EMERGE

WP5 Emergent awareness - Collective understanding and control of multi-agent systems

D5.2 Emergent Awareness Control

Version: 1.0

Date: 31/03/2025

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

Document control

Project title	Emergent awareness from minimal collectives
Project acronym	EMERGE
Call identifier	HORIZON-EIC-2021-PATHFINDERCHALLENGES-01-01
Grant agreement	101070918
Starting date	01/10/2022
Duration	48 months
Project URL	http://eic-emerge.eu
Work package	WP5 Emergent awareness - Collective understanding and control of multi-agent systems
Deliverable	D5.2 Emergent Awareness Control
Contractual delivery date	M30
Actual delivery date	M30
Nature[1]	R
Dissemination level[2]	PU
Lead beneficiary	UOB
Editor(s)	Simon Jones, David Garzon Ramos, Sabine Hauert
Contributor(s)	—
Reviewer(s)	Cosimo Della Santina
Document description	Report describing the conditions leading to different resolutions of awareness with examples in simulation and preliminary results on how to control emergence of awareness.

[1] R: Document, report (excluding the periodic and final reports); DEM: Demonstrator, pilot, prototype, plan designs; DEC: Websites, patents filing, press & media actions, videos, etc.; DATA: Data sets, microdata, etc.; DMP: Data management plan; ETHICS: Deliverables related to ethics issues.; SECURITY: Deliverables related to security issues; OTHER: Software, technical diagram, algorithms, models, etc.

[2] PU – Public, fully open, e.g. web (Deliverables flagged as public will be automatically published in CORDIS project's page); SEN – Sensitive, limited under the conditions of the Grant Agreement; Classified R-UE/EU-R – EU RESTRICTED under the Commission Decision No2015/444; Classified C-UE/EU-C – EU CONFIDENTIAL under the Commission Decision No2015/444; Classified S-UE/EU-S – EU SECRET under the Commission Decision No2015/444

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

Version control

Version[3]	Editor(s), Contributor(s), Reviewer(s)	Date	Description
0.1	Simon Jones, David Garzon Ramos	14/02/2025	Proposed ToC
0.2	Simon Jones, David Garzon Ramos	14/03/2025	Intermediate document for review
0.3	Cosimo Della Santina	24/03/2025	Review of full document
0.4	Sabine Hauert	27/03/2025	Final verifications
1.0	Davide Bacciu	31/03/2025	Document ready for upload

[3] 0.1 – TOC proposed by editor; 0.2 – TOC approved by reviewer; 0.4 – Intermediate document proposed by editor; 0.5 – Intermediate document approved by reviewer; 0.8 – Document finished by editor; 0.85 – Document reviewed by reviewer; 0.9 – Document revised by editor; 0.98 – Document approved by reviewer; 1.0 – Document released by Project Coordinator.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

Abstract

Awareness in biological agents has converging definitions when considering local states describing content-related consciousness from an agent-specific perspective. However, it becomes highly debated when it comes to global states. The issue magnifies when considering collectives of artificial agents. Several frameworks exist, all unsatisfactory in the limitations posed to agents' heterogeneity and disappearance of the local self into an integrated state.

Ultimately, existing frameworks are ineffective in explaining, facilitating, and supporting cooperative behaviours in artificial agents. The lack of a compelling theory of global awareness in AI is currently a significant barrier to the effective deployment of artificial agents in the real world.

EMERGE tackles this grand challenge by introducing the novel concept of collaborative awareness for collectives of minimal artificial beings. We will investigate how simple agents can develop a representation of their mutual existence, environment, and cooperative behaviour towards the realisation of tasks and goals.

EMERGE builds on a scenario of artificial beings with no shared language and constrained individual capabilities, which nevertheless leads to high-complexity behaviours at the collective level. Collaborative awareness becomes an emergent process supporting complex, distributed, and loosely coupled systems capable of high degrees of collaboration, self-regulation, and interoperability without predefined protocols.

EMERGE delivers a philosophical, mathematical, and technological framework that enables us to know how and where to allocate awareness to optimally achieve a goal through the collective. We will demonstrate EMERGE concepts on robotic use cases, with hints of the broader applicability of the framework to the Internet of Things, pervasive computing, and nanotechnologies. We will also investigate the ethical implications of collaborative awareness, focusing on moral responsibility, vulnerabilities, and trust.

Consortium

The EMERGE consortium members are listed below.

Organization	Short name	Country
Università di Pisa	UNIPI	IT
TU Delft	TUD	NL
University of Bristol	UOB	UK
Ludwig Maximilian University of Munich	LMU	DE
Da Vinci Labs	DVL	FR

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

Disclaimer

Acknowledgement

This document is a deliverable of EMERGE project. This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement Nº 101070918.

Table of Contents

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

List of Figures

List of Tables

List of Abbreviations

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

GA	Grant Agreement
CA	Consortium Agreement
IPR	Intellectual Property Rights
WP	Work Package
DSA	Distributed Spatial Awareness
DOTS	Distributed Organisation and Transport System
GBP	Gaussian Belief Propagation
DSA-RW	Distributed Spatial Awareness - Random Walk
DSA-KE	Distributed Spatial Awareness - Knowledge Enhancement
DSA-SF	Distributed Spatial Awareness - Shape Formation
FLOPs	Floating Point Operations per second
FoV	Field-of-View
IR	Infra Red
UV	Ultra Violet
ToF	Time-of-Flight
LLM	Large Language Model
BT	Behaviour Tree

Executive Summary

This document is a deliverable of the EMERGE project, funded under grant agreement number 101070918. This deliverable, **D5.2 Emergent Awareness Control**, is a report describing the conditions leading to different resolutions of awareness with examples in simulation and results on how to control emergence of awareness.

We detail our new method of giving agent collectives a Distributed Spatial Awareness (DSA), based purely on local sensing and communication. We then describe a methodology for combining different dimensions and metrics of awareness with task performance measures. This is illustrated with complex swarm logistics task, showing potential trade-offs needed to build artificial aware systems. Finally, we examine ways to automatically design for the emergence of the most task-effective forms of awareness.

1 Introduction

In Section 2, we first discuss the development and application of a new form of awareness for artificial collectives - Distributed Spatial Awareness, or DSA. By giving agents limited range

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

communication, noisy measurement, and random movement, and by using Gaussian Belief Propagation as a distributed, incremental, and robust factor graph optimisation paradigm, we enable a collective of agents to converge on a shared, swarm-centric, reference frame. This reference frame sits entirely within the tenets of swarm/emergent engineering, having no reliance or dependence on central resources or global knowledge. We call this ability Distributed Spatial Awareness, which when applied to swarm robotics, enables swarm algorithms capable of tasks at higher levels of performance than previously possible, such as pattern formation, or logistics applications. DSA is computationally cheap enough to be supported on low cost hardware, and by tuning parameters related to the frequency of communication and update rate of the underlying distributed factor graphs, we can trade-off degrees of awareness, convergence time, and computational cost.

We then move on to Section 3, where we discuss the ways in which we can use new forms of awareness, together with ways of measuring awareness, introduced in deliverable **D5.1**, to design more effective artificial systems, using swarm robotics as a use-case. We provide clear definitions of terminology, clarifying the lens through which to discuss these artificial aware systems, then describe an implementation pipeline by which to craft and characterise actual systems. We use the example of a swarm robotics collective transport scenario, and target the relationship between task performance and two dimensions of awareness; spatial awareness (DSA) and self-awareness (fault detection). We follow the steps of our implementation pipeline to build a simulation and characterisation platform and use this to show the sometimes surprising relationships, more awareness is not always associated with better performance.

In Section 4 we examine ways that we can actively and automatically design for the emergence of desired forms of awareness, specifically forms of communication and information sharing to enable coordination and cooperation. We construct a model where two different forms are available; direct communication, and indirect communication via the environment - stigmergy. In a set of experiments, we show that our automatic design method can both design effective controllers for different tasks, and select between the most appropriate communication modality. Swarms of simulated augmented e-puck robots can communicate either with coloured LED signalling, or by leaving pheromone traces in the environment, with no predefined communication protocols. The swarm is given a series of tasks requiring degrees of cooperation in order to succeed. Through the automatic design process, action and communication strategies co-emerge.

Finally, in Section 5 we briefly sketch out possible paths for future work, looking at ways to use automatic methods, such as evolutionary algorithms, to make use of the already existing dimensions of awareness in new and innovative ways. We also look at the possibility of algorithmically discovering new dimensions of awareness for heterogeneous systems, including robots, devices, and humans.

2 Distributed Spatial Awareness

Building a distributed spatial awareness within a collective of agents enables higher levels of coordination. When applied to swarm robotics, this enables locally sensing and communicating agents to perform new swarm algorithms. We use local observations by robots of each other and Gaussian Belief Propagation message passing combined with continuous swarm movement to build a global and distributed swarm-centric frame of reference. With low bandwidth and computation requirements, this shared reference frame allows new swarm algorithms. We characterise the system in simulation and demonstrate two example algorithms.

Swarm robotics, inspired by swarms in nature, has the potential for resilient, robust, and re-

European Innovation Council Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

dundant solutions to a wide range of problems such as mapping, logistics, search and rescue, disaster recovery, and environmental monitoring. Many relatively simple and cheap robots, each following simple rules, with local interactions between themselves and the environment are capable of producing a desired emergent swarm-level behaviour [55, 31, 12, 2, 60, 32].

There are many areas of swarm algorithm design where access to global information would be useful, but unless that information is inferred or constructed through purely local interactions, it does not fit within the distributed swarm paradigm. One example is spatial awareness, by which we mean that an agent within the swarm is aware of its own location with respect to the swarm as a whole; the swarm shares a spatial reference frame. The availability of a completely distributed, completely local, low cost, shared reference frame would open up algorithmic approaches previously not possible. By using Gaussian Belief Propagation [13], we can construct this within a swarm of robots based only on local observation and messaging. Robots move around, constructing a distributed, size-limited factor graph of observations of other robots and odometry information. Message passing within and between neighbouring robots results in convergence on a shared frame of reference; each robot knows where it is in relation to it. To demonstrate the potential of this, we focus on two applications, shape formation and logistics.

Shape formation. Shape or pattern formation in robot swarms is widely studied. The use of a swarm to perform a task is often implicitly or explicitly dependent upon that swarm maintaining a particular shape or covering a particular area. Swarm shape formation is a proxy for a number of real-world problems such as search and rescue or emergency communication. Many problems rely on the swarm maintaining a coherent shape or coverage of particular areas. This has been tackled in swarm robotics in a number of ways which in their essence involve the construction of a frame of reference, or coordinate system. These systems often rely on robots transitioning to a static state to serve as anchors for further extensions to the shape and coordinate system, or unrealistic assumptions of position knowledge, [51, 52, 38, 10].

Logistics. The use of swarms for intralogistics is an emerging area, where perhaps we can move beyond the lab into real-world applications. Swarm logistics can be regarded as a real-world application of foraging [62]. With our DOTS [30] robots we aim to demonstrate a simple but functional application where we specifically focus on the out-of-the-box, easily deployable solutions for everyday environments, as described in [29]. Even simple random walk algorithms are capable of effective retrieval in logistics applications [41] and adding spatial awareness allows for improved algorithms. Messages from users propagate from robot to robot, robots store and retrieve items, all in parallel and without central resources. Although simple, this scenario encapsulates many issues that will need to be solved for viable swarm logistics systems to become a reality.

2.1 Simulator

We work in simulation, with the intention of transferring the work to real robots in the future. As such, we use an abstract model of our real robots, the DOTS; each robot is modelled as a 2 kg disk, 250 mm in diameter, that can move holonomically at up to $v_{\max} = 1\,\mathrm{m\,s^{-1}}$, and can sense and identify other robots and objects to a distance of $r_{\mathrm{sense}} = 0.5\,\mathrm{m}$. Communication is possible between any robots that can sense each other. Each robot has imperfect sensors distorted by Gaussian noise; a velocity sensor $\vec{v}_{\mathrm{sense}} = GT_{v_{\mathrm{robot}}} \cdot \mathcal{N}(1, \sigma^2_{\mathrm{velocity}})$, and a relative

European Innovation Council
Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

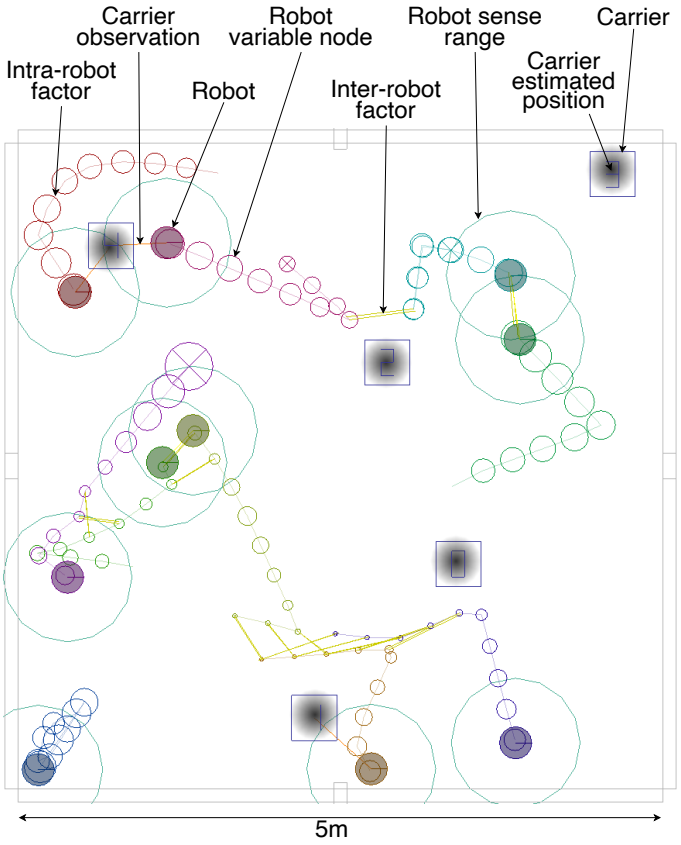

Figure 1: Simulation of robots performing DSA-RW to locate cargo carriers, with various elements of the visualisation labelled. Within the carrier squares are overlayed the current swarm estimates of the carrier position, already showing good correspondence after two minutes of simulated time.

position sensor $\vec{p}_{\text{object}} = GT_{\text{object}} - GT_{\text{robot}} + \mathcal{N}(0, \sigma^2_{\text{position}})$ where GT_k is the ground truth from the simulator. The robot maintains odometry $\vec{p}_{\text{odom}} = \sum \vec{v}_{\text{sense}}$, integrated since the last variable node was created.

The simulator is written in C++, using the Box2D physics engine. Updates to the simulation occur at 60 Hz. The arena is a square area with fixed walls. Robot motion is modelled as a disk with friction sliding on the arena surface, collisions between robots and with walls follow physics. Low level proportional control of the force applied to the robot body is used to satisfy the commanded velocity. As well as robots, there can be objects that can be detected by the robots. For each robot a list of robots and objects within r_{sense} range is maintained, and at each timestep a set of abstract senses is constructed and a controller routine is executed on those senses to generate a commanded velocity. Figure 1 shows a visualisation.

2.2 Gaussian Belief Propagation

Gaussian Belief Propagation (GBP) is a method of performing distributed iterative probabilistic inference or state estimation on a graph of relations between Gaussian variables by means of message passing. It is not new but has received recent attention, with convergence guarantees and applicability to distributed systems. Davison [13] makes the argument that the completely decentralised and incremental processing of GBP is a better fit for future systems as processing power becomes more distributed. This naturally fits with the swarm robotics paradigm, and

European Innovation Council Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

allows the inference of global properties by using purely local interactions.

For our work with the DOTS robots, we use only a linear 2D state representation, and only factors of one or two variables. This simplifies the mathematics, but the principle can and has been applied to non-linear representations such as $SE(2)$. Gaussian variables $\vec{x}_i$ in the moments form $\mathcal{N}(\vec{x}_i; \mu_i, \Sigma_i)$ can also be represented in the canonical/information form $\mathcal{N}^{-1}(\vec{x}_i; \eta_i, \Lambda_i)$, connected by the identities: $\Lambda = \Sigma^{-1}, \eta = \Lambda\mu$, where Λ is the precision matrix and η the information vector. Hereafter, when specifying components, we use the notation $(\eta, \Lambda)_G \equiv (\mu, \Sigma)_G$. Factor $f_j(\vec{x}_j)$ connects to a single variable and specifies a prior, defined by the Gaussian constraint $\vec{z}_j = (\eta_j, \Lambda_j)_G$. Factor $f_k(\vec{x}_{\mathrm{k1}}, \vec{x}_{\mathrm{k2}})$ connects two variables $\vec{x}_{\mathrm{k1}}, \vec{x}_{\mathrm{k2}}$ and specifies a measurement, comprising the functional relationship $\vec{h}_k(\vec{x}_{\mathrm{k1}}, \vec{x}_{\mathrm{k2}}) = \vec{x}_{\mathrm{k2}} - \vec{x}_{\mathrm{k1}}$, and the constraint $\vec{z}_k = (\eta_k, \Lambda_k)_G$. The GBP algorithm requires three steps: **factor-to-variable message** $m_{f\to x}$, **variable-to-factor-message** $m_{x\to f}$, and **belief update** $b(x)$. All messages are in the form $(\eta, \Lambda)_G$. The schedule of operations, and nodes upon which the operations take place, can be entirely arbitrary and asynchronous.

