. Local broadcasters and bee swarms outperform the solitary swarms in most Heap1 environments, since the robots can share information about where reward is located. Finally, in the intermediate Heap2 and Heap4 scenarios, bee swarms generally cannot perform as well as the other swarms,

³ Exploitational interference was investigated in detail in Pitonakova et al. (2014) as "environmental" interference.

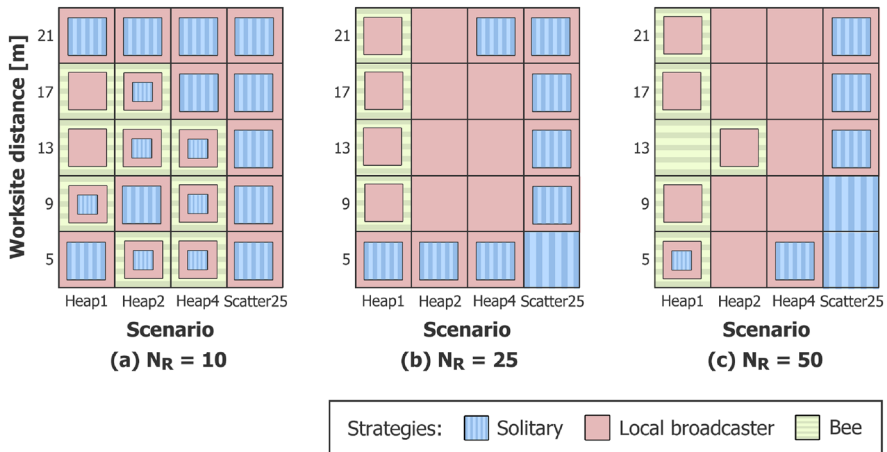


Fig. 3 Winning strategies that complete the static Consumption mission the fastest in different environments using **a** 10, **b** 25 and **c** 50 robots. Box plots of these results are shown in online supplementary material, Figures S2.1–S2.3

both due to exploitative interference and due to the fact that the bee robots have to spend additional time travelling to the base to recruit. In these environments, solitary swarms and local broadcasters perform similarly well when D is small, while more difficult environments with large D favour local broadcasters.

5 Information flow analysis

Analysing when and how information is acquired by a swarm and how it spreads between robots is the first step towards understanding why some control strategies are more suitable than others in a given environment. In this section, two swarm characteristics that characterise its information flow are introduced: *the scouting efficiency* and *the information gain rate*.

5.1 Scouting efficiency

A swarm’s scouting efficiency can be approximated by measuring the time of the first worksite discovery in a given experimental run. The longer it takes a swarm to discover its first worksite, the worse scouting efficiency it has. Note that the time of the *first*, rather than of the *last* or *median* worksite discovery, is evaluated, as it is the least affected by interference between robots.

All swarms are less efficient at scouting in environments where worksites are far away from the base, since they have a larger area to search. However, the scouting efficiency of bee swarms is affected more significantly by worksite distance than that of other swarms, since bee scouts periodically return to the base in order to check whether there are any recruiters there, which limits the amount of time that they spend scouting. The inefficient scouting behaviour of bee swarms is most obvious in the difficult Heap1 environments (see Fig. 4 for results from Heap1 environments and online supplementary material, Figures S2.4–S2.6, for results from other environments).

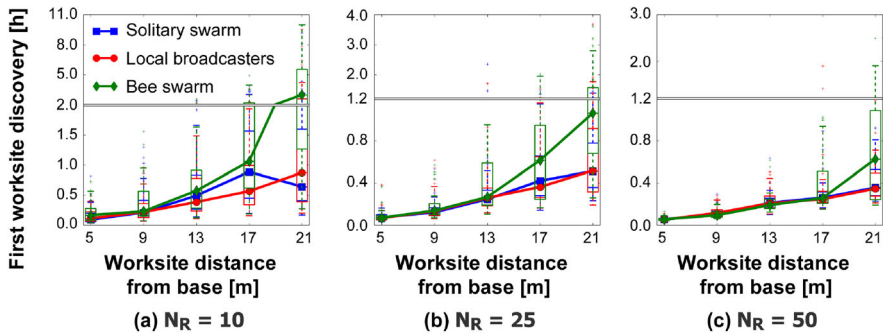


Fig. 4 Time of the first worksite discovery in the static Consumption mission, Heap1 environments, using **a** 10, **b** 25 and **c** 50 robots. The negative effect of large worksite distance, D , is larger in bee swarms, compared to the other control strategies. Note the different scales of the y-axis for different values of N_R

5.2 Information gain

The amount of information that the swarm has at a given point in time is defined as:

$$I(t) = \sum_W^{N_A} S_W(t) \quad (1)$$

where N_A is the number of *active* (i.e. not depleted) worksites in the environment and $S_W(t)$ is the number of *subscribed* robots that know the location of a worksite W at time t . The information gain,⁴ ΔI , of a swarm represents the change in I and it is defined as:

$$\begin{aligned} \Delta I(t) &= I(t) - I(t-1) \\ &= \sum_W^{N_A} [S_W(t) - S_W(t-1)] \end{aligned} \quad (2)$$

Finally, we can obtain normalised information gain, $\Delta I(t)'$, by dividing ΔI by the number of robots, N_R :

$$\Delta I(t)' = \frac{\Delta I(t)}{N_R} \quad (3)$$

By measuring information gain, we can identify when scouts find new worksites or when robots are recruited for work. In these cases, a swarm gains new information and ΔI is positive. Similarly, when robots abandon active worksites, they no longer know their locations and they have a negative information gain. When no information is gained or lost, $\Delta I = 0$.

For example, in Scatter25, where it is relatively easy to discover worksites as they are numerous, all swarms generate a large information gain, especially at the beginning of experimental runs, when scouting is the most successful. Solitary robots (Fig. 5a) maintain $\Delta I' > 0$ until all worksites are depleted, as the robots are rather evenly spread across the work arena and thus suffer from minimal interference. The region of the graph during which $\Delta I' > 0$ is referred to as a *positive information gain region*. Solitary swarms have a single positive information gain region. On the other hand, bee robots (Fig. 5b) learn about worksites in a

⁴ Information gain is a more general metric inspired by *information value* that has been formalised in Pitonakova et al. (2016a)

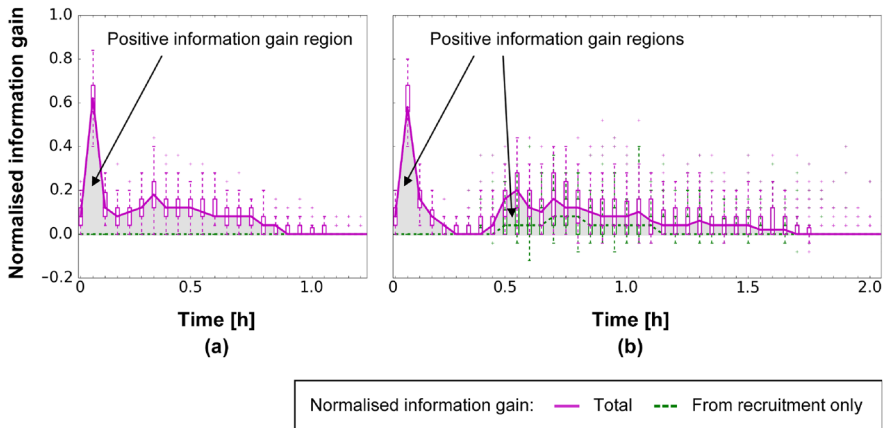


Fig. 5 Normalised information gain of **a** 25 solitary robots, **b** 25 bee robots in the static Consumption mission, Scatter25 scenario with $D = 9$ m

solitary fashion at the beginning of each run, and recruit each other in the base later, and therefore have two positive information gain regions in this scenario.

5.3 Information gain rate

Information gain rate, i , characterises how quickly positive information gain regions of a swarm can grow, in other words, how good robots are in discovering worksites and sharing worksite locations with each other.

