EMERGE

WP5 Emergent Awareness

D5.1 Measuring emergent awareness

Version: 1.0

Date: 29/09/2023

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 University of Bristol
Deliverable	D5.1 Deliverable
Contractual Delivery Date	30/09/2023
Actual Delivery Date	28/09/2023
Nature[1]	R
Dissemination level[2]	PU
Lead Beneficiary	University of Bristol
Editor(s)	Sabine Hauert, Suet Lee
Contributor(s)	Emma Milner, Simon Jones, Nadine Meertens, Andrea Cossu, Cosimo Della Santina, Ophelia Deroy
Reviewer(s)	Davide Bacciu, Andrea Cossu
Document description	This deliverable provides a framework to measure changes in behaviour based on different dimensions of awareness. We show the framework could be used across use-cases present in the project, from soft robots and networks to swarm robotics and natural systems. We also provide more in-depth metrics associated with swarm robotics.

[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	Suet Lee	30.08.2023	Created document
0.5	Ophelia Deroy	11.09.2023	Review and comments
0.8	Suet Lee, Sabine Hauert	25.09.2023	Complete document for internal review
1.0	Suet Lee, Sabine Hauert	29.09.2023	Document integrating review comments: submission ready.

[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

With proper "conceptual engineering," the concept of awareness presents a valuable pathway for the advancement of future artificial systems.

Our starting point is that awareness is a solution to technical, environmental, but also theoretical and human challenges - with evidence from recent large-scale investments from the European Innovation Council (25 millions in 2022). Giving awareness to AI however is more than an implementation problem left to computer scientists and engineers, and engages also end-users, cognitive scientists, and ethicists. We identify four core topics to bring these communities together.

1. Awareness brings system improvements : As the number of devices and systems increases, it is becoming more challenging to have central control over them, and it is essential to ensure that these agents can operate more locally and with more autonomy. Giving awareness to these agents can enhance their efficiency, resilience, and flexibility, allowing them to adapt to unforeseen situations and operate continuously - also saving costs and energy. Aware systems may also be easier to monitor and control.

2. Awareness brings explanatory value: we explain why using the term "awareness" is better than others which are being discussed, especially machine consciousness or artificial sentience. We argue that awareness can capture emergent properties in the system, and is a way to simplify the interactions between humans and systems or make AI explainable.

3. Awareness is measurable : we suggest that distinguishing dimensions of awareness (e.g. spatial, temporal, goal, self, group) offers a way to assess each system and compare systems, analogous to what is done for disorders of consciousness in humans, or awareness in non-human animals.

4. Awareness is a tractable ethical construct : the measurability (3) as well as the instrumental (1) and explanatory (2) value of the concept of awareness makes it possible to answer ethical questions about the relationship between humans and machines and the responsibilities that come with designing and deploying such systems.

In this report we focus on point 3, providing a framework to measure changes in behaviour based on different dimensions of awareness. We show the framework could be used across use-cases present in the project, from soft robots and networks to swarm robotics and natural systems.

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

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.

Ludwig Maximilian University of Munich	LMU	DE
Da Vinci Labs	DVL	FR

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

Document control 2
Version control 3
Abstract 4
Consortium **4**
Disclaimer 6
Acknowledgement 6
Table of contents 7
List of abbreviations 9
List of tables 9
List of figures 9
Introduction 11
State-of-the-art 11
Terminology for Examining Awareness 12
Implementation process to assess awareness metrics 14
Summary 14
Performance Metrics **15**
Example metrics 15
Performance evaluation 15
Example of Implementation 16
Use-case Studies **17**
Swarm robots 17
Soft robots 19
Networks 20
Animals: Rhesus monkeys 21
Discussion **22**
Comparisons of awareness 22
Comparing between varied systems 22
Aggregation of performance scores 23
Aggregation over tasks 23
Aggregation over metrics 23
Next Steps **24**
Conclusion **25**
References **26**
Appendix A - Swarm Performance Indicators **27**
Introduction 27
Literature review 28
Research gap 28
Terminology 29
Ability to cope with agent faults 29
Ability to cope with changing swarm size 30
Ability to cope with changing external parameters 31