Belief update: A variable $\vec{x}_i$ has its belief updated as the product of messages from all connected factors. In canonical form, this is expressed as a sum:

$$b(\vec{x}_i) = \left(\sum_{f \in n(\vec{x}_i)} \eta_{f\to x}, \sum_{f \in n(\vec{x}_i)} \Lambda_{f\to x} \right)_G \tag{1}$$

where $n(\vec{x}_i)$ are all the factors connected to $\vec{x}_i$.

Variable-to-factor message: The message to a connected factor is the product of the incoming messages from all other connected factors:

$$m_{x\to f_j} = \left(\sum_{f \in n(\vec{x}_i)\setminus f_j} \eta_{f\to x}, \sum_{f \in n(\vec{x}_i)\setminus f_j} \Lambda_{f\to x} \right)_G \tag{2}$$

Factor-to-variable message: For a single variable factor, this is simply the factor.

$$f(\vec{x}) : m_{f\to x} = \vec{z} \tag{3}$$

For a two variable measurement factor, this is the product of the factor and the message from the other variable, marginalising out the other variable. Alternatively, the message to a variable is the precision weighted sum of the message from the other variable and the measurement vector $\vec{z}$.

$$\text{Let } \alpha_p = \frac{\Lambda_f}{\Lambda_f + \Lambda_p} \tag{4}$$

$$f(\vec{x}_i, \vec{x}_j) : m_{f\to x_i} = \big((1-\alpha_{x_j})\eta_f + \alpha_{x_j}\eta_{x_j}), \alpha_{x_j}\Lambda_{x_j}\big) \qquad \text{message to } \vec{x}_i \tag{5}$$

$$f(\vec{x}_i, \vec{x}_j) : m_{f\to x_j} = ((1-\alpha_{x_i})\eta_f + \alpha_{x_i}\eta_{x_i}), \alpha_{x_i}\Lambda_{x_i}) \qquad \text{message to } \vec{x}_j \tag{6}$$

One modification we make to the original algorithm is to note that:

$$m_{x\to f_j} = b(\vec{x}_i) - m_{f_j\to x} \tag{7}$$

Variable nodes just calculate their beliefs and send them as messages, and factor nodes locally calculate what the variable-to-factor message would have been by subtracting the last message sent to that variable. This minimises the non-local knowledge needed in any node.

European Innovation Council Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

Table 1: Simulation parameters

Parameter	Value	Description
n_{window}	10	Max number of variable nodes in local factor graph
t_{node}	0.5 s	New variable node creation interval
r_{sense}	0.5 m	Object and robot sense and communication radius
σ_{velocity}	0.1 m/m	Velocity sense noise (metres per metre travelled)
σ_{position}	0.02 m	Position sense noise
v_{fast}	0.5 ms^{-1}	Movement performing DSA-RW and DSA-KE
v_{slow}	0.05 ms^{-1}	Movement within shape while performing DSA-SF

2.3 Distributed factor graph construction

We construct a shared reference frame for the swarm in the following way: Each robot builds a local factor graph with variables representing 2D pose, unary anchor factors connecting to a single variable, and binary relative measurement factors connecting two variables. Each robot shares synchronised time, and at regular intervals t_{node} from a starting epoch a new timestep ts_i is issued and a new variable node $\vec{x}_i$ is created. At any given time, the pose of the robot relative to the shared reference frame is $\vec{p}_{\text{robot}} = \vec{x}_i + \vec{p}_{\text{odom}}$.

The first node to be created will have an anchor factor connected to it with a weak prior of pose $(0, 0)$. Successive nodes have a measurement factor connecting them: $f_{\vec{x}_i \to \vec{x}_{i-1}}(\mu = \vec{p}_{\text{odom}}, \Sigma = \sigma^2_{\text{velocity}} I_2)$. As new variable nodes are created, old ones are removed to maintain a maximum number of nodes n_{window}, with the now oldest variable having an anchor factor attached to it set to the belief of that variable. At every t_{message} interval, a factor node is chosen at random, and messages are propagated to each connected variable according to the GBP algorithm. Each connected variable then has its belief updated.

As described, this local factor graph is purely a bounded time window sampling of the odometry of the robot. In order to connect the local factor graphs together into a swarm whole, we add two things: 1) Robots observe other robots within their sensory range and create measurement factors $f(\mu = \vec{p}_{\text{object}}, \Sigma = \sigma^2_{\text{position}} I_2)$ that link the current variable node on the observing robot to the observed robot. These factors, termed outward-facing, are given the current timestep ts_i and the ID of the other robot. This uniquely identifies the variable at the other robot that it links to. 2) Robots in sensory range exchange GBP messages. At the same t_{message} interval as above, the robot chooses a single other robot from any within r_{sense} range and sends a message request. This consists of a list of the timesteps of all outward-facing factors that connect to the other robot. The responding robot replies with a list of beliefs from the variables those factors uniquely connect to, and a list of messages from remote factors that connect to local variables of the requesting robot.

The factors linking variable nodes on different robots create the constraints that produce convergence of local pose estimates into a shared consistent state, i.e a shared reference frame. This process is illustrated in Figure 2. Assume that robot x has sensed robot y, this is the process that follows. It sends a list of the timesteps of outward-facing factors connected to robot y, which will just be $\{6\}$, for the factor connecting x_6 to y_6. Robot y replies with

European Innovation Council
Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

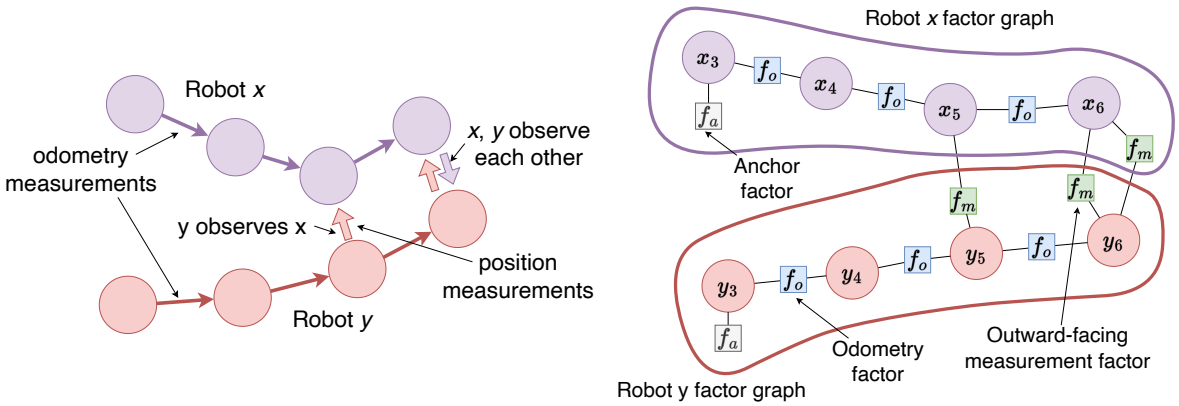

Figure 2: Robots creating connected factor graphs. Left: Two robots move on trajectories, making odometry measurements, that bring them within sensing range of each other. Right: The internal factor graphs that are created on each robot, assuming $n_{\text{window}} = 4$ and some time has passed. The oldest variable node in each graph, x_3, y_3, has an anchor factor. Between each variable node is an odometry measurement factor node. In timestep ts_5, robot y has observed robot x, creating an outward-facing relative position measurement factor. In ts_6, both robots have observed each other.

$\{b(y_6), m_{f\to x_6}, m_{f\to x_5}\}$. Robot x has now got message information needed to update the beliefs of x_5, x_6, and to create an outward message from f_m attached to x_6. Relevant parameters are detailed in Table 2.

2.4 Convergence measures and proxies

As robots encounter each other and exchange messages, and individual robots perform message passing on their individual fragments of the factor graph, the pose of each robot becomes constrained against others, thus the assumed origin or reference frame of each robot converges, Figure 3. This process is dynamic, and will never completely converge, since robots are moving and have noisy perceptions, and the whole graph does not remain fully connected. The error is the mean deviation from the swarm centroid of the robot origins:

$$r_{\text{error}} = \frac{1}{n_{\text{robots}}} \sum_{i=1}^{n_{\text{robots}}} |\vec{r}^{\,i}_{\text{origin}} - \mu_{\text{origin}}| \tag{8}$$

This is only knowable from the global perspective of the simulator. We define the convergence time t_{conv} as the time taken for error to reach $r_{\text{error}} < 2\sigma_{\text{position}}$ since position observation noise dominates convergence error.

To make the shared reference frame useful to a swarm system, individual agents within the swarm need to know when they can rely on estimates but they have no access to the global measure r_{error}. In characterising the system, we collect data on the time taken for agents to encounter at least half the robots in the swarm, denoted $t_{\text{met\_half}}$. We reason that this, with some constant, should be a reasonable proxy for the time taken to build a fully converged reference frame:

$$t_{\text{convproxy}} = \beta \cdot t_{\text{met\_half}} \tag{9}$$

To implement this, each robot keeps a count of the number of unique other robots it has encountered. When this exceeds half the size of the swarm, the time is noted and $t_{\text{proxyconv}}$

European Innovation Council
Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

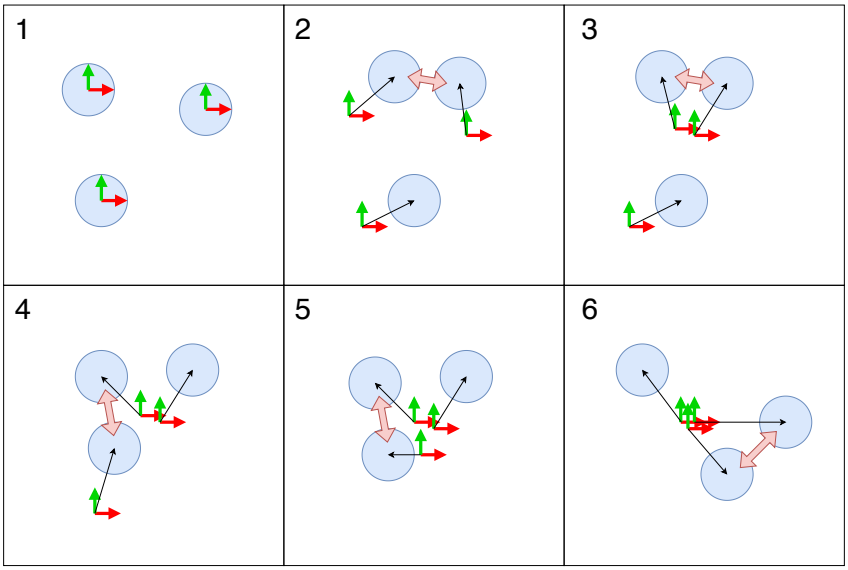

Figure 3: Illustration of shared reference frame convergence. 1) All robots start by thinking they are at the centre of the swarm. 2) Observation and communication imposes constraints on location of the swarm centroid; the top two robots communicate.. 3) .. and each robot updates its own estimate. 4) More communication imposes further constraints. 5) Origin estimates move closer.. 6) ..and approach convergence.

calculated. Only when the elapsed time is greater than this is it possible to use the shared reference frame derived $\vec{p}_{\text{robot}}$. The two example algorithms below wait for this before switching behaviour from the baseline random walk controller DSA-RW.

2.5 Simulation results

We cover three areas. Firstly, we examine the computation, bandwidth, and convergence properties across many different scenarios. These allow us to develop a proxy measure of convergence, indicating when it is meaningful to use the shared reference frame. Then we look at performance with a simple shape formation algorithm, and finally, a logistics application, where distributed and up-to-date information about dynamic objects in the environment is acquired, both with random movement, and also with actively seeking areas of low knowledge to be explored.

Parameterisation We examine how long the distributed factor graph takes to converge, t_{conv}, and how much computation and bandwidth is used in different scenarios.

We look at different t_{message} update periods of the factor graph, and different numbers of robots in an arena sized to keep a constant robot density of $0.4\,\text{m}^{-2}$. See Figure 4. Convergence takes longer when there are robots over a larger area, this is expected, since all robots have to be able to influence each other, though not necessarily by direct communication. Density has little effect on convergence, the dominating factors are update rate and arena area. There is little gain from rapid updates over 10 Hz.

Computation and bandwidth are both proportional to update frequency so we use a fixed sized arena of $25\,\text{m}^2$ and update rate of 10 Hz to look at other factors. All operations are performed in 32 bit floating point. Since we are working in 2D space with linear factors, the precision matrix Λ can be represented as a single number. Belief update needs 3 operations per attached

European Innovation Council Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

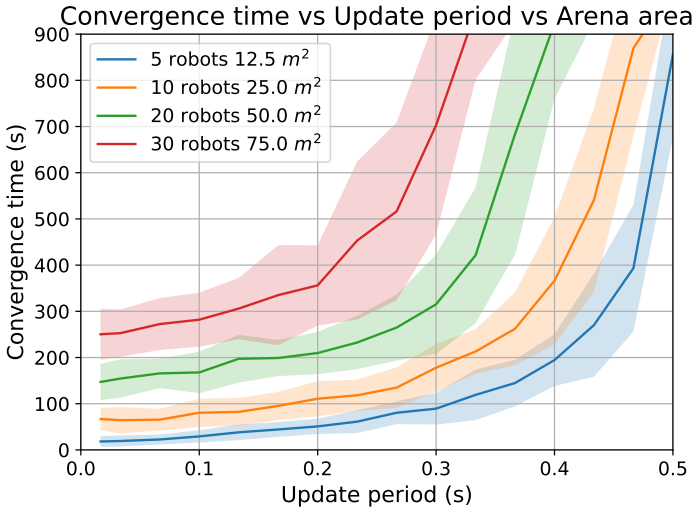

Figure 4: Convergence time with different factor graph update rates and different arena areas, with fixed robot density. Reduction in convergence time is minimal below 0.1 s update period.

Table 2: Computation, bandwidth, and convergence time for different densities of robots within a 5 m × 5 m arena, with updates at 10 Hz. Resource usage is low, making it possible to run the DSA algorithm on even very limited computational hardware.

Robots	Density (robots/m^2)	FLOPs	Bandwidth (bytes/s)	T_{conv} (s)
12	0.5	650	380	68
25	1.0	780	730	78
50	2.0	800	1100	89
100	4.0	900	1600	140

factor (Eqn 1). Factor message generation only needs computation for measurement factors; $2(\text{Eqn 4}) + 1 + 2 + 2 + 1(\text{Eqn 5}) + 2 + 2 + 1(\text{Eqn 6}) = 13$ operations. Assuming a naive message protocol, every request has an overhead of 12 bytes, and 4 bytes per outward-facing factor, and each response has an overhead of 12 bytes, and 16 bytes per returning belief and factor-to-variable message.

There is a weak dependence of computation on density, as there are more chances to create additional outward-facing factors. Bandwidth is strongly dependent on robot density, as there are many more opportunities for a robot to communicate, Table 2. It should be noted that the raw figures for supporting the shared reference frame are remarkably low; for a 25 m^2 arena, a small swarm will converge in less than 70 s, with each robot using only a few hundred floating point operations and exchanged message bytes per second. This is achievable even on low cost processors.

As noted above, we need a local proxy for the convergence time t_{conv}. In order to determine an appropriate value for β (Eqn 9), we ran a set simulations over different numbers of robots between 5 and 100, and different arena sizes between 4 m^2 and 100 m^2, fixing the update period $t_{\text{message}} = 0.1$ s. Simulations were run for 1000 simulated seconds. Out of 4200 simulations, 2886 were able to find an initial placement for the robots in the arena size, and

European Innovation Council
Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

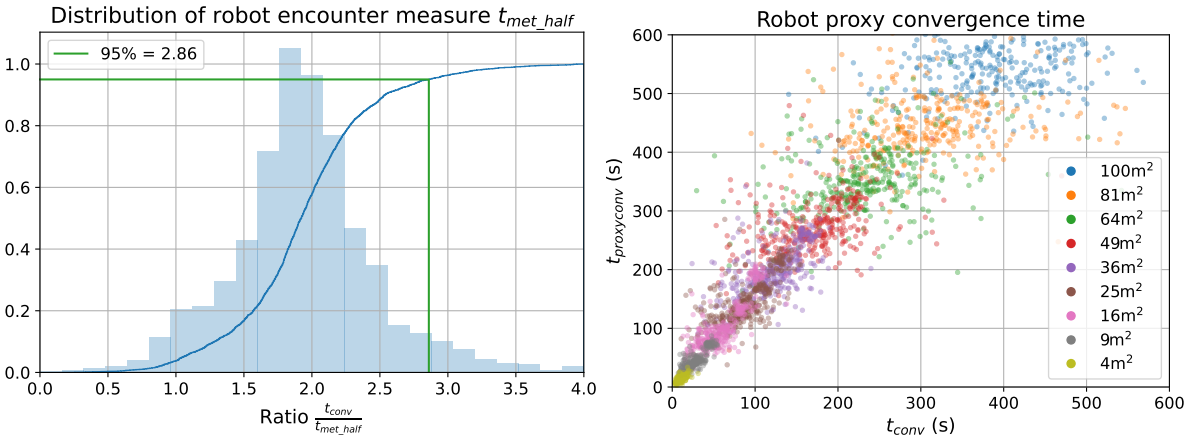

Figure 5: The distribution of the robot encounter measure $t_{\text{met\_half}}$ over 2886 simulations with different numbers of robots between 5 and 100, and different arena sizes between 4 m^2 and 100 m^2. Using a constant $\beta = 3$ ensures the proxy convergence time $t_{\text{proxyconv}}$ exceeds the true convergence time t_{conv} in more than 95% of simulation. Right scatter plot coloured according to arena area.

reached convergence within the simulation run time. We can see from the graphs in Figure 5 that using a value of $\beta = 3$ ensures the proxy measure $t_{\text{proxyconv}}$ exceeds the true measure t_{conv} in 95% of simulations.