In order to calculate i , the normalised information gain time series is first compressed into time intervals T_i seconds long by summing the individual values of $\Delta I(t)'$ in each T_i interval. Compressing the data this way makes it possible to identify and measure trends in $\Delta I'$, since individual information gain events, such as a robot finding a worksite or a robot being recruited, that usually occur a few seconds apart, are grouped together into discrete time intervals. Positive regions are then distinguished from the rest of the compressed time series by considering intervals during which the compressed information gain, ΔI^* , remains positive. T_i is a parameter to the information gain rate calculation, set to $T_i = 60$ s. Using a different value does not affect the order of the swarms based on i (see online supplementary material, Section S3).

Information gain rate of each positive region, i_p , is defined as:

$$i_p = \frac{\sum_{T=0}^{T_p} \Delta I^*(T)}{T_p} \tag{4}$$

where T_p is the length of a positive region in seconds and $\Delta I(T)^*$ is the total compressed information gain in a given time interval. The information gain rate of a swarm i is the maximum⁵ value of i_p measured in an experimental run:

$$i = \max(i_p) \tag{5}$$

⁵ The *maximum*, rather than *median* or *average* i_p is used, because it characterises the best-case increase in information gain, and it is thus less likely to be affected by physical and exploitative interference between robots. In other words, i represents the upper limit of how quickly robots can gain information in a given environment.

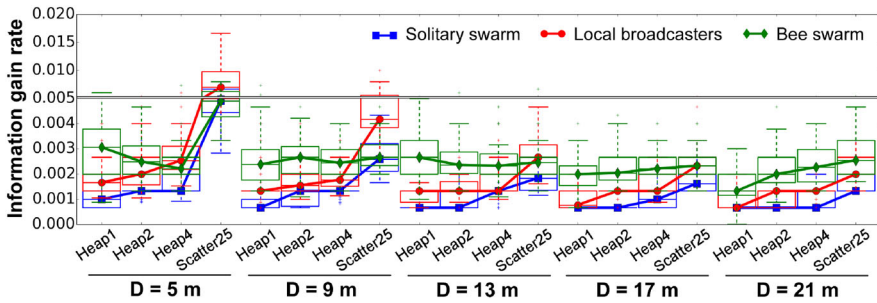


Fig. 6 Information gain rate, i , of 25-robot swarms in the static Consumption mission. Bee swarms enjoy the highest i in most environments, apart from the Scatter25 scenarios when $D \leq 9$ m

In solitary swarms, worksite discoveries are more probable when worksites are abundant or when the work arena is small. Consequently, the information gain rate of solitary swarms is high when the number of worksites, N_W is high and when worksite distance, D , is small (Fig. 6). A similar trend can be observed for local broadcasters. On the other hand, the information gain rate of bee swarms varies less across scenarios and generally does not increase with N_W , unless the worksites are far away from the base ($D = 21$ m) or when swarms are small ($N_R = 10$, see online supplementary material, Figure S2.7). Since bee robots share information in the base, they are able to achieve a relatively high information gain rate in difficult environments like Heap1. However, their information gain rate usually cannot increase further in less difficult environments due to interference between robots. Similar trends can be observed for 50-robot swarms (see online supplementary material, Figure S2.8.).

6 Cost analysis

A swarm that does not know where worksites are located pays a cost associated with this uncertainty—the robots are roaming the environment instead of earning reward. However, it is apparent from the comparison between information flow and mission completion time of swarms that even if robots can obtain information about where worksites are, this information might be difficult to utilise. For example, even though bee swarms achieve the highest i in the Heap environments, they very rarely outperform local broadcasters (Fig. 3). On the other hand, solitary swarms, that achieve the lowest information gain rate in all environments, are able to complete the Consumption mission faster than any other swarm when worksites are easy to find.

We propose here that different swarms pay different *costs* for both *lacking information* and for *exploiting* information. Note that unlike in the optimal foraging literature, where “costs” usually represent the energy costs of behaviours (Charnov 1976; Fagen 1987), all costs are quantified here as proportional to the amount of *reward* that cannot be obtained by robots. Using this representation of costs makes it possible to precisely identify stages of a robot’s work cycle that prevent the robot from obtaining reward and to mathematically relate a swarm’s information flow to the swarm’s performance. In this section, the work cycle of a robot is first described, and the costs paid at each stage of the work cycle are defined.

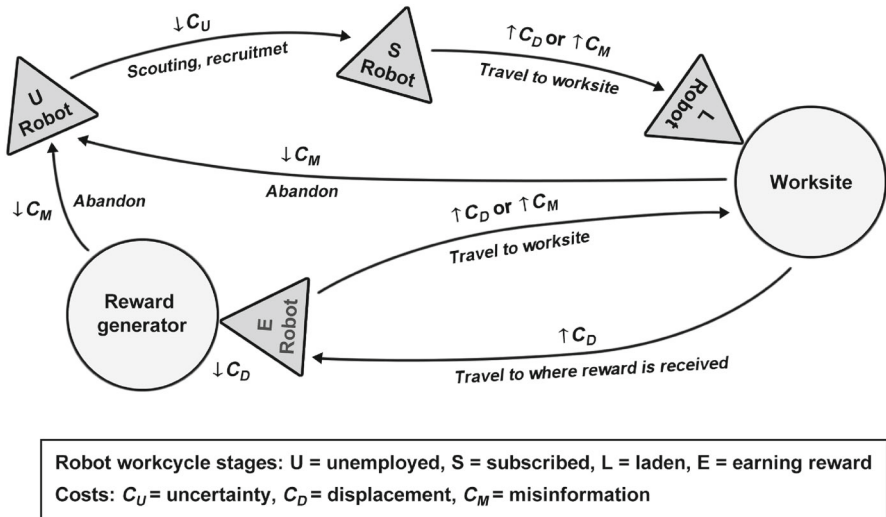


Fig. 7 The robot work cycle, along with the incurred costs. Arrows next to cost symbols that point upwards (downwards) indicate that a robot increases (decreases) the amount of that particular cost paid by the swarm in the previous time step by transitioning from one stage of the work cycle to another

6.1 The robot work cycle

The robot work cycle (Fig. 7) can be generalised to describe both Consumption and Collection as follows. A robot starts by being *unemployed* (*U*) and searches the environment for information. When it discovers a worksite, either by itself or as a result of being recruited, it *subscribes* (*S*) to that worksite. It then travels to it and becomes *laden* (*L*) with resource. It starts *earning* (*E*) reward when it reaches a *reward generator*. In the *Consumption* mission, the reward generator is the worksite itself and laden robots immediately become earning robots. In the *Collection* mission, the reward generator is the base and laden robots need to travel there in order to earn reward. Note that the total number of robots in a swarm, $N_R = U + S$ and that $S \geq L \geq E$.

During each stage of the work cycle, a robot has the potential to incur certain costs, also depicted in Fig. 7. There are three types of cost: *uncertainty cost*, C_U , incurred by unemployed robots that do not know where work is located, *displacement cost*, C_D , that all subscribed robots pay until they reach a reward generator, and *misinformation cost*, C_M , incurred by robots that are subscribed to depleted worksites and are thus unable to find or perform work.

Figure 8 shows an example of how these costs are incurred by robots that utilise recruitment over time. At the beginning of a run, all robots are unemployed, paying the maximum amount of uncertainty cost. C_U decreases when robots learn about a worksite, while C_D increases as some of those robots are recruits that are not yet located at the worksite. When one worksite gets depleted (just before the first hour in Fig. 8), the total uncertainty cost decreases, since there is one less active worksite that the swarm needs to know about. However, robots that are still subscribed to the depleted worksite incur misinformation cost until they determine that the worksite is in fact depleted and they abandon it. The task is completed when all worksites are depleted (at around 1.5h in Fig. 8), and costs fall to 0.