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 Performance Indicators 32
Scalability metric 32
Definition for scalability 33
Derivation of Scalability metric, S 33
Fault Tolerance and Robustness metrics 34
Fault Tolerance metric 34
Definition of Fault Tolerance 34
Derivation of Fault Tolerance metric, FT 35
Robustness metric 35
Definition of Robustness 35
Derivation of Robustness metric, R 36
Adaptability metric 37
Definition of Adaptability 37
Derivation of Adaptability metric, A 38
Summary of metrics 39
Results of metric testing 39
How the results will be structured 40
Scalability 40
Fault tolerance and Robustness 40
Adaptability 40
Use case example: Logistics scenario 41
Experimental set-up 41
Algorithm 43
Scalability results 44
Incremental Scalability 44
Scalability from 1 agent 45
Specification for Scalability 45
Fault Tolerance and Robustness results 46
Fault Tolerance results 46
Robustness results 47
Adaptability results 48
Adaptability to changing warehouse width 48
Adaptability to changing delivery area width 49
Adaptability to changing number of items 50
Use case example: Collective decision making 52
Use case description 52
Experimental set-up 52
Algorithm 53
Scalability results 53
Incremental Scalability 53
Scalability from 10 agents 55
Specification for Scalability 55
Fault Tolerance and Robustness results 55
Fault Tolerance results 55

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.

Robustness results 57
Adaptability results 57
Discussion 59
Successful metrics 59
Context and Boolean classifications 59
Clear definition of terminology of metrics 60
Using proportional change 60
Variability in results 61
Uses 62
Limitations 62
Limitations of the Adaptability metric 62
Limitations of the Scalability metric 62
Limitations of the Fault Tolerance and Robustness metrics 63
Conclusions 63
References 63

List of abbreviations

GA	Grant Agreement
CA	Consortium Agreement
IPR	Intellectual Property Rights
WP	Work Package

List of tables

Table 1: Summary of the Swarm Performance Indicators 38
Table 2: Experiment parameters for the logistics use case 41
Table 3: Experiment parameters for the collective decision making use-case 52

List of figures

Figure 1: Hypothetical awareness profile 11
Figure 2: Implementation process of framework 12
Figure 3: Swarm robots in logistics task 16
Figure 4: Soft robots use-case example 18
Figure 5: Networks use-case example 19
Figure 6: Animals use-case example 20
Figure 7: Extension to framework 23
Figure 8: Aggregation of performance metrics 23
Figure 9: Swarm simulation for logistics task 41

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: Scalability results (logistics) 43
Figure 11: Robustness results (logistics) 46
Figure 12: Adaptability results: changing warehouse width (logistics) 48
Figure 13: Adaptability results: changing exit width (logistics) 49
Figure 14: Adaptability results: changing number of items (logistics) 50
Figure 15: Swarm simulation for decision making 51
Figure 16: Scalability results (decision making) 53
Figure 17: Robustness results (decision making) 55
Figure 18: Adaptability results: changing site quality (decision making) 57
Figure 19: Variability in results for logistics use-case 60

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.

Introduction

Awareness in artificial systems has the potential to make them easier to monitor, control, and more effective at their tasks. Understanding awareness, and engineering for awareness, requires us to have impartial metrics that can measure changes in behaviour based on different dimensions of awareness. In this deliverable we define methods to measure performance of a system based on its dimensions of awareness. We then show the framework could be used across use-cases present in the project, from soft robots and networks to swarm robotics and natural systems.

Awareness cannot be measured directly: we introduce the terminology involved in the study of awareness, which together will allow us to evaluate awareness in a given context for any system.

The main concepts in this study are:

- Dimensions of awareness
- Capacities and mechanisms
- Performance metrics
- Evaluation tasks

In the rest of this section, we explain each concept and explain their relevance and relation to each other. The conceptual structure will give us a concrete approach to implement a measure of awareness in real-world use-case studies.