Shape formation Because each robot has access to the shared reference frame, it is easy to construct algorithms for swarm-wide shape formation. To demonstrate this, we use a simple algorithm called DSA-SF (Distributed Spatial Awareness - Shape Formation) where the shape is defined functionally, $f_{\text{in\_shape}}(p_{\text{robot}})$, and shown in Algorithm 1. Each robot has two modes of behaviour; when not inside the shape according to its estimated position $\vec{p}_{\text{robot}}$, it uses the baseline behaviour DSA-RW. When inside the shape, it slows down to $v_{\text{slow}} = 0.1 \cdot v_{\text{fast}}$, and if there are any neighbours, it moves towards them, causing classic swarm aggregation. We

Algorithm 1 DSA-SF Shape formation

```
if not IN_SHAPE(p_robot) then
    v⃗_cmd_vel = v_fast · DSA-RW_direction
else
    if n_neighbours > 0 then
        v⃗_cmd_vel = (v_slow, ∠_neighbours)
    else
        v⃗_cmd_vel = v_slow · DSA-RW_direction
    end if
end if
```

define some simple shapes such as a circle, vertical and horizontal lines, and wavy lines, and switch between them at intervals.

We ran simulations using 150 robots, with an arena size of 7.5 m per side. After reaching convergence, each robot followed Algorithm 1, with the shape function switching at regular intervals. Figure 6 shows snapshots of the process, with the desired shapes emerging within about 40 seconds in each case.

The algorithm is extremely simple, yet forms and cleanly transitions between shapes even

European Innovation Council Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

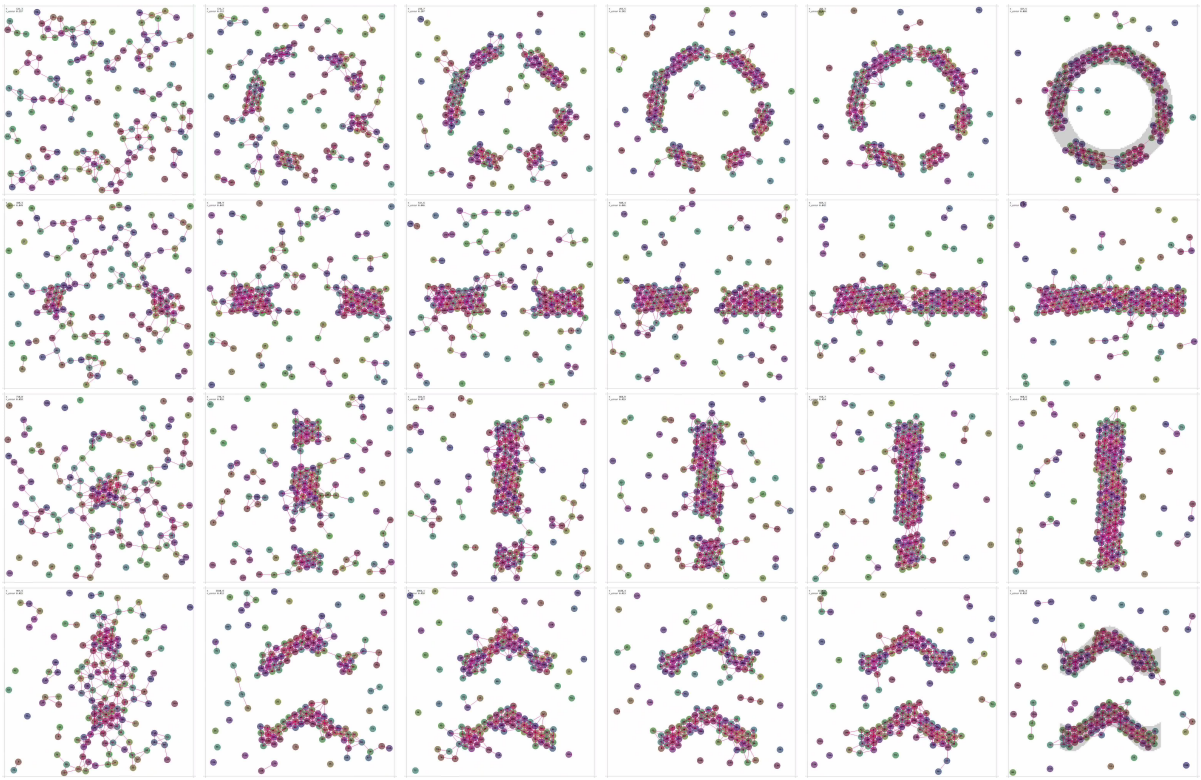

Figure 6: Shape formation. 150 robots perform random walk DSA-RW when outside shape, and slow down and aggregate when inside shape, taking about 40 seconds to form each pattern.

without fine tuning of parameters.

Intralogistics As noted above, even random walkers are capable of performing simple swarm logistics operations. We want to enhance the performance of such systems using the shared reference frame. A key component to a logistics system is knowledge of the location of cargo carriers. Carriers C are dynamic objects that may move. Each robot maintains a set of Gaussian variables and an associated time of observation $c(i) \equiv (\vec{x}^i_{\text{carrier}}, t^i_{\text{observed}}), i \in C$, one for each possible carrier. When a robot observes a carrier k, it sets the tuple $c(k) = ((\vec{p}_{\text{robot}} + \vec{p}_{\text{carrier}}, \sigma^2_{\text{position}} + \sigma^2_{\text{robot}})_G, t)$. Each time a robot exchanges messages with another robot, it sends a list of the observation times $t^i_{\text{observed}}, i \in C$ it has, including $t = 0$ for carriers for which it has no observation. The other robot replies with any carrier observations it has that are more recent, these are used to replace the older observations. More recent observations are privileged, even if they may have greater positional uncertainty, since the carrier may have moved.

Algorithm 2 DSA-KE Knowledge Enhancement

if any $t^i_{observed} = 0, i \in C$ **then**
  $\vec{v}_{\text{cmd\_vel}} = v_{fast} \cdot \text{DSA-RW}_{\text{direction}}$
else
  $j = argmin(t^i_{\text{observed}}, i \in C)$
  $\vec{d} = \vec{x}^j_{\text{carrier}} - \vec{p}_{\text{robot}}$
  $\vec{v}_{\text{cmd\_vel}} = v_{\text{fast}} \cdot \vec{d}$
end if

European Innovation Council
Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

The way the robots move has an impact on the acquisition of knowledge about the environment. When the swarm has no knowledge, the goal is to cover as much of the arena area as possible. For this, we use the baseline behaviour DSA-RW. Once the swarm has a certain level of awareness, it becomes possible to use this information to guide swarm exploration so as to maximise knowledge. This behaviour is called DSA-KE (Distributed Spatial Awareness - Knowledge Enhancement), and is identical to DSA-RW until a robot has some information about all the carriers in the environment. At that point, instead of choosing a direction at random, a robot will choose the direction of the carrier which it has oldest knowledge about, thus the swarm as a whole seeks to minimise overall uncertainty, Algorithm 2.

In order to test the quality of knowledge acquisition in a dynamic environment, we make the carriers move position. This is specified with a single parameter, the aggregate mean carrier velocity $v_{\text{c\_agg}}$. Given n_{carrier} carriers, a carrier is selected at random and moved a fixed distance of 1 m into a random location at a velocity of $v_{\text{c\_agg}} \cdot n_{\text{carrier}}$. The mean perception error of the swarm is given by:

$$s_{\text{error}} = \frac{1}{n_{\text{robots}} n_{\text{carrier}}} \sum_{j=1}^{n_{\text{robots}}} \sum_{i=1}^{n_{\text{carrier}}} |c(i)^j_{\text{est}} - c(i)_{\text{gt}}| \tag{10}$$

where $c(i)_{\text{gt}}$ is the ground truth position of a carrier, transformed into the swarm frame.

Figure 7 shows how the two movement behaviours DSA-RW and DSA-KE perform in acquiring knowledge about the dynamic carrier environment. At zero carrier velocities, the swarm quickly reaches low s_{error} values, around 0.04 m. Since the position sense has injected noise of $\sigma_{\text{psense}} = 0.02m$, this is a reasonable lower bound. As the velocity of the carrier movement increases, the error goes up as unobserved carriers can move further from their last known positions.

The effectiveness of DSA-KE in improving carrier position estimation in the swarm by actively moving to areas where knowledge is weaker, is clearly apparent. In all cases over different numbers of carriers and different carrier velocities, we see improvement, 38% with 10 carriers and mean velocity 0.01 ms^{-1}. When looking at performance vs the number of carriers, we see that DSA-RW is roughly constant, whereas DSA-KE performs extremely well with lower numbers of carriers. The good relative performance of DSA-KE falls off at higher numbers of carriers, so it may be that a better strategy for high carrier numbers would ensure a greater degree of dispersion.

2.6 Discussion and conclusion

It is important to note that both DSA-KE (Knowledge Enhancement) and DSA-SF (Shape Formation) are simple algorithms that serve to illustrate the possibilities that distributed spatial awareness can offer for swarm algorithms. The behaviour of the GBP message passing algorithm underlying the DSA shared reference frame is robust across parameters and the computational cost per robot of maintaining its local factor graph is low, of the order of a few hundred floating point operations per second, and the communications likewise is low cost, being a few hundred bytes per second. This is within reach of even cheap microcontrollers. But this low-cost outlay provides a rich, entirely decentralised, pseudo-global source of information not typically available to agents within swarms.

We consider GBP to be a powerful technique to infer global knowledge within a completely distributed paradigm, and wish to bring other items of state and knowledge within this over-arching framework; it will be interesting to compare some more traditional swarm consensus

European Innovation Council Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

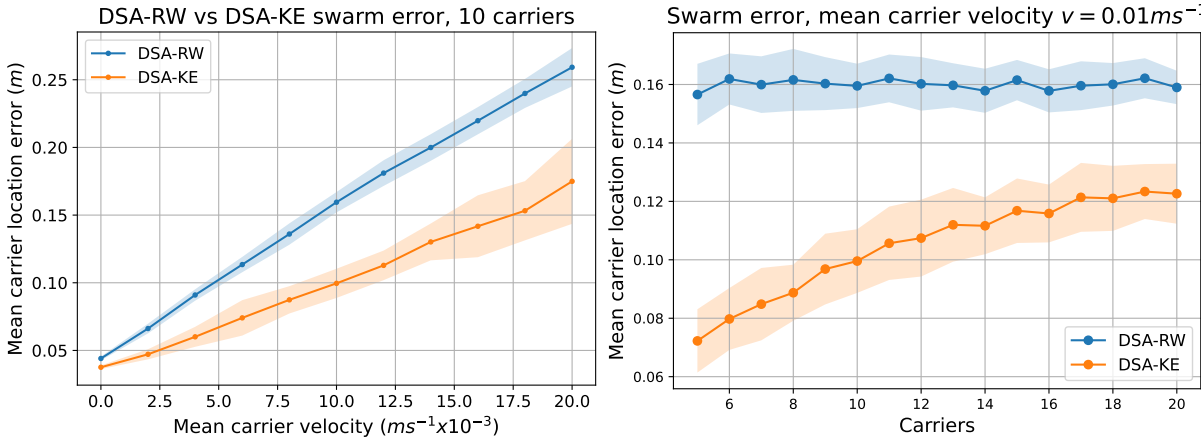

Figure 7: Swarm carrier location error s_{error} with the two robot behaviours while varying the carrier velocity (left) and the number of carriers in the arena (right). The number of robots is fixed at 10. Shaded areas indicate $\pm\sigma$ over 50 different random seeds. As the mean carrier velocity increases, so does the error, which is expected. With a fixed mean carrier velocity and sweeping the number of carriers, we see roughly constant error with the random walk of DSA-RW, but much lower error with DSA-KE as it actively seeks to enhance knowledge, particularly with low carrier numbers. In all cases, the DSA-KE behaviour results in much lower swarm perception error.

algorithms with GBP, many best-of-n swarm algorithms rely on local message passing, perhaps suggesting deeper similarities.

We look forward to exploring new swarm algorithm possibilities that may combine DSA acquired knowledge with other swarm techniques. The current simulation system relies on graph construction happening in synchronisation at regular timesteps, but we don't see this as necessary and plan to remove this restriction, e.g. by estimating between states at time of observation. Finally, we are planning to implement the algorithm on the DOTS swarm in reality as part of our logistics demonstration task.

3 Using awareness metrics to select optimal levels of awareness

Measuring awareness across different systems remains an unresolved challenge. Yet, we argue that it holds untapped potential for enhancing the design, control, and effectiveness of artificial systems. We propose a novel and tractable approach to measuring the impact of the presence of varying degrees of awareness on system performance. Specifically, we demonstrate this approach through a swarm robotics intralogistics scenario, where we assess the influence of two dimensions of awareness - spatial and self - on the performance of the swarm in a collective transport task. Our results reveal how increased abilities along these awareness dimensions affect overall swarm efficiency. This application represents an initial step towards quantifying and controlling awareness in artificial systems and future implementations across different systems [6, 16].

Our focus in this research is on the ***practical measurability*** of awareness, specifically how the presence of more or less awareness impacts performance in a collective of agents. The emphasis here lies not on assessing or formulating the minimal conditions for awareness to be present, but rather under the assumption that there might be some awareness present, we focus on examining how varying the degrees of awareness along several dimensions affect

European Innovation Council
Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

performance. In concrete terms, this entails a functional approach to awareness, referencing an entity's ability to process and respond to its environment, itself, and others through various action-perception abilities. For example, a robot might possess a degree of spatial awareness allowing it to reliably and effectively locate and retrieve objects in space. Importantly, we conceive of awareness as a continuous spectrum rather than a binary on/off property. Furthermore, assuming awareness cannot be captured by a single measure, it must be understood within a multidimensional framework. By operationalizing the concept as such, we aim to provide a practical way to evaluate and compare artificial systems [28, 35, 7].

From these motivations we draw the following aims: first, we provide a framework to measure the changes in the behaviour or performance of an artificial swarm system based on different dimensions of awareness [42]. Secondly, rather than approaching awareness directly we approach it in terms of the abilities it encompasses [25]. This is in contrast to a focus primarily on task completion for artificial systems. We therefore introduce terminology for the study of awareness which will provide us with tools to examine awareness in a given context for any artificial system. Finally, we demonstrate a practical implementation of the framework in a concrete use-case for swarm robotics. In this case, a framework for the examination of awareness may be leveraged to enhance the capabilities of the swarm to achieve certain tasks, or to compare the design of one swarm system to another. Our framework provides a starting point for such an application.

3.1 Levels of awareness

Here we define different concepts and terminology related to awareness that are necessary to describe and implement our framework. Figure 9 summarises the concepts described below.

Dimensions of awareness: In proposing a multidimensional account of awareness, our focus lies not on mapping all the things of which a system is aware, but rather on delineating the ability space of a given system in a manner that enables comparison across different systems. Underlying this approach is the notion that awareness cannot be captured along a single dimension of measurement. By specifying dimensions we enable comparison between different systems without assuming such a single linear standard, instead allowing for a more flexible and nuanced approach to measuring awareness. This is particularly important for assessing different artificial systems, e.g. swarms versus artificial neural networks, since they may require highly different evaluation criteria. A multidimensional approach to measuring awareness therefore makes meaningful comparisons across such heterogeneous systems possible, see Figure 8.

The dimensions we focus on as an example are spatial and self-awareness. Other dimensions which may be considered include temporal, agentive and meta-cognitive. We leave a full exploration of these dimensions to future work, noting that this list is neither exhaustive nor complete - additional dimensions may be identified, and existing ones may prove interrelated.

Our focus on spatial and self-awareness stems from their operational and tractable applicability to the swarm use-case discussed here. Spatial awareness refers to a systems ability to sense and react to its spatial environment in relation to physical locations of itself and other entities within that environment. This might involve abilities such as navigation, localization, and coordination of movements or actions in space. Self-awareness is explored here in its minimal instantiation as bodily awareness, defined as a system's ability to monitor its own physical states. In artificial systems, this could involve the ability to detect and adapt to changes in its structural integrity (e.g. the occurrence of faults), energy levels, or operational limits.

European Innovation Council Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

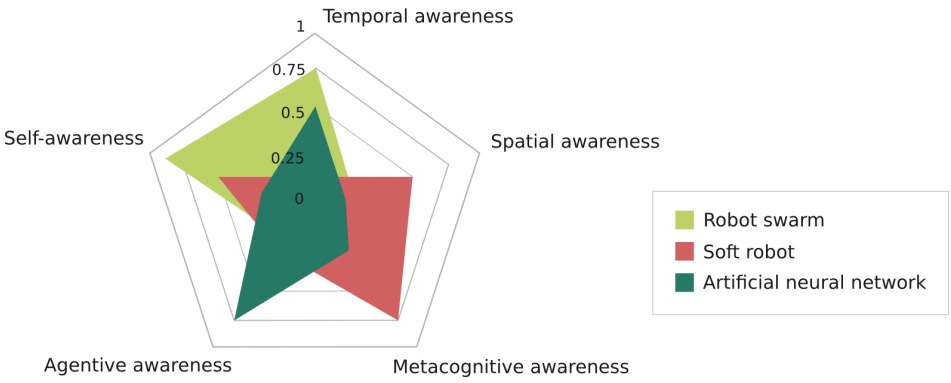

Figure 8: Example awareness profiles for three different artificial systems across five dimensions. The profiles are a visual demonstration of where each system sits in the multi-dimensional space of awareness.