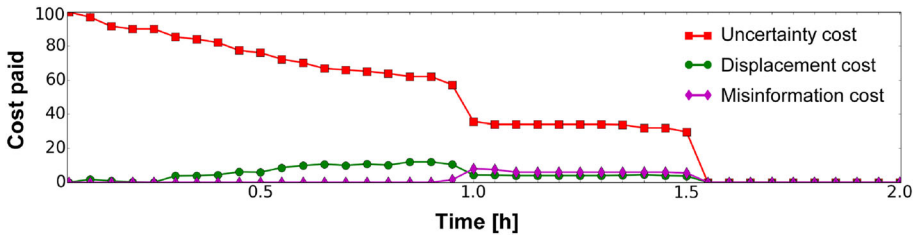


Fig. 8 An example of costs incurred by robots over time during a single experimental run with 25 local broadcasters in the static Consumption mission, Heap2 scenario, $D = 9$ m

Quantifying these costs first requires calculating the amount of reward, r , available per worksite and per robot:

$$r = \frac{R'_T}{N_R \times N'_W} \quad (6)$$

where N_R is the total number of robots and N'_W is the total number of worksites that can be active at the same time (i.e. in static environments, the number of worksites at the beginning of an experiment). R'_T represents the total amount of reward available from the N'_W worksites. During experiments with static environments, $R'_T = 100$ (see Sect. 3.1). Note that if reward r could be obtained by all robots from all worksites at the same time, the task would be instantly completed.

6.2 Uncertainty cost

When all robots are unemployed, no reward can be obtained and the total amount of uncertainty cost thus equals the total reward from all active worksites. When a robot finds out about a worksite, the swarm's C_U decreases by r . At any given time, the amount of C_U a swarm pays is thus:

$$C_U = \sum_{W=1}^{N_A} [(N_R - S_W) \times r] \quad (7)$$

where N_A is the number of active worksites and S_W is the number of robots subscribed to a worksite W . Note that the change in uncertainty cost, ΔC_U , relates to the swarm's information gain, ΔI , in the following way:

$$\begin{aligned} \Delta C_U &= C_U(t) - C_U(t-1) \\ &= \left[\sum_{W=1}^{N_A(t)} (N_R \times r) - \sum_{W=1}^{N_A(t)} (S_W(t) \times r) \right] \\ &\quad - \left[\sum_{W=1}^{N_A(t-1)} (N_R \times r) - \sum_{W=1}^{N_A(t-1)} (S_W(t-1) \times r) \right] \\ &= \left[\sum_{W=1}^{N_A(t)} (N_R \times r) - \sum_{W=1}^{N_A(t-1)} (N_R \times r) \right] - \Delta I \times r \end{aligned} \quad (8)$$

Or:

$$\Delta I \times r = -\Delta C_U + \left[\sum_{W=1}^{N_A(t)} (N_R \times r) - \sum_{W=1}^{N_A(t-1)} (N_R \times r) \right] \quad (9)$$

In other words, the swarm’s information gain between time steps $(t - 1)$ and t is directly proportional to the sum of the decrease in the swarm’s uncertainty cost and the change in the total available reward from all active worksites. If the number of active worksites at time step t remains the same as in time step $(t - 1)$, i.e. when no worksites are depleted or added to the environment, then $\sum_{W=1}^{N_A(t)} (N_R \times r) - \sum_{W=1}^{N_A(t-1)} (N_R \times r) = 0$ and $\Delta I \times r = -\Delta C_U$.

6.3 Displacement cost

The *displacement cost*, C_D , is defined as:

$$C_D = \sum_{W=1}^{N_A} [(S_W - E_W) \times r] + \sum_{W=1}^{N_D} [(L_W - E_W) \times r] \tag{10}$$

where E_W is the number of robots *earning* reward from a worksite W , N_D is the number of depleted worksites and L_W is the number of robots laden with resource from a worksite W . The first term on the right hand side of Eq. 10 represents the displacement cost incurred by robots subscribed to, but not located at, active worksites. These robots are either travelling to worksites after being recruited, or, in the case of the Collection mission, they are travelling between the base and worksites to unload resource (see Fig. 7). The second term represents cases when robots laden with resource from depleted worksites travel to the base during the Collection mission.⁶

The relationship between a reduction in uncertainty cost and an increase in displacement cost reflects the extent to which robots that learn about a worksite are able to obtain reward from it. We can characterise this relationship in terms of the *displacement cost coefficient*, d , as:

$$d = \frac{C_D}{\sum_{W=1}^{N_A} (S_W \times r)} \tag{11}$$

Note that the denominator in Eq. 11 is the term that gets subtracted from the uncertainty cost for the robots that know about worksites (see Eq. 7). When $d = 0$ and $\sum_{W=1}^{N_A} (S_W \times r) > 0$, a decrease in uncertainty cost is fully realised as reward, i.e. $C_D = 0$. When $d = 1$, all robots that know about worksites are displaced from a reward generator and no reward is obtained, i.e. $C_D = \sum_{W=1}^{N_A} (S_W \times r)$. Intermediate values of $0 < d < 1$ indicate that some robots are displaced and some are receiving reward.

The displacement cost coefficient is affected by the way in which a robot control strategy utilises information, as well as by the location at which information is shared. Solitary robots do not pay C_D during the Consumption mission, since they do not recruit and consequently scouts are already present at a worksite when they learn about it. In swarms that do communicate, C_D is incurred by recruited robots until they reach a worksite advertised to them. Additionally, in the bee swarms, scouts incur C_D when they travel to the base and back in order to recruit. Therefore, bee swarms have the highest displacement cost coefficient in most environments (see Fig. 9 for results from 25-robot swarms and online supplementary material, Figures S2.9 and S2.10 for results from 10- and 50-robot swarms, respectively). However, when the number of worksites is small, most notably the Heap1 scenarios and

⁶ The number of laden, rather than subscribed robots, is used in the second right hand side term of Eq. 10, as a robot that depletes a worksite during the Collection mission abandons it immediately (i.e. it is not subscribed to it anymore), but it still needs to travel to the base in order to obtain reward.

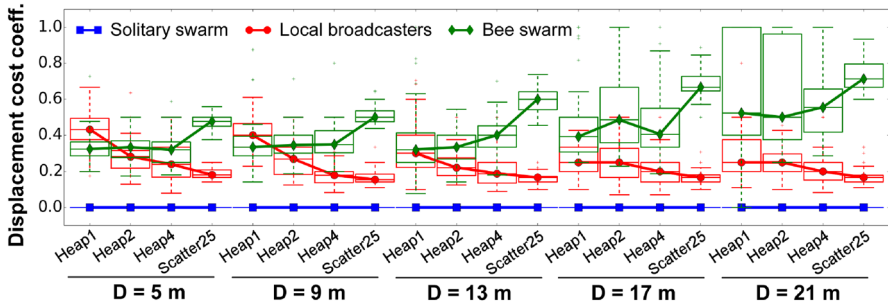


Fig. 9 Median displacement cost coefficient, d , of 25-robot swarms in the static Consumption mission. Bee swarms usually have the highest d , apart from environments where the number of worksites and worksite distance is small. The d of solitary swarms is always 0

when worksites are very close to the base, local broadcasters experience congestion and their d is higher than, or similar to, that of bee swarms.

6.4 Misinformation cost

On some occasions, an *unladen* robot becomes subscribed to a worksite that has been depleted. Such robots incur a *misinformation cost*, C_M :

$$C_M = \sum_{W=1}^{N_D} [(S_W - L_W) \times r] \tag{12}$$

Recall that robots that are *laden* with resource from a depleted worksite pay a displacement cost instead.

Solitary robots do not incur C_M during the Consumption mission because they do not recruit, meaning that when a worksite is depleted, all robots that are subscribed to it immediately become aware of that fact and abandon the worksite (see Fig. 10 for results from 25-robot swarms and online supplementary material, Figures S2.11 and S2.12 for results from 10- and 50-robot swarms, respectively). A similar trend can be observed for bee robots, especially in Heap scenarios, since bee robots recruit in the base and, in the case of Consumption missions, only when they initially discover a worksite, which means that recruits

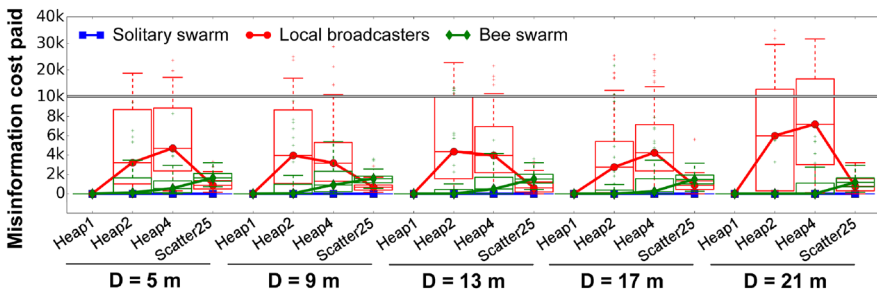


Fig. 10 Misinformation cost, C_M , of 25-robot swarms in the static Consumption mission. Local broadcasters pay a larger C_M than the other swarms in Heap environments, while in Scatter25 environments, C_M of bee swarms is the largest. Solitary swarms do not pay C_M

are usually already present at worksites when they become depleted. On the other hand, local broadcasters pay a relatively high amount of C_M in Heap environments, since they recruit continuously until worksites become depleted, which increases the probability of recruits travelling to worksites from which reward can no longer be gained.