State-of-the-art

There is no standard approach to, or indeed existing framework for, the measurement of awareness across heterogeneous systems to the best of our knowledge. However, there exist measures for other qualities and traits of the various systems which we are interested in, discussed briefly in the following:

- **Animal systems**

  For the animal literature there is an existing framework for comparing and measuring animal consciousness, across different species - which allows the species to vary across many dimensions. Birch et al (2020) [11] formulated this specifically with phenomenal consciousness in mind (subjective experience), not awareness as defined in this project. They pick up on the following dimensions: perceptual richness, evaluative richness, integration at a time, and across time, and self-consciousness. The dimensions are not expected to be uncorrelated, but are conceptually distinct from each other. Birch et al. propose investigating these dimensions by applying existing experimental paradigms such as mirror-mark, or trace conditioning paradigms. As awareness is distinct from consciousness, the dimensions under investigation will not map from this existing framework by Birch et al. unto the one proposed here. But, some of the suggested experimental paradigms and their framing in a multidimensional conceptual framework can function as a stepping stone for further development of the framework proposed here. Moreover, the experimental paradigms from the animal cognition literature can inform the tasks for investigating artificial awareness. D1.1 provides more background on dimensions of awareness.

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.

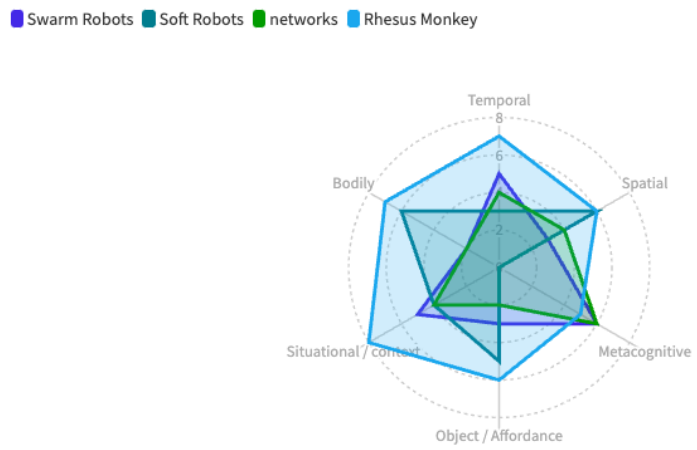

Figure 1: Hypothetical awareness profile for swarm robots, soft robots, networks and rhesus monkeys: each “spine” of the graph represents the axis of value for a different dimension of awareness.

- **Swarm robotics**

  Traits defined in the swarm literature include robustness/fault tolerance [1, 2, 3, 4, 5, 6, 7, 8, 9], scalability [5, 1, 3, 6, 10, 8, 2, 4] and adaptability/flexibility [3, 5, 1, 2]. However, there is a lack of precision in how such traits are defined with varying interpretations. There is currently no defined measure for awareness in swarms. In appendix A, we provide a set of metrics (Swarm Performance Indicators) which are formulated with reference to existing metrics and go towards examining the awareness of swarm robot systems.

Terminology for Examining Awareness

We use the following terminology to define different constructs related to awareness:

- **Dimensions of awareness:** awareness does not vary along a single scale (more or less) but different directions. The term “dimensions” refers to the different aspects of awareness considered and corresponding scales (e.g. spatial, temporal, meta-cognitive). The dimensions allow for comparison between different systems without assuming a single standard. We can draw comparisons between the dimensions of different systems in a non-linear, flexible manner.
- **Capacities**: each dimension of awareness needs to be associated with underlying mechanisms or sets of processes, which are here referred to as “capacities”. A system can eventually operate with or without awareness, or with more or less awareness, and we expect differences in the exercise/operation of the system to show depending on the differences in awareness.
- **Mechanisms of awareness:** refers to the general configuration or means through which a capacity operates. We follow here the definition of mechanism provided in cognitive sciences (“entities and activities organized such that they are productive of regular changes from start or set-up to finish or termination conditions”, Machamer, Darden, and Craver (2000) [12], p. 3) which allows for multiple physical realisations (e.g. different kinds of sensors, different materials) as this allows the conceptual framework to apply to multiple physical systems.