Capacities: Each dimension of awareness must be associated with an underlying set of abilities, which we refer to here as ***capacities***. Capacities refer to a system's dispositional properties to succeed in specific tasks - similar to how a person may exhibit varying levels of skill on tasks such as pattern recognition, spatial navigation, memory retrieval, or even a game like basketball. In other words, capacities refer to what a system is ***able to do***, and to what extent [23], with performance evaluated against established success conditions. Along any given dimension, a system possesses certain capacities to varying degrees, and we expect differences in its operation or performance to reflect these variations in capacities. Instead of cataloging all potential capacities, we focus on a few specific capacities within the dimensions of interest to explore how they affect system performance. In particular, we investigate whether a system's performance on a given task entails reliance on capacities along either of the two dimensions of interest, posing the following key questions: to what extent does a system rely on spatial or bodily capacities for its functioning in a given context? And how might we measure these capacities in practice?

Mechanisms: Mechanisms are the means by which capacities are enabled in a system. For example, in a robot, the capacity for navigation might rely on optical sensors and motors. The same optical sensors could also support the capacity to detect and identify objects in the environment. While mechanisms refer to the components of a system - like sensors, algorithms or motor systems - capacities refer to what the system can do as a result, such as successfully navigating a maze or categorizing objects. There is therefore a many-to-many relationship between mechanisms and capacities: a single mechanism can support multiple capacities, and a single capacity may be instantiated through various distinct or overlapping mechanisms.

Evaluation tasks: Tasks are chosen in order to evaluate the performance of the system with respect to the set of identified capacities, which themselves are associated with the dimensions of interest. In selecting a task, we must be able to reasonably assume that the associated dimensions are present to some degree and necessary for the swarm to perform the task. We can then measure the effect of disabling capacities on swarm performance as an indirect measure of the effect of associated dimensions.

Performance metrics: We can evaluate the performance of a system for a given task using defined metrics. The definition of such metrics will be task specific and we do not provide a

European Innovation Council Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

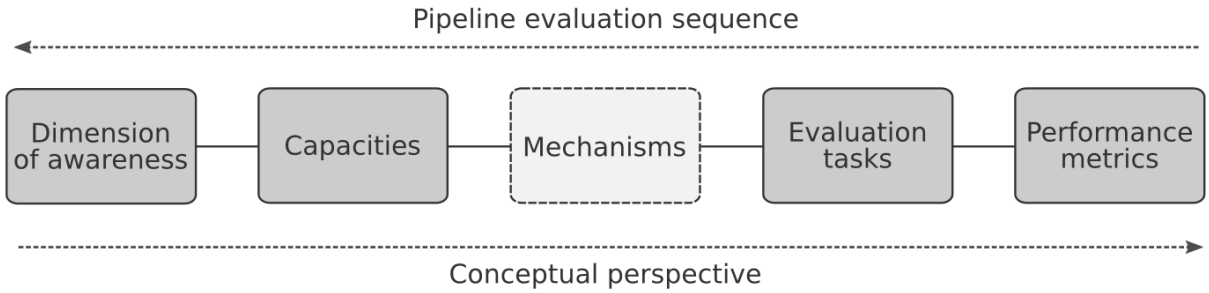

Figure 9: Visualisation of the relations between concepts in the framework. We illustrate two alternate perspectives for the sequence of concepts, arrows indicating the direction of order: the pipeline evaluation starts from the performance metrics progressing from right to left whereas the conceptual perspective progresses in the opposite direction starting from the dimensions.

list of metrics here but assume that they come with the given tasks (to be specified by the engineer/designer of the system). The performance on a given dimension can be compared for a system when a capacity is present and when it is not.

3.2 Implementation pipeline

The concepts defined above and visualised in Figure 9 can now be applied to achieve our aim: to evaluate the performance of the swarm with respect to capacities associated to the dimensions of awareness that we wish to examine. More specifically, we are interested in how performance in a given task changes depending on whether the capacities of interest are present or not present. In practice, the evaluation of the above involves selecting a set of capacities associated to the dimensions of interest, together with a task which allows for such capacities to be evaluated with a performance delta. A single task is sufficient for the implementation in this work. The following is a description of an implementation pipeline for the framework in four steps:

1. The first step is to specify the capacities C of the system associated with the dimensions D of interest. For example, for spatial awareness in a robot an associated capacity might be the ability to detect objects. In particular, capacities are realised through mechanisms available to the robot such as on-board cameras and IR laser sensors. It must be possible to disable the capacities of interest in order that performance can be evaluated when they are present and absent. To do so in practice means disabling the mechanisms on which the capacities are dependent.
2. Following from above, an intermediate step is to identify the mechanisms through which the capacities of interest are realised. From this point onwards, we abstract the mechanisms and refer only to capacities with the assumption that mechanisms exist in the robot that may be disabled, and such that it is possible to identify the dependencies of capacities on them.
3. A task $\mathcal{T}$ is selected such that we can reasonably assume that the dimensions of awareness we are interested have a non-zero impact on the swarm performance. The motivation here is to select a task which provides meaningful information about how capacities affect performance. For example, in an area-coverage task a robot may not make use of its ability to manipulate physical objects - this capacity cannot therefore be evaluated in such a task. We make the assumption here that there exists a task which satisfies our requirements.

European Innovation Council Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

4. Finally, performance metrics M are selected to evaluate the system with and without the presence of each capacity for the selected task. Then, for each capacity and each metric, the delta in performance is evaluated for the swarm performing the task. A high positive delta in performance suggests that the capacity has a strong positive contribution to swarm performance in the given task. We also allow for the case where a capacity may have a negative impact on performance - our interest is focused on relative performance when varying the capacities present.

This implementation pipeline provides us with concrete steps to realise our aims. It is defined in a general way such that it may be applied to any swarm system and dimensions of interest.

3.3 System scenario

Definition of capacities We now have all the ingredients required to implement our framework. Following the first step in the pipeline, we identify the capacities associated to the dimensions of interest: spatial and bodily awareness. In particular, we focus on the awareness and associated capacities for individual robots in the swarm. When enabling and disabling capacities, we do so for every robot in the swarm. The presence or absence of capacities at the level of the individual then produces an additive, or super-additive, impact on the overall swarm performance. We are interested in the following capacities for each dimension of awareness:

1. For spatial awareness, we consider the capacity for robots to locate boxes in the arena. One way in which this can be implemented is with Distributed Spatial Awareness system described in Section 2 which allows robots in a swarm to share and update their beliefs in a distributed manner. The accuracy of beliefs increases over time. As a proxy for the varying capacity of robots to localise objects in a physical space, we adjust the information available to the robot. The varying levels of "skill" a robot might have in this regard are implemented by introducing different levels of certainty in the information they possess. In the first level, robots know the quadrant of the arena that a box is located in. In the second level, robots know the precise coordinates of the centre of each box. Robots act on this information by navigating to the coordinates of the quadrant or the box, for each level respectively. We label these capacities: *knowledge of quadrant* and *knowledge of box*.
2. For bodily awareness, we consider the capacity of robots to self-detect and self-diagnose the presence of a fault. Again we consider two levels: in the first, a robot is able to self-detect the presence of a fault with a binary classification (faulty or non-faulty) without information of the type of fault. In the second level, a robot is able to self-diagnose the type of fault which has occurred. We note here that the second level of fault diagnosis necessarily depends on the first level, the capacity to detect faults. Fault detection is thus subsumed in the fault diagnosis capacity. In both cases, robots attempt to apply mitigation actions to reduce the negative impact of a fault. In the case of diagnosis, we assign the action to be applied depending on the diagnosed fault type: 1) a wheel fault causing a robot to move at 10% speed is mitigated by the robot moving towards a detected box, and 2) a lifter fault preventing a robot from picking up a box is mitigated by a robot opting to leave any team it has joined. These actions have been shown to be effective in mitigating the respective target faults [35]. In the case where a robot is only able to self-detect faults without diagnosis, it will select one of the two mitigation actions at random. The two capacities associated with bodily awareness are labelled *fault detection* and *fault diagnosis*.

European Innovation Council
Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

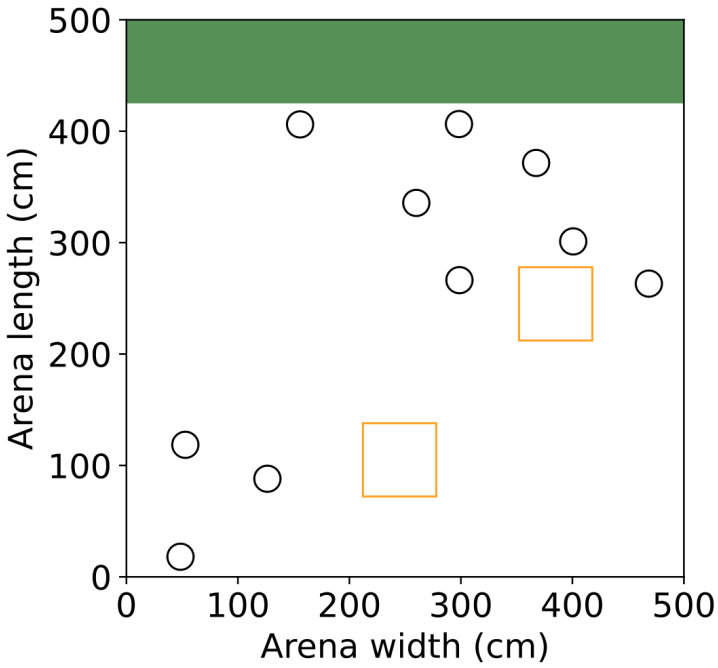

Figure 10: Arena setup: robots are represented by circles, boxes are represented by large orange squares. The deposit zone is the green-shaded strip at the upper boundary of the arena.

Scenario and Task Following identification of the capacities, we select a task where the capacities have a non-negligible impact on swarm performance. An intralogistics scenario provides a suitable task for our purposes as successful completion depends on the robots in the swarm having some degree of spatial and self-awareness. It is also a practical and realistic scenario in which to apply the implementation pipeline and produce a meaningful evaluation. In this scenario, the robots operate in a 5 m × 5 m bounded arena where they are tasked with retrieving and delivering boxes to a designated drop-off zone. The drop-off zone is a 0.75 m deep strip which extends along the width of the upper boundary of the arena. A robot can detect objects such as a box, another robot, or wall, via fiducial tags. In particular, robots perform collective transport where a team of four robots is required to transport each box. It is assumed that one robot takes position at each corner of the box. Boxes are removed from the arena upon successful delivery. They are also respawned every 20 s up to a limit of two boxes being present in the arena at any given time. The total number of respawned boxes allowed in a single trial is 20. The parameters of this scenario are selected to match the real-world robot platform and arena available to us as closely as possible with the aim to close the reality-gap in real-world trials. The scenario configuration is summarised in Figure 10 and Table 3.

There are four main stages of execution in the controller for collective transport behaviour. First, robots perform a random walk in the initial stage of searching for a box. Once a box has been found, a robot will position itself under the box if there is a free pickup spot and wait for a complete team of four robots to form. Upon completion of the team, each robot will independently "vote" for the next collective action: *lift box*, *move*, or *deposit box*. When a box is successfully deposited, the team is disbanded and robots return to the initial stage of searching for a box. Figure 11 shows the details of the collective transport controller.

Additionally, during real-world operation, faults will inevitably accumulate in the swarm [9, 3, 36]. We simulate the occurrence of faults by injecting faults with a given probability $p = 0.002$ at each timestep. There are two possible fault types: a wheel fault which causes a robot to

European Innovation Council Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

Table 3: Scenario configuration

	Property	Value
Arena	Dimensions	5 m × 5 m
	Number of boxes in arena	2
	Maximum boxes to respawn	20
	Number of robots	10
	Box size	0.7 m
	Deposit zone	0.75 m deep horizontal strip along upper boundary
Robot	Diameter	0.25 m
	Cameras	4 x 120°FOV video cameras equidistant on perimeter, 1 m range, used to detect objects; 1 x 120°FOV camera upward-facing for precise positioning under boxes.
	Proximity	16 x IR laser ToF, 3 m range, for collision avoidance
	Communication	Bluetooth, 1 m range
	Robot max speed	2 m s^{-1}

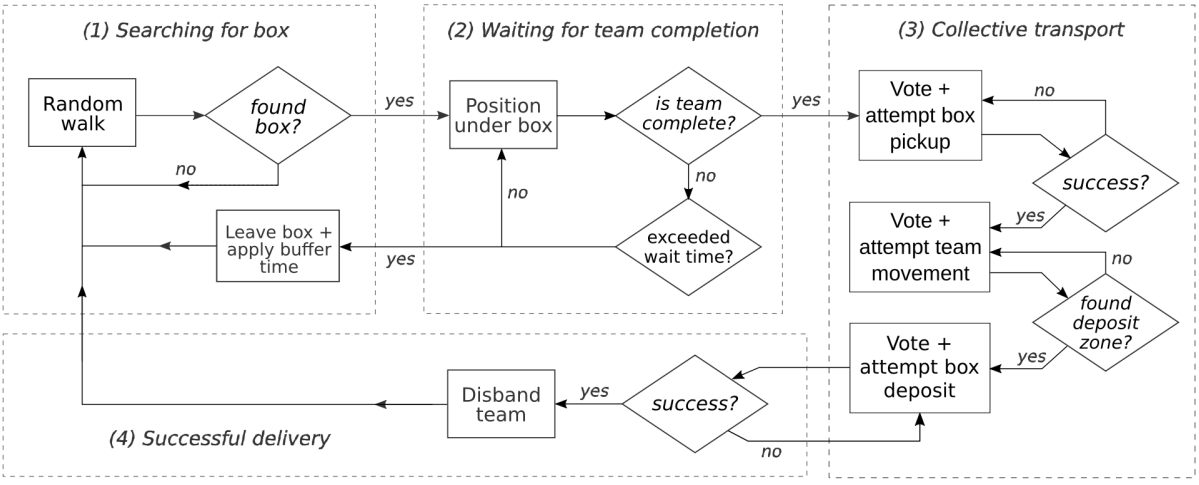

Figure 11: The collective transport controller: there are four main stages of execution which are labelled in the figure.

European Innovation Council
Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

move at 10% of its maximum speed, and a lifter fault which results in a robot being unable to pick up a box. The wheel fault is a partial fault where a robot is still able to aid in collective transport whereas the lifter fault is a critical fault where a robot may cause the team to become stuck [35]. We assume that it is equally likely for both faults to occur. We set a limit of 5 for the total number of faults that can occur in the swarm. Given the experimental setup and swarm controller, the performance of the swarm as a whole can therefore be seen to be dependent on the capacities associated to spatial and bodily awareness of individual robots.

Evaluating performance We select a single performance metric for evaluation: efficiency. In this case, we are interested in how quickly the swarm is able to retrieve and deliver boxes. We define the performance metric as a function of the number of boxes delivered over time. For each trial, we evaluate performance P as the integral under a step function $f(t)$, the number of boxes delivered at time $t \in [0, t_{max}]$. We have the trial duration $t_{max} = 2000s$ and the maximum possible number of boxes to deliver $B = 22$. We have $f(t) \in \{0, 1, ..., B\}$.

We are also interested in comparing performance scores across trial configurations for enabled capacities. For the performance score for each configuration, we compare this to the baseline performance of the swarm when there are no capacities enabled. In order to make such a comparison, we use the Mann-Whitney U test as a measure of *group difference*. For each trial we generate a sample of performance scores S. In particular we generate a sample for the baseline trial configuration S_B. Each of these samples S can then be compared to S_B to determine whether there is a large or small group difference. Group difference corresponds to difference in performance. A small group difference indicates that the scores in S are close to the baseline - there is little change in performance. A large difference however indicates a large performance change. This difference may be positive or negative.

More precisely, we have:

$$\Delta P = 2h\left|\frac{U}{n_1 n_2} - 0.5\right| \tag{11}$$

$$h = \begin{cases} 1, \text{if } R > R_B \\ 0, \text{if } R = R_B \\ -1, \text{otherwise} \end{cases} \tag{12}$$

where ΔP is the group difference (or change in performance) and U is the test statistic when comparing sample S to baseline S_B, and $n1, n2$ are the corresponding sample sizes. Variable h takes into account the directionality of group difference by comparing the rank totals for the two samples. Rank totals for each sample, R and R_B, are produced by ranking the sample data in aggregate. ΔP sits in the range of $[-1, 1]$.

3.4 Simulation results

Given the two capacities we have identified and their two levels of “skill”, we systematically enable each level of capacity independently and evaluate the resulting swarm performance for our selected metric. Additionally, we evaluate combinations of capacities and their levels which has equivalence to interactions between dimensions of awareness. The resulting performance is compared to the baseline performance where all capacities at all levels are disabled, using the measure of group difference ΔP as defined in equation 11.