Note that it only makes sense to consider misinformation cost while there are still active worksites in the environment. Therefore, C_M in static Heap1 environments is always equal to zero.

7 The Information-Cost-Reward framework

The Information-Cost-Reward framework identifies the relationship between scouting efficiency, information gain rate, the tendency of robots to incur costs and the reward that a swarm obtains at a given point in time. If robots could immediately turn information into reward, i.e. if they did not have to travel to worksites and did not suffer from contention for the same unit of reward, the swarm could earn an *expected reward* R' :

$$R' = \sum_{W=1}^{N_W} (S_W \times r) \tag{13}$$

where $N_W = N_A + N_D$ is the total number of worksites (active and depleted). The amount of *actual* reward ΔR , that a swarm is earning at a given point in time, is defined as:

$$\Delta R = \sum_{W=1}^{N_W} (\rho \times E_W) \tag{14}$$

where ρ is the reward intake rate (see Sect. 3.1). Equation 15 shows that a swarm cannot utilise information about worksites and obtain the full expected reward for free—it has to pay the displacement and misinformation costs associated with its control strategy, that are directly proportional to the difference between the expected and the actual reward.

$$\begin{aligned} C_D + C_M &= \left[\sum_{W=1}^{N_A} (S_W \times r) - \sum_{W=1}^{N_A} (E_W \times r) + \sum_{W=1}^{N_D} (L_W \times r) \right. \\ &\quad \left. - \sum_{W=1}^{N_D} (E_W \times r) \right] + \left[\sum_{W=1}^{N_D} (S_W \times r) - \sum_{W=1}^{N_D} (L_W \times r) \right] \tag{15} \\ &= \sum_{W=1}^{N_W} (S_W \times r) - \sum_{W=1}^{N_W} (E_W \times r) \\ &= R' - \frac{r}{\rho} \times \Delta R \end{aligned}$$

Additionally, by also considering all unemployed (i.e. non-subscribed) robots, we can show how the sum of all three costs, uncertainty, displacement and misinformation, relates to the swarm’s actual reward. First, we define the *potential reward*, R^* , as a sum of the expected reward and the reward that could be achieved by all unemployed robots from all active worksites, if the unemployed robots knew where the worksites were located:

$$\begin{aligned}
R^* &= \sum_{W=1}^{N_W} [S_W \times r] + \sum_{W=1}^{N_A} [(N_R - S_W) \times r] \\
&= \left[\sum_{W=1}^{N_A} (S_W \times r) + \sum_{W=1}^{N_D} (S_W \times r) \right] + \left[\sum_{W=1}^{N_A} (N_R \times r) - \sum_{W=1}^{N_A} (S_W \times r) \right] \quad (16) \\
&= \sum_{W=1}^{N_A} (N_R \times r) + \sum_{W=1}^{N_D} (S_W \times r)
\end{aligned}$$

The sum of all three costs is directly proportional to the difference between the potential reward and the actual reward:

$$\begin{aligned}
C_U + C_D + C_M &= \left[\sum_{W=1}^{N_A} (N_R \times r) - \sum_{W=1}^{N_A} (S_W \times r) \right] \\
&\quad + \left[\sum_{W=1}^{N_A} (S_W \times r) - \sum_{W=1}^{N_A} (E_W \times r) + \sum_{W=1}^{N_D} (L_W \times r) \right. \\
&\quad \left. - \sum_{W=1}^{N_D} (E_W \times r) \right] + \left[\sum_{W=1}^{N_D} (S_W \times r) - \sum_{W=1}^{N_D} (L_W \times r) \right] \quad (17) \\
&= \left[\sum_{W=1}^{N_A} (N_R \times r) + \sum_{W=1}^{N_D} (S_W \times r) \right] \\
&\quad - \left[\sum_{W=1}^{N_A} (E_W \times r) + \sum_{W=1}^{N_D} (E_W \times r) \right] \\
&= R^* - \frac{r}{\rho} \times \Delta R
\end{aligned}$$

Figure 11 graphically summarises the Information-Cost-Reward framework formalised above. A swarm is understood as a single entity that acts on its environment in order to obtain reward. Reward, situated in worksites, is dispersed in the environment in a certain way, and there is a certain probability, $p(W)$, associated with a worksite being located at a given point in space. Scouts play the role of a swarm's *sensors*. They find new information about where worksites are, decreasing the amount of C_U that the swarm pays. Since the swarm has new information about worksites, expected reward, R' , is generated. Upon acquiring a new piece of information, scouts can become workers, but they can also pass the information that they have to other members of the swarm in order to recruit more workers. The swarm's information gain, ΔI , describes when information is gained and lost, while the information gain rate, i , characterises how quickly robots can gain information about worksite locations through scouting or recruitment.

Workers act as *actuators* of the swarm. They can share information with each other, and they turn the information that they have into actual reward, R . However, there is a potential, unique to each combination of a robot control strategy, environment structure, and swarm mission that the workers incur displacement cost in order to utilise information and obtain reward. Furthermore, by acting on the environment, workers eventually cause worksites to become depleted. This can result in exploitative interference and in misinformation cost being incurred. At the same time, depletion of worksites decreases $p(W)$, causing scouts to become less successful over time.

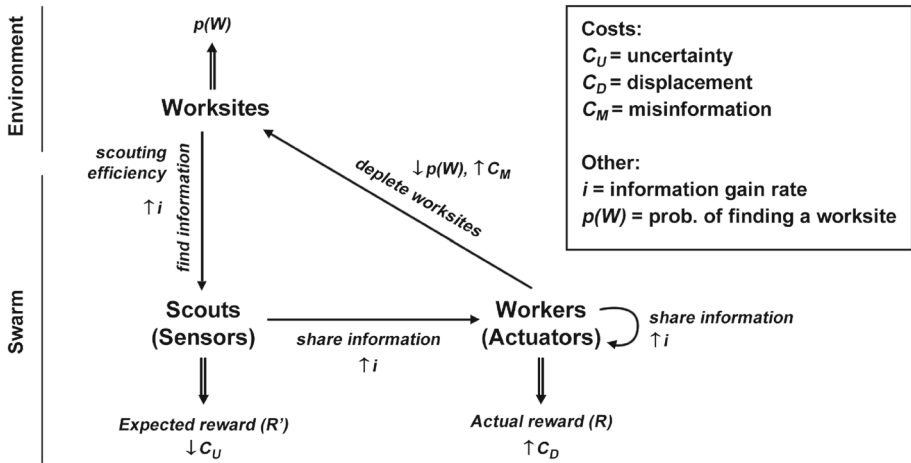


Fig. 11 Graphical representation of the Information-Cost-Reward framework. Arrows next to symbols that point upwards (downwards) indicate that a particular cost or probability increases (decreases) compared to the previous time step, as a result of an action that the swarm takes

Analyses supported by the ICR framework allow us not only to explain the observed performance of a swarm, but also to form hypotheses about how swarms will perform in new missions and environments. This is demonstrated in the following two sections.

8 Displacing the reward: Collection in static environments

Recall that during the Consumption mission, bee swarms are disadvantaged, as they pay an unnecessary amount of displacement cost as a result of limiting their recruitment activity to occur only in the base. However, in the Collection mission, robots need to periodically travel between worksites and the base in order to drop off resource. In the terminology of the ICR framework, a robot needs to periodically *displace* itself away from its worksite in order to obtain reward. It is therefore hypothesised that:

Hypothesis 8.1 Provided that their information gain rate is comparable to that achieved during Consumption tasks, bee swarms will outperform the other control strategies in the Collection mission, since the higher displacement costs that result from recruitment in the base will be compensated for by the fact that the Collection task requires all robots to travel to the base.