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 metrics**: we can evaluate the performance of a system for a given task using defined metrics, and compare the performance on a given dimension when a capacity is present and when it is not.
- **Evaluation tasks:** tasks chosen in order to evaluate the performance of the system with respect to selected capacities and the respective dimensions. The chosen task should be feasible with and without the capacity present.

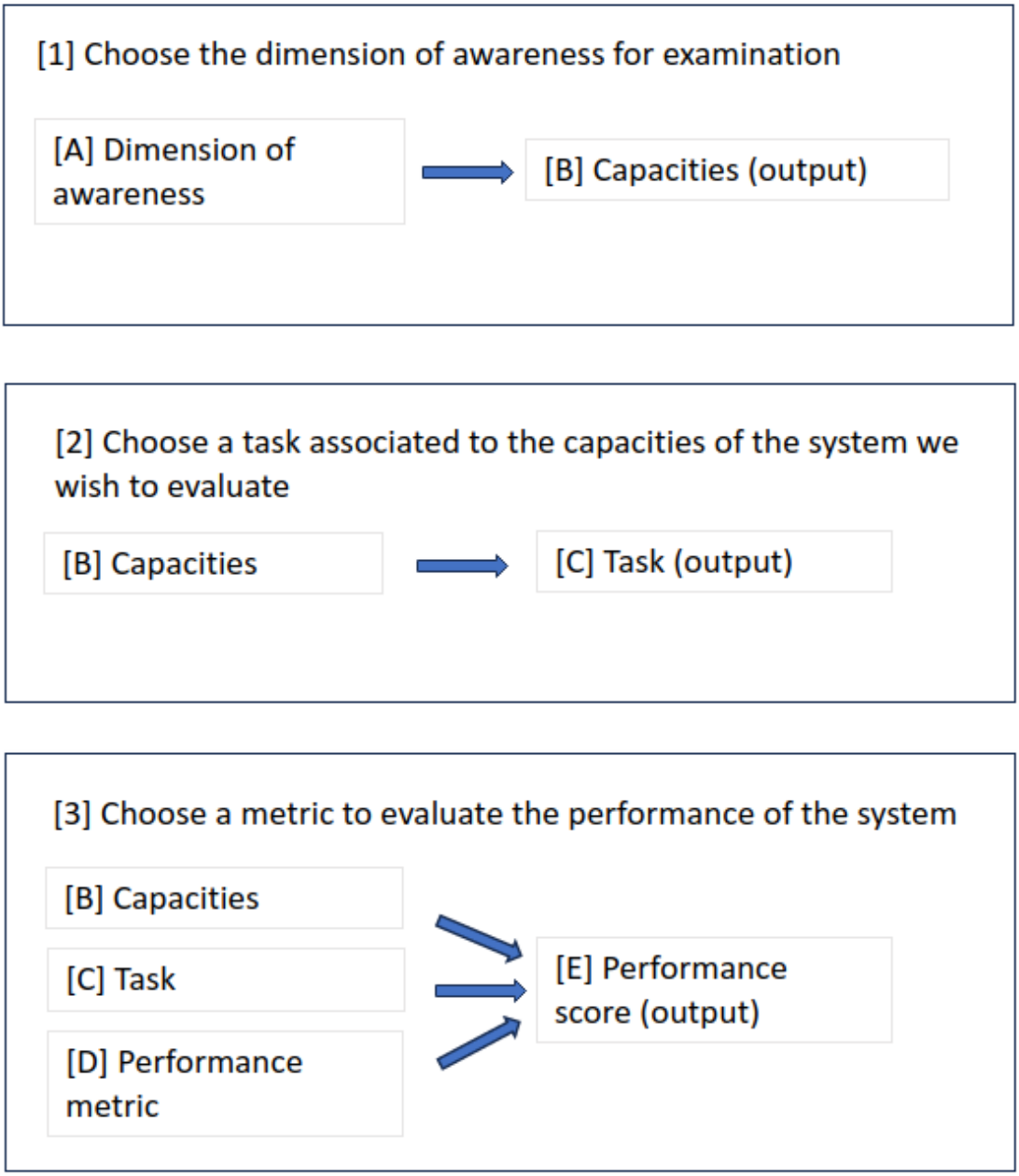

Figure 2: Framework to assess the awareness of a system. The aim is to measure the performance of a system on a task based on a change in their dimensions of awareness implemented using specific capacities.

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.

Implementation process to assess awareness metrics

Figure 1 lays out the steps needed to measure the performance of a system on a task given a change in their dimensions of awareness implemented using specific capacities:

1. Awareness cannot be measured directly: instead we can evaluate the performance of the system with respect to associated capacities.
2. In order to evaluate performance via defined metrics, we select tasks which allow for a differential in capacities of the system.
3. Performance is evaluated via defined metrics: we evaluate the change in performance of the system with and without a capacity present for the selected task. Basic metrics might include robustness, adaptability, speed, for example. Metrics related to human systems/interactivity might include trust, explainability, responsibility.

Summary

The terminology introduced together with the structural relation between them provides a framework which we can apply in real-world use-case studies. In the following sections, we will discuss specific metrics for performance evaluation and finally, use-case studies for various systems across work packages.

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 Metrics

We define metrics to measure the performance of the system with respect to properties relevant to dimensions of awareness. The metrics are agnostic with respect to the dimension of awareness being assessed. We are interested in the difference in performance between a system with a capacity present and without.

Example metrics

Operation-related:

- Speed: measures the speed of a system in completing a task.