Considering all possible combinations of independent variables, we have a total number of 9 trial configurations. For each trial configuration, a total of 100 trials are run where the initial

European Innovation Council Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

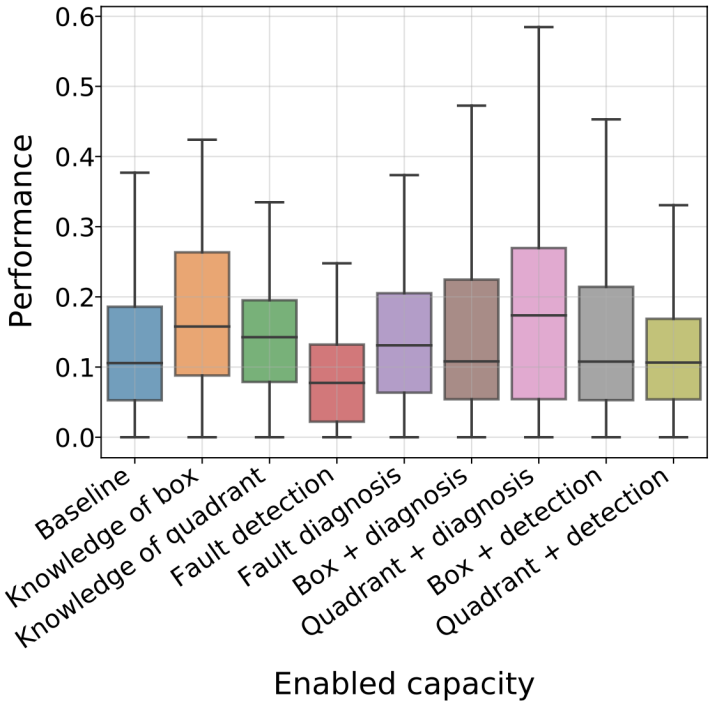

Figure 12: Performance scores for each configuration of capacities present in the swarm. The outliers are not plotted and scores are normalised for the range of minimum score (0) and the maximum score attained across all trials.

Table 4: Performance change with respect to baseline. An asterisk indicates a combination of capacities.

Enabled capacity	ΔP (2 d.p.)
Fault detection	-0.26
Box + detection*	-0.01
Quadrant + detection*	0.04
Fault diagnosis	0.09
Box + diagnosis*	0.12
Knowledge of quadrant	0.13
Knowledge of box	0.24
Quadrant + diagnosis*	0.27

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

positions of boxes and robots are randomised. The performance scores are normalised over a range of minimum and maximum performance. Minimum performance is assumed to be 0 corresponding to complete failure of the swarm to deliver any boxes. Maximum performance is selected as the maximum performance score from all trials.

Figure 12 shows the performance scores across the batch of trials for each configuration of enabled capacities. The distribution of scores shows high variance with left skew. High variance results from the stochasticity in the collective transport controller - box delivery is dependent on the speed of team formation which is highly variable. There is a noticeable difference in performance across enabled capacities however. Table 4 shows the performance change for each enabled configuration of capacities when compared to baseline performance, in ascending order. Notably, enabling the capacity for fault detection has a substantial negative impact on swarm performance compared to the baseline with no capacities enabled. Our approach allows for the possibility that a capacity may not necessarily have a positive change on performance.

When considering each dimension of awareness independently, the associated capacities with largest performance change are knowledge of box location (spatial awareness) and fault diagnosis (bodily awareness). This is an intuitive result since in both cases, these are the capacities with the highest level of information available in our experimental setup. Since the capacities associated with spatial awareness have larger performance change compared to those associated with bodily awareness, this suggests that for this particular task and experimental setup, spatial awareness may be more critical to efficiency of the swarm. Combinations of capacities generally have small performance change. However, the largest performance change is produced by enabling the combination of knowledge of the box quadrant and fault diagnosis. This result is somewhat surprising, given that knowledge of box location has the largest performance change for spatial awareness independently. This demonstrates that combinations of capacities may produce results which are more, or in some cases less, than the sum of their parts.

3.5 Discussion and conclusion

We have presented a practical application of a framework for evaluating the effect that awareness has, through associated capacities, on the performance of artificial agents in a given task. We have introduced terminology for the concepts related to awareness together with the relations between them, which form the structure of an implementation pipeline for application. In particular, we have demonstrated an example implementation for a robot swarm in an intralogistics scenario, examining two dimensions of awareness: spatial and bodily awareness. We have shown that our approach is able to provide insights into how various capacities (associated with the dimensions of interest) impact swarm performance for the chosen performance metric: efficiency of the swarm. We are also able to compare how the capacities associated with each dimension impact swarm performance and how to control awareness for desired performance. In the example implementation, the results suggest that capacities associated with spatial awareness have low or negative impact on swarm performance when considered independently, compared to the capacities associated with bodily awareness which have positive impact on performance.

Beyond implementation in a single system, the proposed framework is designed to enable comparisons to be made between various artificial systems, along various dimensions of awareness. The strength of the framework lies in its flexibility which is two-fold: first, the association of capacities to dimensions and the freedom to choose relevant tasks allows great flexibility in its application to various systems, making it possible to perform inter-system

European Innovation Council Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

comparisons. Secondly, as seen in the example implementation presented, there is flexibility in the comparisons one can make within a single system - we can choose to compare capacities along a single dimension, or even across multiple dimensions, or combinations of capacities equivalent to interactions of dimensions. The power therefore lies with the user to apply the framework according to the desired point of comparison. At its core, the framework draws relations and dependencies between the concepts defined, making the challenge of examining awareness tractable from a system designer's perspective.

Measuring awareness is critical to understand, and design for it. The different dimensions of awareness implemented in a system have the potential to improve the performance of the system. Another potential benefit would be improved interfacing with society by enabling increased control, and better understanding and evaluation with respect to ethical considerations. Without proper evaluations of the abilities of artificial systems we cannot rely upon them and collaborate with them in our daily lives ethically. In this work we present a framework to assess the performance gain provided by different dimensions of awareness, and apply the framework to an intralogistics use-case for a robot swarm. In the future we will apply this framework to both quantify awareness in artificial systems, and evolve solutions of systems based on awareness requirements. As a first step we provide an example implementation of the framework with a view to future implementation across artificial systems, and demonstrate the power of the framework in making tractable the challenge of examining awareness.

4 Control of emergence of awareness

EMERGE outlines a roadmap for designing collaborative systems that leverage emergent awareness for self-regulation and interoperability between individuals. We have demonstrated that equipping individuals with specific system capacities serves as a proxy for regulating and controlling their degree of awareness, expressed across a set of dimensions. Here, we define *control* as the ability of a designer—whether human or computational—to set preferences (either offline or online) that influence the collective behaviour and the potential for awareness of a distributed system. In this context, the strategic selection, control, and optimization of system capacities enable or constrain the space of behaviours the collective can exhibit. Therefore, we explore how automatic design methods—e.g., those based on artificial evolution and similar optimization processes—can drive this selection process to regulate and control the emergent awareness of a collective system.

We focus on controlling individuals' capacity to communicate. As outlined in WP1 of EMERGE, communication is the primary enabler of collective awareness. Individuals capable of selecting and sharing information about their state and environment contribute to this awareness. Communication is the first step toward higher degrees of collaborative awareness: coordination and cooperation. It facilitates the transition from coexisting individuals to systems where strategic information sharing triggers informed collective actions.

4.1 Automatic design of communication strategies in minimal agents

EMERGE considers a set of use cases (soft robots, robot swarms, and cobots) with which to investigate the regulation of emergent awareness capacities through the selection and control of communication mechanisms. We experiment with robot swarms, whose loosely coupled nature supports varying degrees of communication and allows for continuous reconfiguration at runtime—while still enabling diverse collective behaviours.

European Innovation Council Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

We focus on agents that, *a priori*, have no predefined shared language and have constrained individual capabilities. With this system, we explore how automatic design processes—the superclass for the application of evolutionary strategies for programming robots—can select appropriate communication mechanisms. In line with the EMERGE vision, our goal is not to develop a universal communication protocol for coordinating agent collectives. Instead, we aim to create automatic design methods capable of generating domain-specific communication strategies tailored to specific tasks.

The literature in swarm robotics extensively exemplifies the communication between minimal agents, in many cases drawing inspiration from mechanisms found in biological systems. Direct communication and stigmergy are two key mechanisms that enable coordination and cooperation in robot swarms. Direct communication requires robots to be within communication range to exchange information synchronously. Stigmergy, on the contrary, is an indirect form of communication in which robots modify their environment to leave information traces that others can interpret, without a need for synchrony.

Previous work has demonstrated that automatic design can generate coordination strategies using either direct communication [19] or stigmergy [53]. Here, we present results showing that an automatic method can also select between these two mechanisms within a single design process. Therefore, it is a suitable method to design domain-specific communication strategies.

We investigate scenarios in which a robot swarm can use environmental cues and inter-robot signalling to perform its mission more effectively. We conduct simulation experiments with a swarm of robots that address aggregation, homing, and task allocation problems—each requiring domain-specific communication strategies. Robots coordinate via stigmergy by releasing artificial pheromones into the environment or through direct communication by signalling with LEDs, both perceived by onboard cameras The automatic design process selects between these two mechanisms to generate collective behaviours for the task at hand.

The results show that the automatically designed robot swarms operate with communication strategies that leverage either mechanism in a mission-specific way. We observed that when appropriate environmental cues are missing, or when the swarm cannot effectively use them, the design process automatically compensates by relying on communication.

The automatic design of communication strategies can potentially facilitate interfacing robot swarms with external systems. We investigate the selection of communication mechanisms as a step toward enabling interoperability, coordination, and cooperation within the EMERGE heterogeneous robotics framework [14].

4.1.1 Direct communication

In swarm robotics, direct communication—where robots explicitly exchange signals—plays a crucial role in enabling coordination, cooperation, and self-regulation. A widely used mechanism to achieve direct communication with robot swarms is colour-based communication, where robots actively display and perceive colours to coordinate their actions.

Colour lights are commonly used to encode and transmit domain-specific information in swarm robotics. The nature of this information varies from one study to another and it often designed *ad hoc* to achieve particular behaviours. For example, researchers have designed swarms that connect locations of interest by establishing a chain of robots that act as waypoints for their peers [46]. Robots repeat a pattern of three colours along the chain to indicate the direction in which the peers should move. Using the same mechanism, but with a different purpose, researchers have also designed swarms in which robots self-assemble in mergeable structures by physically connecting to each other [40]. In their experiments, robots display

European Innovation Council Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

colours to indicate whether it is expected that other individuals join the structure and in which position they should do so.

More broadly, direct colour-based communication has been employed in various swarm behaviours, including fault detection [11, 40], transport [47, 15], human-swarm interaction [21, 50], clustering [1, 49], and foraging [5]. In these studies, communication strategies were manually designed based on domain-specific knowledge. To control the emergence of a given collective behaviour, *ad hoc* relationships were established between the colours a robot perceives and the corresponding behaviours it should adopt.

It has been demonstrated that automatic design processes can establish domain-specific communication protocols for robot swarms [19]. Given a set of possible communication channels (e.g., a range of colours to encode information), the automatic process simultaneously generates the control software required for robot operation and designs an appropriate communication strategy to be leveraged. This presents an opportunity to regulate emergent awareness through automatic design methods. The designer of system with collective awareness can enable or restrict communication capabilities—granting robots more or less capacity for information sharing—and, in doing so, directly influence the potential for emergent awareness within the collective. However, this constraint does not prevent the automatic design process from optimizing collective behaviour; rather, it defines the extent of awareness that can emerge based on the available communication mechanisms.

Direct communication via colour lights is a practical mechanism applicable in minimalistic physical robots. It is platform-independent, as most swarm robotics platforms include LEDs and cameras [45], and it facilitates the implementation of complex missions by allowing flexible environmental colour encoding [19, 8, 57]. While other modalities, such as infrared [24] or sound [59], have been explored, colour-based communication offer the distinct advantage of facilitating visualization and monitoring—allowing for human interpretability of the swarm behaviours [45, 33].

4.1.2 Indirect communication (Stigmergy)

Stigmergy is a coordination mechanism in which agents self-organize through indirect communication mediated by the environment [22, 26]. In swarm robotics, it enables asynchronous interactions, where robots modify their environment to leave cues that others can perceive and respond to. Similar to direct communication, regulating how and when information is encoded in the environment influences how robots coordinate, adapt, and respond to dynamic conditions. Within the EMERGE framework, stigmergy serves as a communication mechanism that can be operationalized to either enable or constrain a system's potential for emergent awareness by modulating the flow of information within the collective.

A common way to study stigmergy in swarm robotics is through pheromone-based behaviours, which mimic the communication strategies of social insects. For example, ants leave chemical trails—i.e., pheromones—that influence the behaviour of their peers. Similarly, robot swarms can use pheromone-based stigmergy if they are equipped with mechanisms to artificially release and perceive pheromones [54]. The automatic design of pheromone-based stigmergy has been recently demonstrated in robot swarms [53, 54]. This research features collectives of robots that project UV light onto a ground surface coated with photochromic material. When exposed to UV light, the material changes colour from white to magenta and gradually fades back within approximately 50 seconds, mimicking the evaporation of chemical pheromones.

Designing stigmergy-based behaviours remains a significant challenge [27]. Unlike direct communication, which allows robots to explicitly exchange signals (e.g., via color lights), stig-

European Innovation Council Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

mergy relies on environmental modifications that indirectly shape future interactions. This makes the design process less intuitive and more dependent on emergent dynamics. The design pheromone-based communication requires careful tuning of pheromone deposition, evaporation rates, and robot responses to environmental cues. As with direct communication strategies, most implementations of pheromone-based stigmergy rely on manual design tailored to specific missions. This also presents an opportunity to leverage stigmergy as a communication mechanism that can be regulated through automatic design processes to enable or constrain the potential for emergent awareness.

4.1.3 Automatic Modular Design (AutoMoDe)

AutoMoDe, short for Automatic Modular Design [17], is an approach to the design of control software for robot swarms. AutoMoDe methods produce control software for robots by tuning and assembling predefined software modules into a modular control architecture. This process is driven by an optimization algorithm that uses mission-specific performance metrics to search for suitable control software for the robots.

AutoMoDe is a general approach to designing collective behaviours for robot swarms that is characterized by four key components. First, the class of problems that the method is intended to address, including general mission requirements and common environmental features across missions. Second, the robot platform for which the control software will be designed, comprising both the hardware of the robot and the control interface available to operate it. Third, the control architecture used to produce the control software, which considers both the predefined software modules and the architecture into which they can be assembled. Fourth, the optimization strategy that drives the design process by selecting and refining control software instances with respect to a mission-specific performance measure.

We use AutoMoDe to investigate how an optimization-driven design process can select communication mechanisms for robot swarms in a domain-specific way. As discussed in previous sections, the choice between communication mechanisms serves as a means to control the potential for emergent awareness within the system—this is our class of problems. Specifically, we explore AutoMoDe's ability to leverage direct communication or stigmergy to design good performing control software for the robots, leveraging the capacities that each mechanism can enable. To evaluate this automatic design process, we conceived simulation experiments where both direct communication and stigmergy are viable solutions, yet their selection and implementation directly influence overall system performance. In these experiments, AutoMoDe is used to simultaneously (i) design the collective behaviours required for a given mission, (ii) select the most effective communication mechanism, and (iii) develop a communication protocol that drives robot interactions. To implement this approach, we adapt two methods from the AutoMoDe family, each designed for missions requiring specific communication strategies: AutoMoDe-`TuttiFrutti` [19], which targets direct communication, and AutoMoDe-`Habanero` [53], which focuses on stigmergy. We integrate the algorithms of both methods into a single process capable of selecting between the two, allowing AutoMoDe to autonomously operationalize the most suitable communication mechanism for a given task.

In the following sections, we describe the robot platform used in the experiments and its capabilities, the modular control software on which AutoMoDe operates, and the design process for selecting communication mechanisms and developing the collective behaviour of the robots.

Robot platform and simulator

We consider a simulated version of the e-puck robot [43, 18]—see Figure 13. Simulations are performed in ARGoS3 [48], version beta 48, together with the argos3-epuck library [18].

European Innovation Council Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

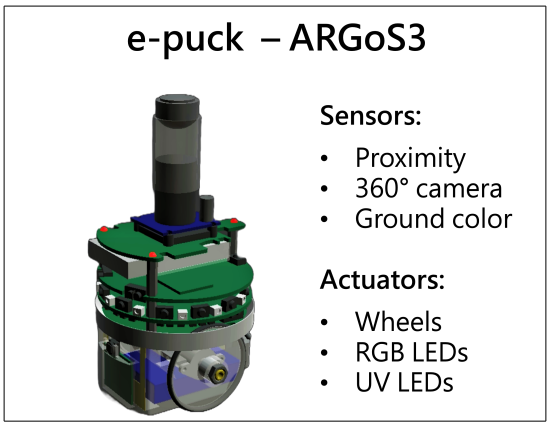

Figure 13: The e-puck robot considered in the experiments, along with its sensors and actuators. The picture shows the simulated version of the physical robot.

The sensors and actuators of the robot, along with their corresponding inputs and outputs in the control software, are listed in Table 5. The e-puck has eight proximity sensors $(prox_i)$ distributed around its chassis to detect nearby obstacles. Three infrared ground sensors (gnd_j) allow it to differentiate between black, gray, and white surfaces. An omnidirectional vision turret detects red, green, cyan, magenta signals (cam_c) within a $360°$ field of view and a 0.5 m range. The camera's field of view (fov) can be adjusted to narrow $(\frac{1}{12}\pi)$ or omnidirectional (2π) perception. For each detected colour, a unit vector (V_c) represents the relative aggregate position of robots or objects displaying that colour. The control software adjusts wheel velocities (v_k) between -0.12 and 0.12 m/s. The RGB LEDs on top of the e-puck can display cyan or remain off. For stigmergy-based communication, the e-puck is equipped with hardware to lay and detect artificial pheromones. The control software controls UV LEDs (phe) on the lower body to project magenta pheromone trails $(trail)$, or none $(\varnothing)$.

Modular control architecture

AutoMoDe assembles predefined software modules into probabilistic finite-state machines. The set of modules in `TuttiFrutti` comprise six low-level behaviours—the actions that a robot can take, and seven transition conditions—the events that trigger the transition between low-level behaviours. These modules provide the e-puck different ways of interacting with robots and objects that display colours and with the pheromone trails. Table 6 lists AutoMoDe's low-level behaviours and transition conditions.