An ICR analysis (see online supplementary material, Section S4) reveals that the scouting efficiency and the information gain rate of swarms are indeed similar during Collection and Consumption. This is because the structure of the environments remains the same. Bee swarms usually have the highest information gain rate, followed by local broadcasters and solitary swarms, but their scouting efficiency is negatively affected by worksite distance to a higher degree than in the other swarms. On the other hand, displacement cost, C_D , is incurred by robots from all three swarms and during the majority of a robot’s work cycle.

Hypothesis 8.1 is supported in the Heap environments (Fig. 12), most notably in Heap1, where bee swarms gain a significantly higher reward from Collection than other swarms. In these environments, bee swarms can take advantage of their high i , while their displacement

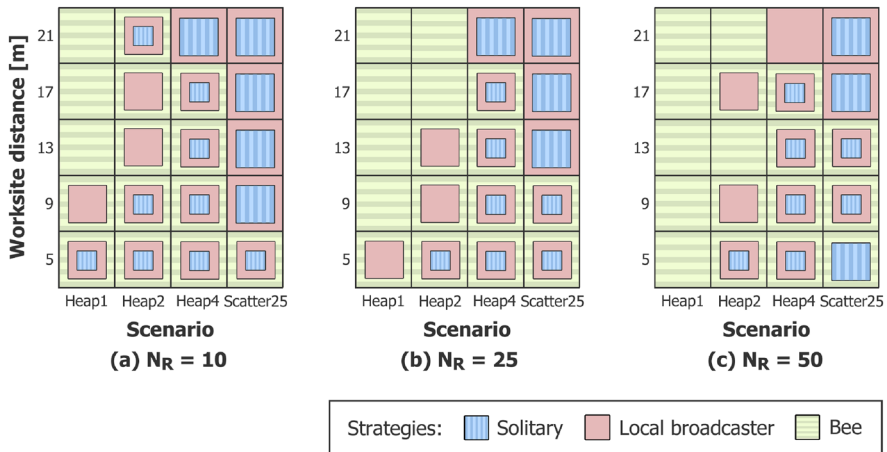


Fig. 12 Winning strategies that complete the static Collection mission the fastest in different environments using **a** 10, **b** 25 and **c** 50 robots. Box plots of these results are shown in online supplementary material, Figures S4.11–S4.13

cost becomes less relevant due to the nature of the mission. On the other hand, they cannot outperform the other swarms in easy environments, where the number of worksites is high ($N_W = 25$) or when worksites are close to the base ($D = 5$ m). This is caused by the fact that apart from information gain rate and displacement cost, misinformation cost affects the swarm performance as well. Bee swarm robots incur the highest C_M in most environments (see online supplementary material, Section S4.4), i.e. they exhibit the highest exploitative interference. This is very disadvantageous in environments like Scatter25, where the swarm needs to spread out in order to explore and exploit the environment efficiently.

9 Missing opportunities: Working in dynamic environments

In the dynamic environments explored here, worksite locations change periodically over time. Each experimental run is split into a number of change intervals. At the end of each interval, undepleted worksites are removed from the environment and new worksites are added at locations randomly generated according to the scenario type. There are two versions of each dynamic mission, a *slow* and a *fast* version. The environment changes 10 times in the slow and 20 times in the fast version, representing different degrees of challenge. Refer to the online supplementary material, Section S5 for further details on how dynamic environments were set up.

Unlike in static environments, swarms in dynamic environments need to re-discover worksites after the environment changes. Furthermore, reward from a given worksite is only available temporarily, which means that it needs to be extracted as soon as possible, making it more important for the swarm to spread itself evenly across worksites and to minimise its displacement and misinformation costs. It is thus hypothesised that:

Hypothesis 9.1 Inefficient scouting behaviour of bee swarms will cause them to have lower scouting efficiency relative to the other swarms in a greater number of dynamic, compared to static, environments.

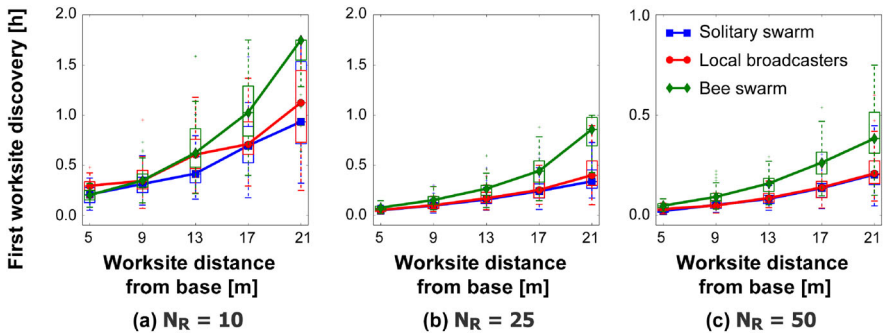


Fig. 13 Median time of the first worksite discovery in each change interval in the slow dynamic Consumption mission, Heap1 environments, using **a** 10, **b** 25 and **c** 50 robots. The negative effect of large worksite distance, D , is larger in bee swarms, compared to the other control strategies. Note the different scales of the y-axis for different values of N_R

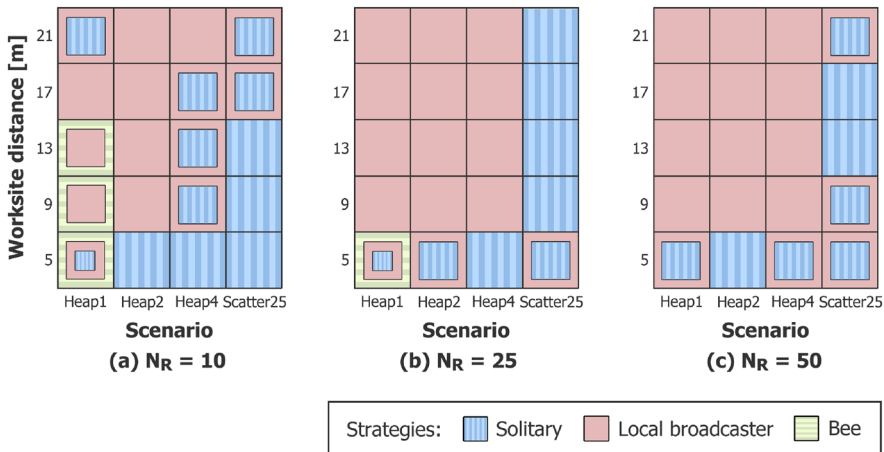


Fig. 14 Winning strategies in the slow dynamic Consumption mission using **a** 10, **b** 25 and **c** 50 robots. Box plots of these results are shown in online supplementary material, Figures S5.45a, S.5.46a and S.5.47a

Hypothesis 9.2 Bee swarms, that have an unnecessarily high displacement cost coefficient in the static Consumption mission, will perform significantly worse than other control strategies in the dynamic version of the mission.

Hypothesis 9.3 Solitary swarms, that have the lowest tendency to incur misinformation cost, will achieve a performance benefit over other strategies in more scenarios when an environment is dynamic. By extension, strategies that incur high misinformation cost will perform poorly in dynamic environments.

Figure 13 shows the scouting efficiency of swarms in the slow dynamic Consumption mission. Refer to online supplementary material, Section S5.1 for analysis of scouting efficiency in other missions. In line with Hypothesis 9.1, bee swarms are not able to discover new worksites after the environment changes as quickly as the other swarms. Their difficulty is apparent regardless of worksite distance form the base, but it is stronger when worksites are further away. This also negatively affects the ability of bee robots to gain and share infor-

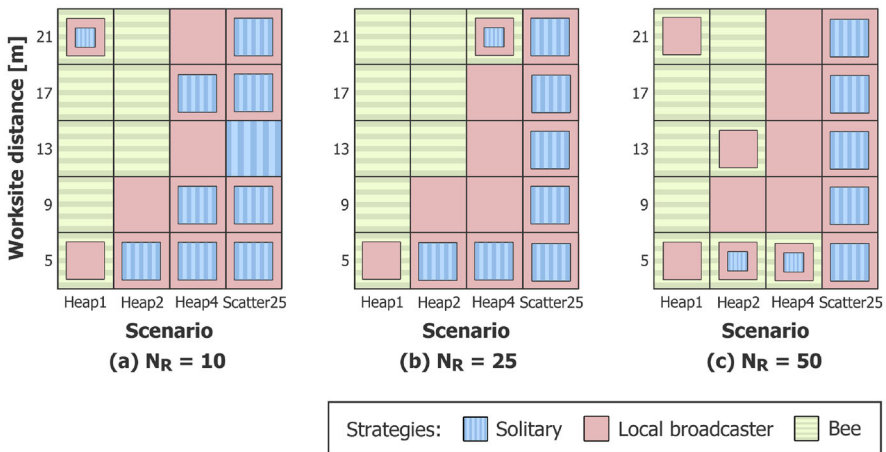


Fig. 15 Winning strategies in the slow dynamic Collection mission using **a** 10, **b** 25 and **c** 50 robots. Box plots of these results are shown in online supplementary material, Figures S5.48a, S.5.49a and S.5.50a

mation, preventing bee swarms from reaching the highest i in many dynamic environments (see online supplementary material, Section S5.2).