- Robustness: measures how well the system is able to function in a task when there are errors or faults present.
- Adaptability: measures how well the system is able to function when there is a range of environments in the task.
- Scalability: measures how well the system functions as it grows in scale (e.g. number of internal components, size of network, etc.).
- Energy: measures the energy use of the system performing the task.
- Autonomy: measures the ability of the system to act without external input to complete a task.

Human-related:

- Trust: measures how trustworthy the system is from a human operator's perspective.
- Explainability: measures how explainable the system is to a human operator.
- Safety: measures how safe the system is for use in a human environment.

Performance evaluation

For a given set of capacities C = $\{c_i\}$, we will evaluate performance with respect to each capacity independently.

For a given metric M and task T, we evaluate three variables with respect to the selected capacity under evaluation, $\{c_i\}$:

- Optimal performance: P_p
- Performance with c_i present: P_m
- Performance without any of the capacities in C present: P_n

Performance is measured using metric M. In particular, "optimal" performance may be evaluated experimentally as the performance of the system when the optimal action policy is executed for a given task. This is the upper limit of performance attainable by the system. We assume that $P_m > P_n$, i.e. a capacity will always improve performance in the system.

The quantity we are interested in is given by the equation:

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.

$$\mathbf{P} = (P_m - P_n) / (P_p - P_n)$$

This gives us the change in performance with and without capacity, c_i, with a normalising factor $N = P_p - P_n$. P has values in the range of 0 to 1. When P is small or close to 0, this indicates that the capacity c_i has low benefit to the system in executing the task. When P is high or close to 1, this indicates that the capacity c_i has large benefit to the performance of the system.

Example of Implementation

A robot is tasked with moving from point A to B in a room without colliding with obstacles. Researchers want to evaluate the performance P of the system in avoiding obstacles when given spatial awareness. Spatial awareness is implemented using two different capacities (camera and IR sensor). The performance metric used relates to the time taken to go from A to B.