Low-level behaviours: `TuttiFrutti` and `Habanero` low-level behaviours are identical in all aspects except for the mechanism they use to enable communication between robots. In EXPLORATION, the robot moves straight until it detects an obstacle in front with its proximity sensors $(prox_i)$. It then rotates for a number of control cycles defined by the integer parameter τ, where $\tau \in \{0, \ldots, 100\}$. STOP sets the robot to a standstill behaviour. COLOR-FOLLOWING and COLOR-ELUSION control movement based on visual input. In COLOR-FOLLOWING, the robot moves toward (V_c). In COLOR-ELUSION, it moves away from $(-V_c)$. Signals from other robots are associated to the colours $\delta \in \{C, M\}$—cyan on RGB LEDs for direct communication and magenta in pheromone trails for stigmergy. Other objects in the environment, such as the walls of the experimental arena, can display the same colours along with two additional ones, $\delta \in \{R, G, C, M\}$. ATTRACTION, REPULSION, COLOR-FOLLOWING, and COLOR-ELUSION incorporate physics-based obstacle avoidance [4]. In `TuttiFrutti`'s low-level behaviours, the parameter $\gamma \in \{\varnothing, C\}$ determines whether the RGB LEDs display cyan as a signalling mechanism. In `Habanero`'s low-level behaviours, the parameter $phe \in \{\varnothing, trail\}$ determines whether the UV LEDs activate to leave a magenta trail as a signalling mechanism. The

European Innovation Council Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

Table 5: The control interface for the e-puck. The robot can perceive: red (R); green (G); cyan (C); and magenta (M). The robot can display no colour $(\varnothing)$ or cyan (C) with its top RGB LEDs and can also set its lower-body UV LEDs to project: no pheromone trail $(\varnothing)$; or a magenta one $(trail)$. V_c is calculated by aggregating the positions of perceived colour signals into a unique vector.

Input	**Value**	**Description**
$prox_{i\in\{1,\dots,8\}}$	$[0,1]$	reading of proximity sensor i
$gnd_{j\in\{1,\dots,3\}}$	$\{black, gray, white\}$	reading of ground sensor j
fov	$\{\frac{1}{12}\pi, 2\pi\}$	camera field of view
$cam_{c\in\{R,G,C,M\}}$	$\{yes, no\}$	colours perceived
$V_{c\in\{R,G,C,M\}}$	$(1.0; [0,2]\,\pi\,\mathrm{rad})$	their relative aggregate direction
Output	**Value**	**Description**
$v_{k\in\{l,r\}}$	$[-0.12, 0.12]\,\mathrm{m\,s^{-1}}$	target linear wheel velocity
$LEDs$	$\{\varnothing, C\}$	colour displayed by the LEDs
phe	$\{\varnothing, trail\}$	projection of UV lights

Period of the control cycle: $0.1\,\mathrm{s}$.

parameters τ, δ, γ, and phe are tuned by the automatic design process.

Transition conditions: BLACK-FLOOR, GRAY-FLOOR, and WHITE-FLOOR trigger a transition when the robot steps on a floor region (gnd_j) that is black, gray, or white, respectively. The probability of transitioning is determined by the parameter $\beta \in [0,1]$. FIXED-PROBABILITY triggers a transition with a fixed probability $\beta \in [0,1]$, without any additional conditions. COLOR-DETECTION is based on the colours perceived by the robot (cam_c). A transition occurs with probability $\beta \in [0,1]$, triggered by a specific colour $\delta \in \{R, G, C, M\}$. Robots in the swarm can activate signalling mechanisms with $\delta \in \{C, M\}$—cyan on RGB LEDs for direct communication and magenta in pheromone trails for stigmergy. Other objects in the environment, such as the walls of the experimental arena, can display the same colours along with two additional ones, $\delta \in \{R, G, C, M\}$. The parameters β and δ are tuned by the automatic design process.

Selection of communication mechanisms via automatic design

As mentioned in previous sections, our aim is to use AutoMoDe to simultaneously (i) design the collective behaviours required for a given mission, (ii) select the most effective communication mechanism, and (iii) develop a mission-specific communication protocol that drives robot interactions.

AutoMoDe conducts an optimization process to identify a combination of modules and parameters that maximize swarm performance based on a mission-specific metric. The control software is structured as a probabilistic finite-state machine, limited to four states (low-level behaviours) and four outgoing transitions per state. Transitions always occur between different states, and self-transitions are not permitted.

The modules and their parameters are selected off-line through an optimization process conducted with Iterated F-race [39]. Iterated F-race is an optimization algorithm that explores the design space to find control software configurations suited for the mission at hand. The

European Innovation Council
Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

Table 6: AutoMoDe's software modules. The modules are defined according the control interface of the e-puck robot e-puck robot, see Table 5. `TuttiFrutti` 's low-level behaviours enable direct communication using RGB LEDs, where robots perceive each other through the cyan light emitted by their LEDs. `Habanero` 's low-level behaviours use stigmergy, where robots do not perceive each other directly but instead react to the magenta pheromone trails they lay. In both cases, the distinction lies solely in the colour to which robots respond; there is no algorithmic difference in the software modules. The transition conditions remain the same for both communication mechanisms.

TuttiFrutti's behaviour*	**Parameter**	**Description**
EXPLORATION	$\{\tau, \gamma\}$	movement by random walk
STOP	$\{\gamma\}$	standstill state
COLOR-FOLLOWING	$\{\delta, \gamma, fov\}$	steady movement towards robots/objects of color δ
COLOR-ELUSION	$\{\delta, \gamma, fov\}$	steady movement away from robots/objects of color δ
WAGGLE	$\{\gamma\}$	in-place waggle motion
Habanero's behaviour**	**Parameter**	**Description**
EXPLORATION	$\{\tau, phe\}$	movement by random walk
STOP	$\{phe\}$	standstill state
COLOR-FOLLOWING	$\{\delta, phe, fov\}$	steady movement towards trails/objects of color δ
COLOR-ELUSION	$\{\delta, phe, fov\}$	steady movement away from trails/objects of color δ
WAGGLE	$\{phe\}$	in-place waggle motion
Transition condition	**Parameter**	**Description**
BLACK-FLOOR	$\{\beta\}$	black floor beneath the robot
GRAY-FLOOR	$\{\beta\}$	gray floor beneath the robot
WHITE-FLOOR	$\{\beta\}$	white floor beneath the robot
FIXED-PROBABILITY	$\{\beta\}$	transition with a fixed probability
COLOR-DETECTION	$\{\delta, fov, \beta\}$	robots/objects/trails of color δ perceived
TRAIL-DETECTION	$\{fov, \beta\}$	pheromone trail perceived

* `TuttiFrutti`'s low-level behaviors can set the RGB LEDs ($\gamma \in \{\varnothing, C\}$) alongside the action described.

** `Habanero`'s low-level behaviors can set the UV LEDs ($phe \in \{\varnothing, trail\}$) alongside the action described.

Modules that use the camera can set its field of view (fov) alongside the action described.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

performance of the configurations is estimated through simulations performed in ARGoS3 [48], version beta 48, together with the argos3-epuck library [18]. The duration of the optimization process is determined by a predefined simulations budget. Once the budget is exhausted, the design process ends and AutoMoDe returns the best configuration found.

In typical AutoMoDe setups, `TuttiFrutti` and `Habanero` are used independently, each targeting a specific class of problems. `TuttiFrutti` is applied to missions that require direct communication, while `Habanero` is used for missions that rely on stigmergy. However, within the EMERGE framework, we propose a setup where AutoMoDe autonomously selects the appropriate communication mechanism rather than relying on predefined choices. Instead of prescribing *a priori* whether direct communication or stigmergy is needed, AutoMoDe identifies the most suitable mechanism for the mission at hand.

To achieve our goal, we structure the design space so that the optimization process converges to combinations of modules from either `TuttiFrutti` or `Habanero`. This is implemented by partitioning and constraining the parameter space with dependent parameters—see [39] for details. In this design process, Iterated F-race operates over a super-parameter that determines whether direct communication or stigmergy will be used. Based on this choice, it then enables and tunes the corresponding set of software modules to design the collective behaviour of the robots. This selection process is what allows AutoMoDe to control how robots communicate to complete their mission.

Rather than relying on human expertise to define the communication-related potential for emergent awareness, we leverage the systematic and performance-driven nature of AutoMoDe's optimization process. AutoMoDe is free to select a communication mechanism and devise an effective way to use it based on the mission requirements. Specifically, it has freedom to automatically generate a domain-specific communication protocol for the robots.

4.2 Experiments

We conduct experiments with e-puck robots that must perform missions in which environmental signals, expressed as colours in the walls of the arena, provide relevant information to the swarm. We consider three missions: HOMING, AGGREGATION, and TASKING. For each of them, we propose two scenarios—each posing challenges on the way use and share environmental information to perform the mission. The time available to the robots to perform a mission is always $T = 120\,\mathrm{s}$. The performance of the swarm is evaluated according to a mission-specific objective function, which we keep unmodified for their corresponding two scenarios.

The three missions are adaptations we make from equivalent missions proposed in the literature [19, 54, 53] to study AutoMoDe. We select these missions because we conjecture that, to successfully perform them, AutoMoDe operationalise the two available mechanisms in different ways. We challenge the design process to design robot swarms that operate under varying persistence of environmental information, robot availability for information sharing, and interference in communication channels.

Protocol: For each mission and scenario, we conduct 10 independent design processes with AutoMoDe, resulting in 60 instances of control software—10 per scenario and mission. AutoMoDe is allocated a budget of 100 000 simulation runs to generate each control software instance. The design process and assessment are performed in ARGoS3. To evaluate the effectiveness of the produced control software, each instance is tested 10 times in simulation using different random seeds, which define a random initial positioning for the robots at the start of the mission. We report the assessment with box-plots that show the empirical distribution of performance obtained for each instance of control software produced in the experiments. For

European Innovation Council Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

each instance, we identify whether AutoMoDe selected direct communication or stigmergy as the mechanism to enable the coordination between robots.

In the following, we provide the specifications for each mission and comment on the experimental results.

4.2.1 Persistency of information

HOMING mission: The swarm comprises 16 robots. The robots must reach the red corner of the arena and remain there until the mission ends. The homing region is the square portion of the arena enclosed by red walls. The swarm's score is based on the total time robots spend inside this region, given by the function $f_{\text{AG}} = \sum_{t=1}^{T} \sum_{i=1}^{N} I_i(t)$, which must be maximized. Here, N is the number of robots, and T is the mission duration. The indicator function $I_i(t)$ equals 1 if robot i is inside the red corner at time t, and 0 otherwise. The performance is measured at every simulation step ($\Delta t = 0.1\,\text{s}$).

In Scenario 1, the environmental signal (i.e., the colour of the corner) remains visible for the entire mission (100% of the time). In Scenario 2, the environmental signal is visible for only 20% of the mission time, after which the red LEDs are switched off. The two scenarios differ in the persistence of information availability. In Scenario 2, robots must quickly locate and propagate information about the homing region before it disappears. This creates a difference in the urgency with which environmental information must be detected and shared within the swarm.

Figure 14 shows the arena for HOMING and the experimental results.

Results: AutoMoDe successfully designed control software that enabled the robots to reach and remain in the red corner of the arena. The selection of the communication mechanism and the strategy used to complete the mission varied between the two scenarios. This indicates that AutoMoDe designed collective behaviours that leveraged communication in a domain-specific way. In Scenario 1, AutoMoDe converged to solutions that relied on direct communication in 7 out of 10 runs. In Scenario 2, AutoMoDe instead generated solutions based on stigmergy in 7 out of 10 runs. The performance range remained similar in both cases. This suggests that the optimization process effectively adapted to different communication strategies, finding suitable solutions for each scenario without sacrificing performance.

In Scenario 1, the most common strategy is direct communication. A robot that finds the red corner moves toward it while activating its cyan LEDs to signal its state. Other robots, even if they do not see the red corner, follow the cyan signal toward it. Since the red lights persist throughout the mission, robots can rely on this environmental cue to stay in place.

In Scenario 2, the red signal disappears after 20% of the mission duration. Direct communication alone is insufficient, as robots cannot rely on continuous environmental cues to remain in place. Instead, the most common strategy is stigmergy. A robot that reaches the red corner drops magenta pheromones, attracting others to the same location. When the red lights disappear, the pheromone trail remains, allowing the swarm to stay in the designated region.

4.2.2 Density of information sharing

TASKING mission: The robots must service 16 workstations located in the arena. Workstations are indicated by gray floor patches and wall segments displaying green. A task in a workstation is completed when a robot enters a workstation and remains there for 4 seconds. The swarm's

European Innovation Council
Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

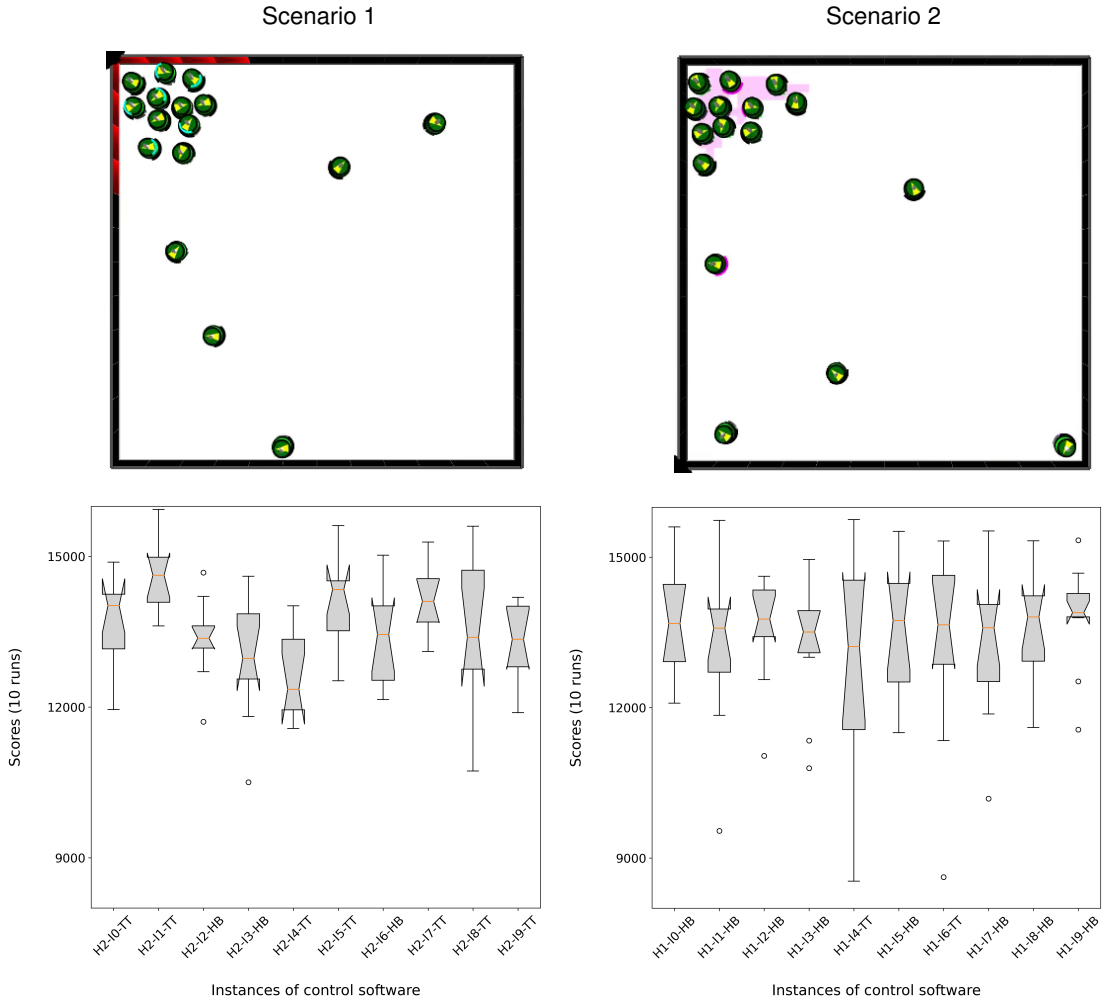

Figure 14: Experiments on persistency of information. Demonstrative swarm behaviours and experimental results for the mission HOMING. Columns suffixed ‘TT’ denote controllers using direct communication, and those suffixed ‘HB’ stigmergic communication.

score is based on the number of unique workstations serviced, with a penalty for repeated visits. The score is given by the function $f_{\text{TS}} = w(T) - w_{rep}(T)$, which must be maximized. Here, T represents the mission duration, $w(T)$ is the number of distinct workstations serviced, and w_{rep} is the number of repeated services.

In Scenario 1, the swarm consists of 16 robots, which must service the 16 workstations. In Scenario 2, the swarm consists of 3 robots, servicing the same 16 workstations. AutoMoDe must design a collective behaviour that maximizes the service capability, adapting in a domain-specific way to the constraints imposed by swarm size. The robot density affects how information is shared, introducing different communication requirements. In sparse conditions, robots might need to rely more on long-term information propagation, while in denser settings, local communication could suffice.

Figure 15 shows the arena for TASKING and the experimental results.