Figure 14 and Figure S5.43 in the online supplementary material depict winning strategies in the slow and fast dynamic Consumption missions and support the expectations of Hypothesis 9.2. Even though bee swarms perform similarly well compared to the other swarms in the Heap1 environments during the static Consumption mission (see Fig. 3), their performance is very rarely equivalent and never better than that of other swarms in the dynamic Consumption mission. While their poor performance is caused partially by their inefficient scouting, it is also low due to their inefficient use of information and extremely high displacement cost coefficient (see online supplementary material, Section S5.3). On the other hand, in line with Hypothesis 9.3, solitary swarms outperform the other swarms in a greater number of environments compared to the static Consumption mission, due to the lower costs that they incur (see online supplementary material, Sections S5.3 and S5.4).

According to Hypothesis 9.3, local broadcasters, which pay the highest amount of C_M in Heap environments (see online supplementary material, Section S5.4), should be outperformed by other swarms. This is, however, not true, and local broadcasters are a winning strategy in most Heap environments in the dynamic Consumption mission. This is because they are able to trade off high C_M for a relatively fast information gain rate and a relatively low displacement cost incurred (see online supplementary material, Sections S5.2 and S5.3).

Winning strategies in the dynamic Collection mission are depicted in Fig. 15 and in online supplementary material, Figure S5.44. As was the case in the static Collection mission, bee swarms are the winning strategy in the most difficult environments, such as Heap1 and Heap2 with a large D , especially when the environment changes slowly. However, in line with Hypothesis 9.3, bee swarms do not perform as well as in the static Collection mission (see Fig. 12) in scenarios with intermediate and low difficulty, such as the Heap2, Heap4 and Scatter25 scenarios with a small D . In these dynamic environments, bee swarm robots tend to overcommit to worksites, incurring a higher amount of C_M than the other swarms (see online supplementary material, Section S5.4). Their displacement cost is still “discounted” to some extent by the nature of the Collection task, but to a lesser degree than was the case in static

environments. The disadvantage of bee swarms is more pronounced when the environment changes quickly (see online supplementary material, Figure S5.44).

The advantage of solitary swarms over the other swarms is smaller in the dynamic Collection, compared to the dynamic Consumption mission, especially when the environment changes slowly. However, Hypothesis 9.3 is still supported, as solitary swarms never outperform the other swarms when performing Collection in static environments (see Fig. 12). The difficulty encountered by solitary swarms during dynamic Collection missions stems from the fact that the robots need to travel between the base and worksites and are therefore not able to sample the environment as quickly. This difficulty is also reflected by their lower information gain rate (see online supplementary material, Section S5.2).

10 Discussion

10.1 Summary

We have performed simulated experiments with three types of homogeneous robot swarms (solitary, local broadcaster and bee) in the context of collective foraging and have identified relevant metrics, under the umbrella of the Information-Cost-Reward framework, that affect swarm performance. Scouting efficiency is related to the ability of swarms to discover new worksites. Information gain rate, i , measures how well robots are able to obtain new information and share it amongst themselves, decreasing the swarm's uncertainty cost, C_U . Displacement cost, C_D , and misinformation cost, C_M , characterise how efficiently a swarm can turn information about worksites into reward. Displacement cost is incurred by robots that are informed about worksites but are spatially remote from them and are therefore unable to immediately obtain reward. Misinformation cost is incurred by robots with outdated information that are attempting to reach a worksite that has already been depleted by others. Recruitment often leads to a tendency of robots to incur C_D and C_M , with the extent of these costs depending on the structure of the environment and on scouting and recruitment strategies of the robots. The sum of C_D and C_M accounts for the difference between the swarm's expected reward, which the swarm should be receiving based on the number of robots that know about worksites, and the actual reward that the swarm receives at a given point in time. The sum of all three costs, C_U , C_D and C_M , is equal to the difference between the potential reward available in the environment and the swarm's actual reward.

The relationship between the characteristics of a robot control strategy and properties of the environment is complex, and there is often a trade-off between being able to achieve a high information gain rate, usually through recruitment, and having to pay costs associated with discovering, sharing and using information about worksites. This trade-off is usually described in terms of the "exploration versus exploitation" paradigm. As we have shown here, exploration can further be divided into its scouting and recruitment aspects, and its effectiveness measured by scouting efficiency and information gain rate. Exploitation has various costs associated with it that manifest themselves to varying degrees depending on the swarm's control strategy and the nature of the swarm's mission and of its environment.

Solitary swarms, that do not use recruitment and thus have a small i but also pay a small amount of C_D and C_M , outperform the other swarms in the least difficult environments where worksites are easy to find. This is especially true in dynamic environments, where worksite locations change over time, and where worksites thus need to be discovered and exploited quickly. Local broadcasters, that recruit each other near worksites, usually have a

higher i and pay a higher amount of C_D and C_M than solitary swarms and are thus suitable in environments where worksites are more difficult to discover. Bee swarms, where robots recruit each other in the base, usually achieve the highest i , but they also incur high displacement and misinformation costs. This is often disadvantageous in the Consumption mission and in dynamic environments, where robots are required to spend as much time as possible outside of the base exploring and exploiting the environment. On the other hand, the costs of bee swarms are, to a certain extent, “discounted” during Collection, since the mission itself required the robots to travel to the base, allowing bee swarms to outperform other control strategies in many environments. Similar trends in relative performances of various robot control algorithms with and without recruitment have been identified by other authors (e.g. Krieger and Billeter 2000; Gutiérrez et al. 2010; Wawerla and Vaughan 2010; Sarker and Dahl 2011; Lee et al. 2013; Hecker and Moses 2015).

10.2 The ICR framework and future work

The ICR framework is conceptually located between swarm analysis approaches, which are based on information theory and measure entropy in observed collective behaviour, and mathematical modelling approaches, which describe the behaviour of swarms as a whole by using differential equations.

It has previously been suggested that swarms should be understood as information-processing cognitive systems (Trianni et al. 2011; Reina et al. 2015b). In contrast to traditional information-based approaches, the ICR framework characterises information flow not in terms of transfer entropy, but in terms of the information gain rate, i . This approach has the following three advantages. Firstly, transfer entropy is to some extent a proxy measure of information flow, because it measures the coupling between “states” of two agents at two different time steps, rather than measuring whether information was exchanged between them. On the other hand, information gain, ΔI , the rate of which is used in this paper to characterise information flow, is calculated as a change in the number of informed robots and it thus directly captures information exchange. Secondly, since ΔI is based on the number of informed robots, it does not require us to define what a robot “state” is in relation to having information about a worksite. Informed robots may be performing a number of different operations, for example, gathering resources, travelling to the base, etc., as is typical for foraging and other swarm tasks. Thirdly, unlike entropy, information gain can be directly related to the amount of uncertainty cost that the swarm pays (see Sect. 6.2, Eqs. 8 and 9) and thus to the ability of the swarm to turn information into reward (see Sect. 7, Eq. 17).