Results: AutoMoDe successfully designed control software that enabled the robots to collectively service the workstations. As in the first mission, the selection of the communication mechanism and the strategy used to complete the task varied between the two scenarios. This demonstrates that AutoMoDe leveraged communication in a domain-specific way. In

European Innovation Council
Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

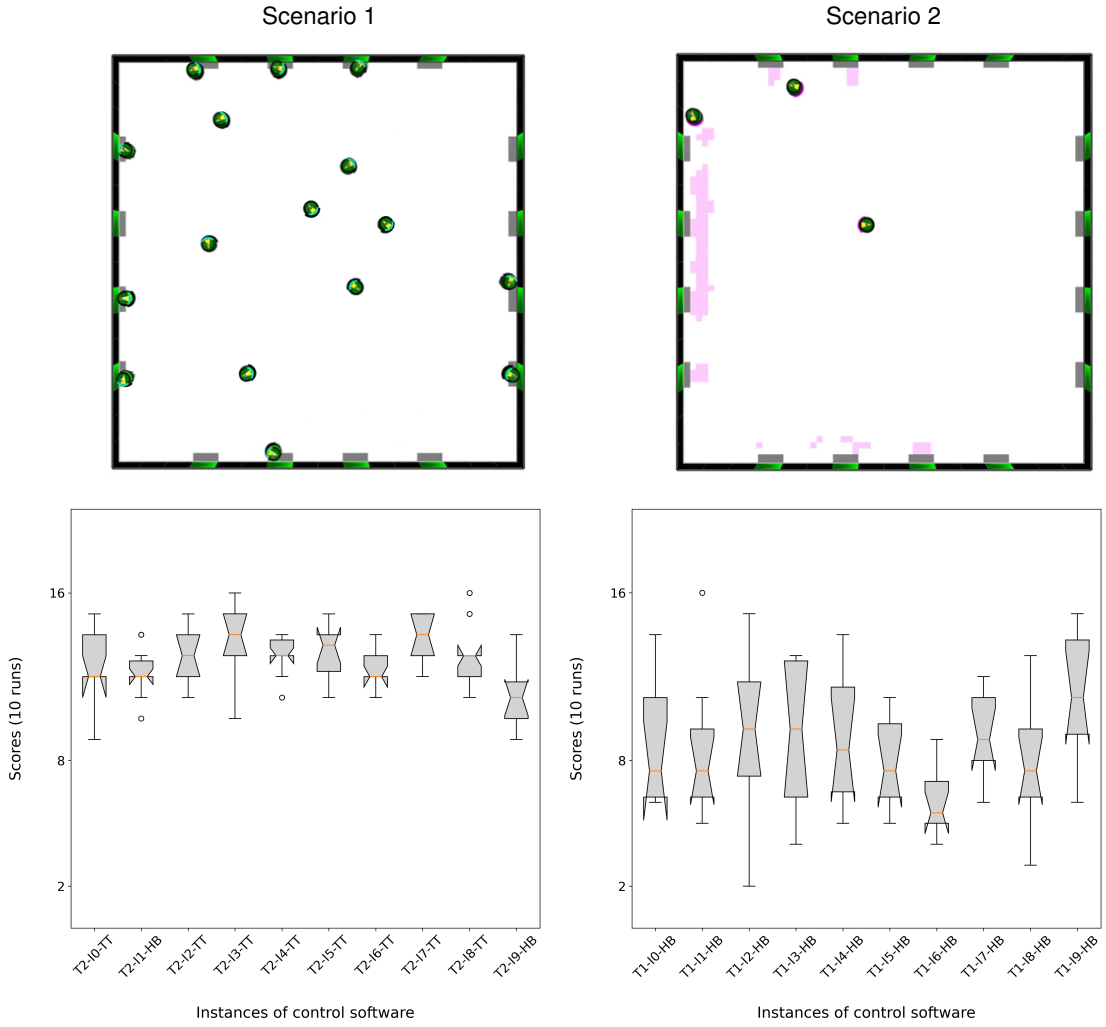

Figure 15: Experiments on the density of information sharing. Demonstrative swarm behaviours and experimental results for the mission TASKING. Columns suffixed 'TT' denote controllers using direct communication, and those suffixed 'HB' stigmergic communication.

Scenario 1, AutoMoDe converged to solutions relying on direct communication in 8 out of 10 runs. In Scenario 2, it exclusively generated solutions based on stigmergy in 10 out of 10 runs. Unlike the first mission, here we observe a more evident pattern/advantage in the selection of a specific communication mechanism for each scenario. The performance range differed, which was expected due to the change in the number of robots from 16 to 3. Notably, in Scenario 2, the swarm of three robots achieved scores close to the maximum (16 workstations serviced), showing that the optimization process effectively leveraged stigmergy to compensate for the low robot count. In Scenario 1, even with 16 robots, the swarm did not always succeed in servicing all 16 workstations, suggesting that task redundancy and congestion limited performance despite the larger team size.

In Scenario 1, the most common strategy is direct communication. Robots approach the workstations by following the green lights. Once they arrive, they activate their cyan LEDs. Passing robots detect the cyan signal and are repelled, preventing them from entering an already serviced workstation. This allows the 16 robots to ensure that each workstation is serviced only once. Robots that have already occupied a workstation remain there, acting as memory units to indicate completion. In this scenario, direct communication is sufficient, as the swarm can rely on physical presence to track serviced workstations.

European Innovation Council
Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

In Scenario 2, fewer robots are available, making it impossible to station one at each workstation. As a result, direct communication alone is insufficient. Instead, the most common strategy is stigmergy. Robots still approach workstations by following the green walls but do not remain there. Instead, they lay pheromone trails to indicate that a workstation has been serviced, then return to search for others. The pheromone signal persists even after the robot has left, allowing the swarm to track serviced workstations without requiring a robot to stay in place. This strategy enables the sparse swarm to efficiently coordinate task allocation through environmental memory.

4.2.3 Interference in communication channels

AGGREGATION mission: The swarm comprises 16 robots. The robots must approach one another, form a cluster, and remain close until the mission ends. At the start of each run, they are randomly positioned in the arena. The swarm's score is calculated as the average distance between robots, given by the function $f_{\text{AG}} = \sum_{t=1}^{T} d_{\text{avg}}(t)$, which must be minimized. At each time step t, the average inter-robot distance d_{avg} is added to f_{AG}. The performance is measured at every simulation step ($\Delta t = 0.1\,\text{s}$).

In Scenario 1, the swarm has available a single environmental signal—a cyan corner—indicating a possible aggregation site in the arena, which robots may choose to disregard. In Scenario 2, the swarm has available two environmental signals—two cyan corners, each suggesting a different, distant aggregation site. Robots can also choose to ignore these signals. The walls in the corners display cyan, the same colour used by robots for direct communication. This introduces interference between the signal provided by the environment and the signal related to one of the possible communication channels of the robots.

Figure 16 shows the arena for AGGREGATION and the experimental results.

Results: AutoMoDe successfully designed control software that enabled the robots to aggregate in the arena. As in the other two missions, the selection of the communication mechanism and strategy used to complete the task varied between scenarios. AutoMoDe leveraged communication in a domain-specific way, adapting to each environment. In Scenario 1, AutoMoDe converged to direct communication in 10 out of 10 runs. In Scenario 2, it exclusively generated stigmergy-based solutions in 10 out of 10 runs. There is therefore a clear pattern in the selection of communication mechanisms to perform each scenario. The performance range differed between the two cases.

In Scenario 1, the common strategy is direct communication. Robots activate their cyan LEDs to signal their presence and follow the same signal to move toward each other. The environment also displays a cyan signal, which could interfere with communication by making robots mistake arena walls for other robots. However, since the cyan signal is in a single location, it does not disrupt behaviour. Instead, it serves as a reference point for aggregation, allowing the swarm to converge quickly without negative effects on communication.

In Scenario 2, using the same strategy risks swarm fragmentation. The cyan environmental signal appears in two locations, and robots that mistake the walls for other robots may aggregate in separate areas, splitting the swarm. This is a negative effect caused by the interference in its communication channel. To counteract interference, the optimization process converges on a stigmergy-based strategy. Robots use pheromones to establish a single aggregation point and ignore the cyan environmental cues. This approach is slower and less precise but keeps the swarm together and prevents fragmentation, even at the cost of lower overall performance.

European Innovation Council
Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

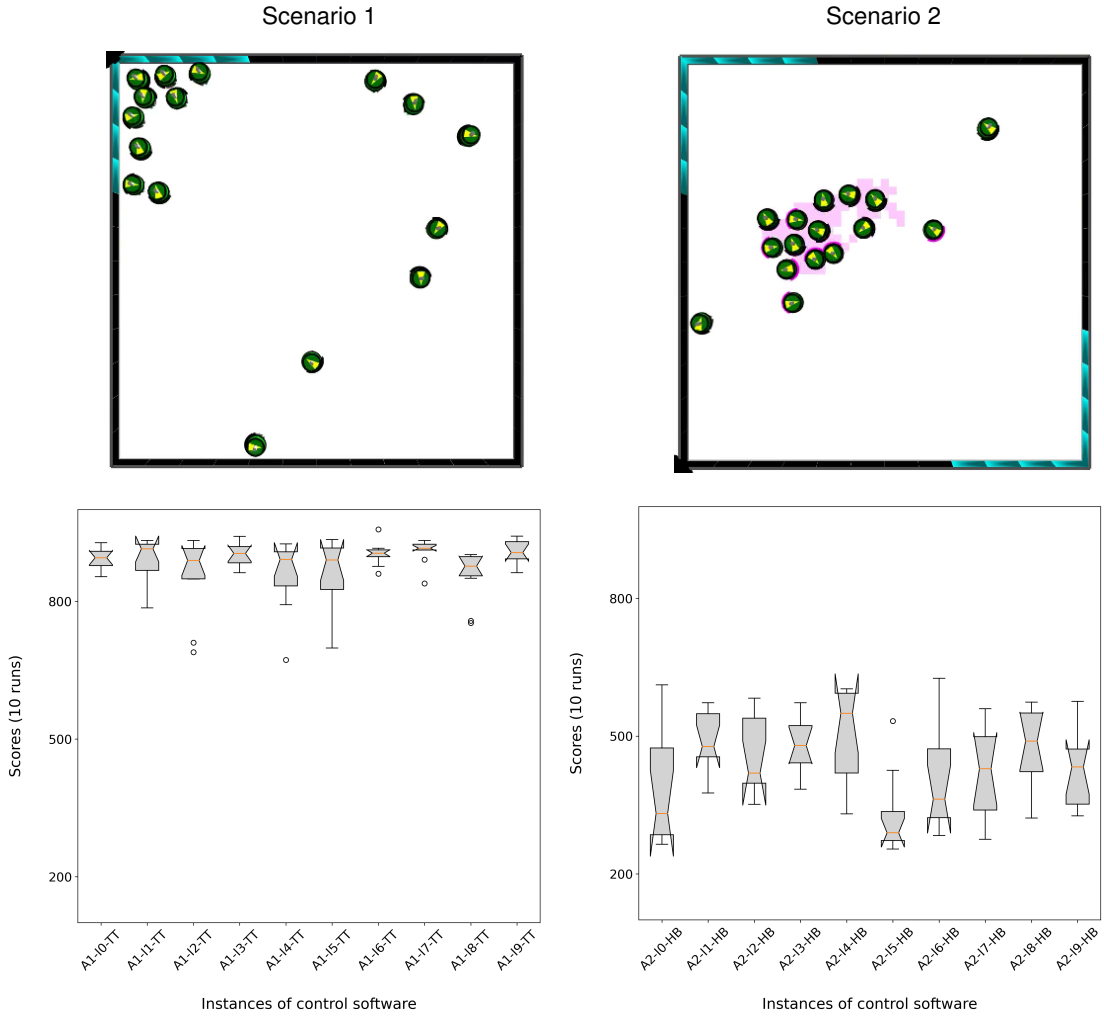

Figure 16: Experiments on interference in communication channels. Demonstrative swarm behaviours and experimental results for the mission AGGREGATION. Columns suffixed 'TT' denote controllers using direct communication, and those suffixed 'HB' stigmergic communication.

4.3 Discussion and conclusion

Experimental results demonstrate that automatic design is an effective approach for the selection, control, and optimization of communication mechanisms in robot swarms. In each mission and experimental scenario, AutoMoDe successfully identified suitable communication protocols, enabling interaction through direct communication and stigmergy. The signaling mechanisms available in AutoMoDe's modules—activating or deactivating the RGB LEDs or UV LEDs—had no predefined meaning for the missions. Despite this, AutoMoDe discovered domain-specific protocols, assigning mission-specific semantics to these mechanisms. For example, in AGGREGATION, pheromone trails signalled "come here", while in TASKING, they indicated "avoid this workstation". The results highlight the potential of optimization methods, such as the one used here or artificial evolution, in exploring and tailoring the capacities of minimal individuals. The automatic nature of the process allows for the discovery of how these mechanisms can be exploited, enabling emergent self-regulation and interoperability among robots.

In our experiments, a designer—a human supported by a computational optimization process—set preferences that ultimately controlled/regulated how awareness would resolve in the robot

European Innovation Council
Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

swarm. These preferences included a mission-specific performance metric and a set of communication mechanisms available to the robots. The strategic selection, control, and optimization of mechanisms—such as those described in WP1—showed to be suitable to enable or constrain the capacities of the collective. For example, if stigmergy had been manually disabled in the design process, the swarm would have lost its ability to retain memory and remain collectively aware of serviced workstations, particularly when operating with few individuals. In this sense, choosing which mechanisms are available during design enables or disables specific capacities in the swarm. This, in turn, regulates its potential for emergent awareness. While manual selection based on external expertise can achieve this control, it limits the ability to fully exploit awareness for optimized system performance. Bias and misconceptions may also be introduced if certain mechanisms or capacities are assumed to be favourable for a given mission—as seen in the fault detection experiments—potentially overlooking alternative, more effective solutions. Instead, within the EMERGE framework, we advocate for an automated selection process. By allowing the design process to autonomously choose the system configuration, we ensure that awareness capacities are not preconditioned but rather optimized for performance based on the task at hand.

These experiments represent a step forward toward achieving higher levels of collaborative awareness: coordination and cooperation. We have shown how automatic design can apply domain-specific selection mechanisms to determine how individuals share information about their state and environment. The automatic design of signalling protocols and communication strategies facilitates the transition from coexisting individuals with minimal interaction to more structured collective systems. By enabling mission-specific information sharing, minimal collectives can display informed collective actions.

The automatic design of domain-specific communication mechanisms can enable interoperability, coordination, and cooperation between heterogeneous systems that initially do not share a common communication protocol. By automatically designing both the collective behaviour of robots and their communication strategies with other agents in the environment, this approach could also facilitate the integration of robot swarms with other collective systems in the EMERGE portfolio (i.e., neural networks, soft robots).

WP3 and WP4 will provide an archetypal representation that can be used to model communication strategies and to gain insights into how internal information flow drives the behavior of distributed systems. It is reasonable to assume that equivalent mechanisms for both direct communication and stigmergy-based communication can be identified in the interactions within neural networks and soft robots. By adopting a shared language to represent these mechanisms—the archetypes, we expect it will be possible to articulate communication strategies across heterogeneous collective systems, even if the technologies (mechanisms) enabling the required capacities are substantially different.

5 Conclusion and future plans

In this deliverable we design new forms of awareness, measure and control their contribution to system performance, and finally automatically control agent awareness and behaviour.

5.1 Evolving exploitation of awareness

One of the most interesting aspects to the ability to give agent collectives a new sense of spatial awareness, DSA, is the possibility that this opens up for new algorithms. The two

European Innovation Council
Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

examples described in Section 2, shape formation and intralogistics, are simple hand-crafted demonstrations. However, in the field of swarm robotics, a common approach to algorithm or controller design is the use of evolutionary algorithms or other automatic design methods to discover controllers for individual members of a swarm such that a desired collective behaviour, encoded as a fitness function, emerges as a result of the interactions between the robots, and of the robots with their local environment. Evolutionary methods often discover unexpected or surprising solutions that a human designer may not conceive. Making the DSA ability available to be manipulated and used by the evolutionary process is an exciting avenue to pursue. The path toward this that we envision uses the evolution of Behaviour Trees as robot controllers, [31], exposing the DSA ability as custom leaf nodes. Behaviour Trees are desirable in this case as a controller architecture due to their hierarchical structure, enabling decompositional analysis for understanding and safety purposes. The granularity of representation of this information to maximise novelty and interesting exploration is an open question.

As well as this, the process underlying DSA, Gaussian Belief Propagation (GBP), has some similarities to other swarm consensus algorithms such as bee nest site selection [56]; where beliefs are advertised to nearby agents, and these beliefs ”compete” by very simple local processes until the swarm converges on a single best nest site. Best-of-n, [61], is widely examined as a swarm consensus problem, but there are significant differences to GBP, which deals with continuous quantities. We anticipate that a possible resolution may involve the use of ring attractor networks of spin variables and similar neural-like structures that have bioplausibility [58]. These can represent various types of continuous variables and have been shown to perform something similar to Bayesian inference with noisy inputs [34]. By carefully constraining allowable neural geometries and message passing primitives, we hope to evolve solutions to achieving consensus on spatial relationships within a swarm.

Thus we plan to work on two levels; evolving higher level Behaviour Tree controllers that make use of the global spatial information of DSA to achieve task performance in novel ways, and at the lower level, to discover new, potentially low cost, bioinspired consensus algorithms for spatial awareness.

5.2 Illuminating the awareness space for heterogeneous collectives

We have demonstrated that automatic design is a promising approach for discovering and controlling emergent awareness in minimal collectives. In the next step, we will expand the design space to strengthen its connection with the dimensions defined in WP1. We aim to simultaneously explore multiple dimensions of awareness and incorporate agent heterogeneity into the system. By doing so, we will illuminate the possible resolutions for awareness when different dimensions (and capacities) are concurrently considered in the design process.