Unlike mathematical modelling approaches, the ICR framework is not suitable for modelling the swarm per se. Because of this, it is not possible to use it for precisely predicting or optimising swarm performance, or to study macroscopic robot population dynamics, as can be done by using probabilistic finite state machines and differential equations (e.g. Liu and Winfield 2010; Montes de Oca et al. 2011; Mather and Hsieh 2012; Reina et al. 2015b; Scheidler et al. 2016; Valentini et al. 2016). Similarly, it is not possible to use the framework to uncover behaviour rules of agents, as can be done by using Turing Learning (Li et al. 2016). The ICR framework is instead useful in the following two ways. Firstly, it allows us to precisely identify what part of the robot work cycle leads to the observed swarm performance by characterising information flow and various costs that robots incur. Secondly, it is useful when forming hypotheses about the performance of swarms in new missions, i.e. it can guide algorithm selection, as it was demonstrated in Sects. 8 and 9. As a further example, consider the following scenario.

Consider a bee-inspired swarm that initially is optimised to discover and collect minerals. When the same swarm is deployed to search for and rescue hikers, performance strongly deteriorates. A parameter optimisation technique is used to determine a new set of parameter values that satisfy the performance requirements of the new search-and-rescue mission. However, it is likely that the newly optimised behaviour will deliver a poor performance again in a different context, for example when hiker density or hiker movement patterns change over the course of the year. Using the ICR framework to analyse the swarm performance could reveal that bee-inspired swarms perform poorly in dynamic environments due to their poor scouting efficiency and the high displacement and misinformation costs that they pay. We could, for example, hypothesise that the scouting efficiency of the swarm will improve when a number of bee scouts will always be active in the environment. Or, we could form and test a hypothesis that a local broadcaster algorithm, where robots do not periodically travel to a central location, will incur a lower costs and thus deliver a better performance. In other words, we would be able to apply high-level knowledge about various robot control algorithms to form hypotheses about their performance.

Because it can facilitate understanding of underlying mechanisms that lead to macro-level behaviour, the ICR framework is the first step towards a swarm design methodology where design principles can be separated from algorithm implementation and shared between research groups and with the robotics industry. Recent experiments have shown that both manual and automated swarm design methodologies significantly benefit from being constrained to a set of possible robot behaviours, as opposed to being open-ended (Francesca et al. 2014). Using the ICR framework, we could identify such behaviour sets by studying how different parts of robot control algorithms affect the overall swarm performance. To this end, we are currently working on a catalogue of “design patterns” for robot swarm foraging. We are also performing more simulated and real-world experiments with different robot control strategies in order to verify the ICR framework and validate the simulation studies presented here.

In the future, the ICR framework has to be verified, and possibly refined, based on data from a wider variety of swarm foraging tasks. These tasks could involve, for instance, costly transitions between worksites, a changing number of robots or worksites, heterogeneous robot swarms, or worksites that multiple robots have to cooperate on. It is probable that the ICR framework will need to be extended to account for types of costs other than those discovered in simulation studies presented here. For example, a “cooperation” cost might be incurred when robots need to wait for each other in order to work together. Nevertheless, here we have demonstrated a methodology for identifying these costs and for relating them to the swarm information flow and performance. Finally, it is important to further understand the effect of robot parameters, such as sensor range, communication range, wheel speed, sensory-motor noise on information flow and costs.

11 Conclusion

Algorithms for swarm foraging are currently difficult to understand and design due to the nonlinear nature of the emergent collective behaviour. The Information-Cost-Reward framework developed in this paper demonstrates how information flow in swarms can be formally related to the amount of reward that a swarm receives from the environment during foraging. Because of its emphasis on a more high-level, information-centred understanding, the framework allows us to not only explain existing behaviour, but also to hypothesise about the relative performance of robot control strategies in new missions and environments. This work is a step towards a design methodology where swarm control algorithms are created by

selecting the appropriate robot behaviour, such as a certain scouting or recruitment strategy, based on its effect on swarm performance. Unlike the existing swarm modelling methods, that are often used for robot algorithm optimisation, the ICR framework is useful during algorithm selection, complementing existing approaches.

The ICR framework also has implications for the field of swarm cognition (Trianni et al. 2011). By understanding a swarm as a single entity that gathers and processes information, the framework can help us to gain insights into how collective intelligence emerges, since it allows us to ask fundamental questions, such as what the importance of embodiment in collective intelligence is, or what properties of information flow inside a distributed system lead to a desired collective action.

Acknowledgements This work was supported by EPSRC Doctoral Training Centre grants EP/G03690X/1 and EP/N509747/1. All source code, data and ANOVA tables supporting this study are openly available from the University of Southampton repository at <http://eprints.soton.ac.uk/id/eprint/403544>.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Amazon Prime Air. (2016). <http://amzn.to/2af8Zcd>. Accessed on August 17, 2017.
- Balch, T. (2000). Hierarchic social entropy: An information theoretic measure of robot group diversity. *Autonomous Robots*, 8(3), 209–237.
- Balch, T., & Arkin, R. C. (1994). Communication in reactive multiagent robotic systems. *Autonomous Robots*, 1(1), 27–52.
- Bonani, M., Longchamp, V., Magnenat, S., Philippe, R., Burnier, D., et al. (2010). The MarXbot, a miniature mobile robot opening new perspectives for the collective-robotic research. In *Proceedings of the 2010 IEEE/RSJ international conference on intelligent robots and systems (IROS 2010)* (pp. 4187–4193). Piscataway, NJ: IEEE Press.
- Brambilla, M., Ferrante, E., Birattari, M., & Dorigo, M. (2013). Swarm robotics: A review from the swarm engineering perspective. *Swarm Intelligence*, 7(1), 1–41.
- Brambilla, M., Brutschy, A., Dorigo, M., & Birattari, M. (2014). Property-driven design for robot swarms: A design method based on prescriptive modeling and model checking. *ACM Transactions on Autonomous and Adaptive Systems*, 9(4), Article no. 17.
- Cartade, P., Lenain, R., Thuilot, B., Benet, B., & Berducat, M. (2012). Motion control of a heterogeneous fleet of mobile robots: Formation control for achieving agriculture task. In *Proceedings of the international conference on agricultural engineering (CIGR-AgEng '12)*.
- Charnov, E. L. (1976). Optimal foraging, the marginal value theorem. *Theoretical Population Biology*, 9(2), 129–136.
- Dahl, R. S., Mataric, M. J., & Sukhatme, G. S. (2009). Multi-robot task allocation through vacancy chain scheduling. *Robotics and Autonomous Systems*, 57, 674–687.
- Ducatelle, F., Di Caro, G. A., Pinciroli, C., & Gambardella, L. M. (2011). Self-organized cooperation between robotic swarms. *Swarm Intelligence*, 5(2), 73–96.
- Ducatelle, F., Di Caro, G. A., Forster, A., Bonani, M., Dorigo, M., Magnenat, S., et al. (2014). Cooperative navigation in robotic swarms. *Swarm Intelligence*, 8(1), 1–33.
- Fagen, R. (1987). A generalized habitat matching rule. *Evolutionary Ecology*, 1, 5–10. <https://doi.org/10.1007/BF02067264>.
- Fernandez-Marquez, J. L., Di Marzo, Serugendo G., Montagna, S., Viroli, M., & Arcos, J. L. (2013). Description and composition of bio-inspired design patterns: A complete overview. *Natural Computing*, 12(1), 43–67.
- Francesca, G., Brambilla, M., Brutschy, A., Garattoni, L., Miletitch, R., Podevijn, G., Reina, A., Soleymani, T., Salvaro, M., Pinciroli, C., Trianni, V., & Birattari, M. (2014). An experiment in automatic design of robot swarms. In M. Dorigo, M. Birattari, S. Garnier, H. Hamann, M. Montes de Oca, C. Solnon, & T.

- Stützle (Eds.). *Proceedings of the international conference on swarm intelligence (ANTS 2014). Lecture Notes in Computer Science* (Vol. 8667, pp 25–37). Berlin: Springer.
- Gardelli, L., Viroli, M., & Omicini, A. (2007). Design patterns for self-organising systems. In H. D. Burkhard, G. Lindemann, Verbrugge, et al. (Eds.), *Proceedings of the 5th international central and eastern European conference on multi-agent systems (CEEMAS 2007)* (pp. 123–132). Berlin: Springer.
- Gerkey, B., & Mataric, M. J. (2003). Multi-robot task allocation: Analyzing the complexity and optimality of key architectures. In *Proceedings of the 2003 IEEE international conference on robotics and automation (ICRA 2003)* (Vol. 3, pp. 3862–3868). Piscataway, NJ: IEEE Press.
- Griswold, A. (2016). Uber's self-driving cars are on the road. Quartz. <http://bit.ly/2ogEPxN>. Accessed on August 17, 2017.
- Gutiérrez, Á., Campo, A., Monasterio-Huelin, F., Magdalena, L., & Dorigo, M. (2010). Collective decision-making based on social odometry. *Neural Computing and Applications*, 19(6), 807–823.
- Hamann, H. (2013). Towards swarm calculus: Urn models of collective decisions and universal properties of swarm performance. *Swarm Intelligence*, 7(2), 145–172.
- Hecker, J. P., & Moses, M. E. (2015). Beyond pheromones: Evolving error-tolerant, flexible, and scalable ant-inspired robot swarms. *Swarm Intelligence*, 9, 43–70.
- Jevtic, A., Gutiérrez, Á., Andina, D., & Jamshidi, M. (2012). Distributed bees algorithm for task allocation in swarm of robots. *IEEE Systems Journal*, 6(2), 296–304.
- Krieger, M. J. B., & Billeter, J. B. (2000). The call of duty: Self-organised task allocation in a population of up to twelve mobile robots. *Robotics and Autonomous Systems*, 30(1–2), 65–84.
- Labella, T. H., Dorigo, M., & Deneubourg, J. L. (2006). Division of labour in a group of robots inspired by ants' foraging behaviour. *ACM Transactions on Autonomous and Adaptive Systems*, 1(1), 4–25.
- Lee, J. H., Ahn, C. W., & An, J. (2013). A honey bee swarm-inspired cooperation algorithm for foraging swarm robots: An empirical analysis. In *Proceedings of the 2013 IEEE/ASME international conference on advanced intelligent mechatronics (AIM 2013)* (pp. 489–493). Piscataway, NJ: IEEE Press.
- Lemmens, N., de Jong, S., Tuyls, K., & Nowe, A. (2008). Bee behaviour in multi-agent systems. In K. Tuyls, A. Nowe, Z. Goussoum, et al. (Eds.), *Adaptive agents and multi-agent systems III. Adaptation and multi-agent learning*. Lecture notes in computer science (Vol. 4865, pp. 145–156). Berlin: Springer.
- Lerman, K., Jones, C., Galstyan, A., & Mataric, M. J. (2006). Analysis of dynamic task allocation in multi-robot systems. *The International Journal of Robotic Research*, 25, 225–242.
- Li, W., Gauci, M., & Groß, R. (2016). Turing learning: A metric-free approach to inferring behavior and its application to swarms. *Swarm Intelligence*, 10(3), 211–243.
- Liu, W., & Winfield, A. F. T. (2010). Modelling and optimisation of adaptive foraging in swarm robotic systems. *The International Journal of Robotics Research*, 29(14), 1743–1760.
- Mataric, M. J., Sukhatme, G. S., & Ostergaard, E. H. (2003). Multi-robot task allocation in uncertain environments. *Autonomous Robots*, 14(2–3), 255–263.
- Mather, T. W., & Hsieh, M. A. (2012). Ensemble modeling and control for congestion management in automated warehouses. In *Proceedings of the 2012 IEEE international conference on automation science and engineering (CASE 2012)* (pp. 390–395). Piscataway, NJ: IEEE Press.
- Matplotlib: Box plots. (2017). https://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.boxplot. Accessed on August 17, 2017.
- Miller, J. M., Wang, X. R., Lizier, J. T., Prokopenko, M., & Rossi, L. F. (2014). Measuring information dynamics in swarms. In M. Prokopenko (Ed.), *Guided self-organisation: Inception, Emergence, Complexity and Computation* (Vol. 9, pp. 343–364). Berlin: Springer.
- Montes de Oca, M. A., Ferrante, E., Scheidler, A., Pinciroli, C., Birattari, M., & 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. <https://doi.org/10.1007/s11721-011-0062-z>.
- Pinciroli, C., Trianni, V., O'Grady, R., Pini, G., Brutschy, A., et al. (2012). ARGoS: A modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, 6(4), 271–295.
- Pini, G., Brutschy, A., Pinciroli, C., Dorigo, M., & Birattari, M. (2013). Autonomous task partitioning in robot foraging: An approach based on cost estimation. *Adaptive Behavior*, 21(2), 118–136.
- Pitonakova, L., Crowder, R., & Bullock, S. (2014). Understanding the role of recruitment in collective robot foraging. In H. Lipson, H. Sayama, J. Rieffel, et al. (Eds.), *Proceedings of the fourteenth international conference on the synthesis and simulation of living systems (ALIFE 14)* (pp. 264–271). IJde, MA: MIT Press.
- Pitonakova, L., Crowder, R., & Bullock, S. (2016a). Information flow principles for plasticity in foraging robot swarms. *Swarm Intelligence*, 10(1), 33–63.
- Pitonakova, L., Crowder, R., & Bullock, S. (2016b). Task allocation in foraging robot swarms: The role of information sharing. In G. Gershenson, T. Froese, J. M. Siqueiros, et al. (Eds.), *Proceedings of the fifteenth*

- international conference on the synthesis and simulation of living systems (ALIFE XV)* (pp. 306–313). Cambridge, MA: MIT Press.
- Reina, A., Miletitch, R., Dorigo, M., & Trianni, V. (2015a). A quantitative micro-macro link for collective decisions: The shortest path discovery/selection example. *Swarm Intelligence*, 9(2), 75–102.
- Reina, A., Valentini, G., Fernández-Oto, C., Dorigo, M., & Trianni, V. (2015b). A design pattern for decentralised decision making. *PLoS ONE*, 10(10), e0140950.
- Sarker, M. O. F., & Dahl, T. S. (2011). Bio-inspired communication for self-regulated multi-robot systems. In T. Yasuda (Ed.), *Multi-robot systems, trends and development*. (pp. 367–392). InTech. <https://doi.org/10.5772/13104>.
- Scheidler, A., Brutschy, A., Ferrante, E., & Dorigo, M. (2016). The k-unanimity rule for self-organized decision making in swarms of robots. *IEEE Transactions on Cybernetics*, 99, 1175–1188.
- Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27, 379–423.
- Sperati, V., Trianni, V., & Nolfi, S. (2011). Self-organised path formation in a swarm of robots. *Swarm Intelligence*, 5(2), 97–119.
- Stiefelwagen, M., Van Der Werff, K., Meijer, B. R., & Tomiyama, T. (2004). Distributed autonomous agents, navigation and cooperation with minimum intelligence in a dynamic warehouse application. In *Proceedings of the 2004 IEEE international conference on systems, man and cybernetics* (Vol. 6, pp. 5573–5578). Piscataway, NJ: IEEE Press.
- Trianni, V., Tuci, E., Passino, K. M., & Marshall, J. A. R. (2011). Swarm cognition: An interdisciplinary approach to the study of self-organising biological collectives. *Swarm Intelligence*, 5(1), 3–18.
- Tukey, J. (1949). Comparing individual means in the analysis of variance. *Biometrics*, 5(2), 99–114.
- Valentini, G., Ferrante, E., Hamann, H., & Dorigo, M. (2016). Collective decision with 100 Kilobots: Speed versus accuracy in binary discrimination problems. *Autonomous Agents and Multi-Agent Systems*, 30(3), 553–580.
- Wang, X. R., Miller, J. M., Lizier, J. T., Prokopenko, M., & Rossi, L. F. (2012). Quantifying and tracing information cascades in swarms. *PLoS ONE*, 7(7), e40,084.
- Wawerla, J., & Vaughan, R. T. (2010). A fast and frugal method for team-task allocation in a multi-robot transportation system. In *Proceedings of the 2010 IEEE international conference on robotics and automation (ICRA 2010)* (pp. 1432–1437). Piscataway, NJ: IEEE Press.
- Yang, Y., Zhou, C., & Tian, Y. (2009). Swarm robots task allocation based on response threshold model. In *Proceedings of the 4th international conference on autonomous robots and agents (ICARA 2009)* (pp. 171–176). Piscataway, NJ: IEEE Press.