Our work so far has focused on enabling communication as a first step toward higher degrees of awareness. However, we have yet to fully explore mechanisms and capacities related to spatial, temporal, body, and metacognition dimensions of awareness. We plan to extend our research in these directions. We expect that the automatic design approach used here, or similar ones based on evolutionary optimization, can naturally expand to these dimensions. Just as AutoMoDe selected behavioural modules based on direct communication and stigmergy, it could also select mechanisms that provide awareness capacities in space, time, body, and metacognition. For example, building on the work conducted in DSA, AutoMoDe could design behaviours using either traditional local perception or the enhanced awareness provided by DSA. In all cases, selection will be driven by performance optimization, allowing AutoMoDe to balance trade-offs in a domain-specific way. Expanding the optimization-based design process to incorporate these dimensions presents challenges. To address this, we plan to explore

European Innovation Council Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

alternative approaches similar to AutoMoDe that can efficiently partition the design space. Potential strategies include evolutionary optimization methods for *illumination processes* based on quality diversity, such as novelty search [37] and MAP-Elites [44]. We expect that these illumination methods will also support efforts in WP3 and WP4 to explore suitable archetype networks for representing interaction dynamics in minimal collectives. In a sense, discovering how combinations of capacities across dimensions result in different resolutions of awareness is akin to discovering configurations of archetype units and connectors that can capture the dynamics of those resolutions.

In the long term, we plan to apply our research to heterogeneous collectives composed of agents with different capacities. Our work has demonstrated that automatic design can generate communication strategies for groups of robots. This capability has also been shown in limited semi-heterogeneous cases [20]. We aim to extend our work to enable interaction between agents with substantial different capabilities—tentatively, in transport, manipulation, and communication. We anticipate that capacities for body and metacognition awareness will play a crucial role in these systems. The automatic design process will have to explore combinations of mechanisms that allow robots to recognize their own capabilities, share this information with the collective, and engage in tasks suited to their abilities. As in our current experiments, the proper selection of communication mechanisms will be essential for effective interaction between agents. We expect a noticeable challenge on the concurrent automatic design of control software across agents with different abilities—the size and complexity of the design space is significantly larger than the one we considered so far. We expect that partitioning the design space—also using evolutionary algorithms based on novelty search and MAP-Elites—will help address this issue.

European Innovation Council
Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

References

[1] Michael Allwright, Navneet Bhalla, Haitham El-faham, Anthony Antoun, Carlo Pinciroli, and Marco Dorigo. SRoCS: leveraging stigmergy on a multi-robot construction platform for unknown environments. In Marco Dorigo, Mauro Birattari, Simon Garnier, Heiko Hamann, Marco Montes de Oca, Christine Solnon, and Thomas Stützle, editors, *Swarm Intelligence: 9th International Conference, ANTS 2014*, volume 8667 of *Lecture Notes in Computer Science*, pages 158–169, Cham, Switzerland, 2014. Springer.

[2] Mauro Birattari, Antoine Ligot, Darko Bozhinoski, Manuele Brambilla, Gianpiero Francesca, Lorenzo Garattoni, David Garzón Ramos, Ken Hasselmann, Miquel Kegeleirs, Jonas Kuckling, Federico Pagnozzi, Andrea Roli, Muhammad Salman, and Thomas Stützle. Automatic Off-Line Design of Robot Swarms: A Manifesto. *Frontiers in Robotics and AI*, 6:59, 2019.

[3] Jan Dyre Bjerknes and Alan FT Winfield. On fault tolerance and scalability of swarm robotic systems. In A. Martinoli, F. Mondada, N. Correll, G. Mermoud, M. Egerstedt, M.A. Hsieh, L.E. Parker, and K Støy, editors, *Distributed Autonomous Robotic Systems. The 10th International Symposium (DARS 2010)*, pages 431–444. Springer, 2013.

[4] Johann Borenstein and Yorem Koren. Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(5):1179–1187, 1989.

[5] Manuele Brambilla, Arne Brutschy, Marco Dorigo, and Mauro Birattari. Property-driven design for swarm robotics: a design method based on prescriptive modeling and model checking. *ACM Transactions on Autonomous Adaptive Systems*, 9(4):17:1–17:28, 2014.

[6] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, 2013.

[7] Rachael L Brown. Mapping out the landscape: a multi-dimensional approach to behavioral innovation. *Philosophy of Science*, 89(5):1176–1185, 2022.

[8] Arne Brutschy, Lorenzo Garattoni, Manuele Brambilla, Gianpiero Francesca, Giovanni Pini, Marco Dorigo, and Mauro Birattari. The TAM: abstracting complex tasks in swarm robotics research. *Swarm Intelligence*, 9(1):1–22, 2015.

[9] Jennifer Carlson and Robin R Murphy. Reliability analysis of mobile robots. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, volume 1, pages 274–281. IEEE, 2003.

[10] Daniel Carrillo-Zapata, James Sharpe, Alan Frank T Winfield, Luca Giuggioli, and Sabine Hauert. Toward controllable morphogenesis in large robot swarms. *IEEE Robotics and Automation Letters*, 4(4):3386–3393, 2019.

[11] Anders Lyhne Christensen, Rehan O'Grady, and Marco Dorigo. From fireflies to fault-tolerant swarms of robots. *IEEE Transactions on Evolutionary Computation*, 13(4):754–766, 2009.

[12] Michael Crosscombe, Jonathan Lawry, Sabine Hauert, and Martin Homer. Robust distributed decision-making in robot swarms: Exploiting a third truth state. In Tony Maciejewski, editor, *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4326–4332, Vancouver, Canada, 2017. IEEE.

[13] Andrew J Davison and Joseph Ortiz. Futuremapping 2: Gaussian Belief Propagation for Spatial ai. *arXiv preprint arXiv:1910.14139*, 2019.

[14] Ophelia Deroy, Davide Bacciu, Bahador Bahrami, Cosimo Della Santina, and Sabine

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

Hauert. Shared awareness across domain-specific artificial intelligence: an alternative to domain-general intelligence and artificial consciousness. *Advanced Intelligent Systems*, page 2300740, 2024.

[15] Marco Dorigo, Dario Floreano, Luca Maria Gambardella, Francesco Mondada, Stefano Nolfi, Tarek Baaboura, Mauro Birattari, Michael Bonani, Manuele Brambilla, Arne Brutschy, Daniel Burnier, Alexandre Campo, Anders Lyhne Christensen, Antal Decugnière, Gianni A. Di Caro, Frederick Ducatelle, Eliseo Ferrante, Alexander Förster, Javier Martinez Gonzales, Jérôme Guzzi, Valentin Longchamp, Stéphane Magnenat, Nithin Mathews, Marco Montes de Oca, Rehan O'Grady, Carlo Pinciroli, Giovanni Pini, Philippe Retornaz, James Roberts, Valerio Sperati, Timothy Stirling, Alessandro Stranieri, Thomas Stützle, Vito Trianni, Elio Tuci, Ali Emre Turgut, and Florian Vaussard. Swarmanoid: a novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics & Automation Magazine*, 20(4):60–71, 2013.

[16] Marco Dorigo, Guy Theraulaz, and Vito Trianni. Swarm robotics: Past, present, and future [point of view]. *Proceedings of the IEEE*, 109(7):1152–1165, 2021.

[17] Gianpiero Francesca, Manuele Brambilla, Arne Brutschy, Vito Trianni, and Mauro Birattari. AutoMoDe: a novel approach to the automatic design of control software for robot swarms. *Swarm Intelligence*, 8(2):89–112, 2014.

[18] Lorenzo Garattoni, Gianpiero Francesca, Arne Brutschy, Carlo Pinciroli, and Mauro Birattari. Software infrastructure for e-puck (and TAM). Technical Report TR/IRIDIA/2015-004, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium, 2015.

[19] David Garzón Ramos and Mauro Birattari. Automatic design of collective behaviors for robots that can display and perceive colors. *Applied Sciences*, 10(13):4654, 2020.

[20] David Garzón Ramos and Mauro Birattari. Automatically designing robot swarms in environments populated by other robots: an experiment in robot shepherding. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12240–12247, Piscataway, NJ, USA, 2024. IEEE.

[21] Alessandro Giusti, Jawad Nagi, Luca Maria Gambardella, and Gianni A. Di Caro. Distributed consensus for interaction between humans and mobile robot swarms (demonstration). In *AAMAS '12: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems – Volume 3*, pages 1503–1504, Richland, SC, USA, 2012. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS).

[22] Pierre-Paul Grassé. La reconstruction du nid et les coordinations interindividuelles chez Bellicositermes natalensis et Cubitermes sp. la théorie de la stigmergie: essai d'interprétation du comportement des termites constructeurs. *Insectes Sociaux: International Journal for the Study of Social Arthropods*, 6(1):41–80, 1959.

[23] Jacqueline Harding and Nathaniel Sharadin. What is it for a machine learning model to have a capability?, 2024.

[24] Ken Hasselmann and Mauro Birattari. Modular automatic design of collective behaviors for robots endowed with local communication capabilities. *PeerJ Computer Science*, 6:e291, 2020.

[25] José Hernández-Orallo. Evaluation in artificial intelligence: from task-oriented to ability-oriented measurement. *Artificial Intelligence Review*, 48:397–447, 2017.

[26] Francis Heylighen. Stigmergy as a universal coordination mechanism I: definition and components. *Cognitive Systems Research*, 38:4–13, 2016.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

[27] Francis Heylighen. Stigmergy as a universal coordination mechanism II: varieties and evolution. *Cognitive Systems Research*, 38:50–59, 2016.

[28] Simon Jones and Sabine Hauert. Distributed Spatial Awareness for Robot Swarms. In *Distributed Autonomous Robotic Systems. The 17th International Symposium (DARS 2024)*. Springer, 2024.

[29] Simon Jones, Emma Milner, Mahesh Sooriyabandara, and Sabine Hauert. Distributed Situational Awareness in Robot Swarms. *Advanced Intelligent Systems*, 2(11):2000110, 2020.

[30] Simon Jones, Emma Milner, Mahesh Sooriyabandara, and Sabine Hauert. DOTS: An Open Testbed for Industrial Swarm Robotic Solutions, 2022.

[31] Simon Jones, Matthew Studley, Sabine Hauert, and Alan FT Winfield. Evolving behaviour trees for swarm robotics. In Roderich Groß, Andreas Kolling, Spring Berman, Emilio Frazzoli, Alcherio Martinoli, Fumitoshi Matsuno, and Melvin Gauci, editors, *13th International Symposium on Distributed Autonomous Robotic Systems (DARS 2016)*, London, UK, 2016. Springer.

[32] Simon Jones, Alan FT Winfield, Sabine Hauert, and Matthew Studley. Onboard evolution of understandable swarm behaviors. *Advanced Intelligent Systems*, 1, 2019.

[33] Andreas Kolling, Phillip Walker, Nilanjan Chakraborty, Katia Sycara, and Michael Lewis. Human interaction with robot swarms: a survey. *IEEE Transactions on Human-Machine Systems*, 46(1):9–26, 2016.

[34] Anna Kutschireiter, Melanie A Basnak, Rachel I Wilson, and Jan Drugowitsch. Bayesian inference in ring attractor networks. *Proceedings of the National Academy of Sciences*, 120(9):e2210622120, 2023.

[35] Suet Lee and Sabine Hauert. A data-driven method to identify fault mitigation strategies in robot swarms. In *International Conference on Swarm Intelligence*, pages 16–28. Springer, 2024.

[36] Suet Lee, Emma Milner, and Sabine Hauert. A data-driven method for metric extraction to detect faults in robot swarms. *IEEE Robotics and Automation Letters*, 7(4):10746–10753, 2022.

[37] Joel Lehman and Kenneth O. Stanley. Abandoning objectives: evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223, 2011.

[38] Guannan Li, David St-Onge, Carlo Pinciroli, Andrea Gasparri, Emanuele Garone, and Giovanni Beltrame. Decentralized progressive shape formation with robot swarms. *Autonomous Robots*, 43:1505–1521, 2019.

[39] Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Leslie Pérez Cáceres, Mauro Birattari, and Thomas Stützle. The irace package: iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58, 2016.

[40] Nithin Mathews, Anders Lyhne Christensen, Rehan O'Grady, Francesco Mondada, and Marco Dorigo. Mergeable nervous systems for robots. *Nature Communications*, 8(1):439, 2017.

[41] Emma Milner, Mahesh Sooriyabandara, and Sabine Hauert. Stochastic behaviours for retrieval of storage items using simulated robot swarms. *Artificial Life and Robotics*, 27(2):264–271, 2022.

[42] Emma Milner, Mahesh Sooriyabandara, and Sabine Hauert. Swarm performance

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

indicators: metrics for robustness, fault tolerance, scalability and adaptability. *arXiv preprint arXiv:2311.01944*, 2023.

[43] Francesco Mondada, Michael Bonani, Xavier Raemy, Jim Pugh, Christopher Cianci, Adam Klaptocz, Stéphane Magnenat, Jean-Christophe Zufferey, Dario Floreano, and Alcherio Martinoli. The e-puck, a robot designed for education in engineering. In Paulo Gonçalves, Paulo Torres, and Carlos Alves, editors, *ROBOTICA 2009: Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, pages 59–65, Castelo Branco, Portugal, 2009. Instituto Politécnico de Castelo Branco.

[44] Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. `http://arxiv.org/abs/1504.04909`, 2015.

[45] Nadia Nedjah and Luneque Silva Junior. Review of methodologies and tasks in swarm robotics towards standardization. *Swarm and Evolutionary Computation*, 50:100565, 2019.

[46] Shervin Nouyan, Alexandre Campo, and Marco Dorigo. Path formation in a robot swarm: self-organized strategies to find your way home. *Swarm Intelligence*, 2(1):1–23, 2008.

[47] Shervin Nouyan, Roderich Groß, Michael Bonani, Francesco Mondada, and Marco Dorigo. Teamwork in self-organized robot colonies. *IEEE Transactions on Evolutionary Computation*, 13(4):695–711, 2009.

[48] Carlo Pinciroli, Vito Trianni, Rehan O'Grady, Giovanni Pini, Arne Brutschy, Manuele Brambilla, Nithin Mathews, Eliseo Ferrante, Gianni A. Di Caro, Frederick Ducatelle, Mauro Birattari, Luca Maria Gambardella, and Marco Dorigo. ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, 6(4):271–295, 2012.

[49] Giovanni Pini, Arne Brutschy, Alexander Scheidler, Marco Dorigo, and Mauro Birattari. Task partitioning in a robot swarm: object retrieval as a sequence of subtasks with direct object transfer. *Artificial Life*, 20(3):291–317, 2014.

[50] Gaëtan Podevijn, Rehan O'Grady, and Marco Dorigo. Self-organised feedback in human swarm interaction. In *Proceedings of the workshop on robot feedback in human-robot interaction: how to make a robot readable for a human interaction partner, Ro-Man 2012*, 2012.

[51] Craig W Reynolds. Flocks, herds and schools: A distributed behavioral model. In *ACM SIGGRAPH Computer Graphics*, volume 21, pages 25–34. ACM, 1987.

[52] Michael Rubenstein, Alejandro Cornejo, and Radhika Nagpal. Programmable self-assembly in a thousand-robot swarm. *Science*, 345(6198):795–799, 2014.

[53] Muhammad Salman, David Garzón Ramos, and Mauro Birattari. Automatic design of stigmergy-based behaviours for robot swarms. *Communications Engineering*, 3:30, 2024.

[54] Muhammad Salman, David Garzón Ramos, Ken Hasselmann, and Mauro Birattari. Phormica: photochromic pheromone release and detection system for stigmergic coordination in robot swarms. *Frontiers in Robotics and AI*, 7:195, 2020.

[55] Melanie Schranz, Martina Umlauft, Micha Sende, and Wilfried Elmenreich. Swarm Robotic Behaviors and Current Applications. *Frontiers in Robotics and AI*, 7:36, 2020.

[56] Thomas D Seeley and Susannah C Buhrman. Nest-site selection in honey bees: how well do swarms implement the" best-of-n" decision rule? *Behavioral Ecology and Sociobiology*, 49:416–427, 2001.

[57] Touraj Soleymani, Vito Trianni, Michael Bonani, Francesco Mondada, and Marco Dorigo.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

Bio-inspired construction with mobile robots and compliant pockets. *Robotics and Autonomous Systems*, 74:340–350, 2015. Intelligent Autonomous Systems (IAS-13).

[58] Vivek H Sridhar, Liang Li, Dan Gorbonos, Máté Nagy, Bianca R Schell, Timothy Sorochkin, Nir S Gov, and Iain D Couzin. The geometry of decision-making in individuals and collectives. *Proceedings of the National Academy of Sciences*, 118(50):e2102157118, 2021.

[59] Vito Trianni and Stefano Nolfi. Self-organizing sync in a robotic swarm: a dynamical system view. *IEEE Transactions on Evolutionary Computation*, 13(4):722–741, 2009.

[60] Georgios Tzoumas, Lenka Pitonakova, Lucio Salinas, Charles Scales, Thomas Richardson, and Sabine Hauert. Wildfire detection in large-scale environments using force-based control for swarms of UAVs. *Swarm Intelligence*, 17(1-2):89–115, 2023.

[61] Gabriele Valentini, Eliseo Ferrante, and Marco Dorigo. The best-of-n problem in robot swarms: Formalization, state of the art, and novel perspectives. *Frontiers in Robotics and AI*, 4:9, 2017.

[62] Alan FT Winfield. Foraging robots. In *Encyclopedia of Complexity and Systems Science*, pages 3682–3700. Springer, 2009.

European Innovation Council
Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.