1. Set up obstacles in the arena.
2. Calculate the best possible route and the best time the robot could take
    - Optimal agent time = 5s, $P_p = 1/5 = 0.2$
3. Disable both the camera and IR sensor to evaluate performance
    - Time without camera nor IR sensor = 100s, $P_n = 1/100 = 0.01$
4. Camera test:
    - Test performance with only the camera working P_m
    - Time with camera = 10s, $P_m = 1/10 = 0.1$
    - Compute **P** = (0.1 - 0.01) / (0.2 - 0.01) = 0.48
5. IR sensor test:
    - Test performance with only the IR sensor working P_m
    - Time with IR sensor = 6s, $P_m = 1/6 = 0.17$
    - Compute **P** = (0.17 - 0.01) / (0.2 - 0.01) = 0.84

Camera test result, P = 0.48

IR test result, P = 0.84

In conclusion, the IR sensor capacity is more beneficial to the performance of the robots in detecting obstacles, in this evaluation of spatial awareness.

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.

Use-case Studies

Swarm robots

Goal

We evaluate the spatial awareness of a system using the Distributed Situational Awareness capacity in a swarm logistics task.

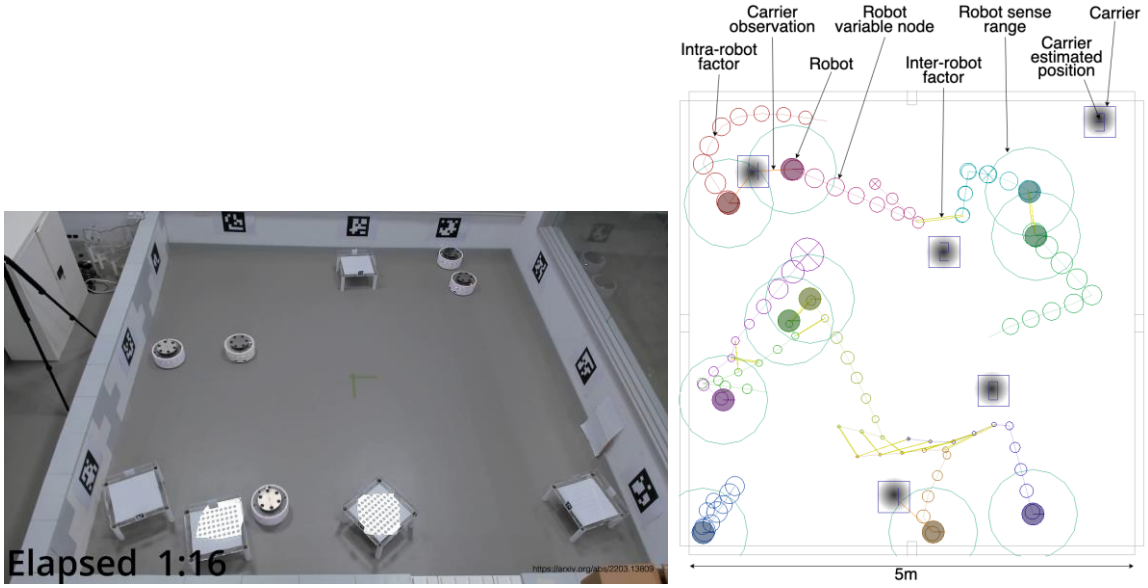

Figure 3: Left - A swarm of DOTS robots performing a logistics task (finding boxes and bringing them to the deposit area). Right - simulation of Distributed Situational Awareness, with DOTS robots estimating global locations of carriers with only local observations and message passing.

Definitions

Task: Find boxes and move them to the deposit area on the left of the arena.

Dimensions of awareness: Spatial

Capacity: Distributed Situational Awareness (DSA) infers the swarm-centric global values of the location of boxes using only local observations and messaging.

Mechanism: DSA is implemented by constructing a global and distributed frame-of-reference for the swarm using Gaussian Belief Propagation (GBP). Each robot gathers information about other nearby robot relative locations using the four perimeter cameras and constructs a local factor graph fragment. Local message passing implements the GBP algorithm achieving rapid convergence of the complete distributed factor graph, giving each robot knowledge of its global position within the swarm. This knowledge is used to track and communicate observed carrier locations.

Performance metrics:

- Speed

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.

- Robustness
- Adaptability
- Scalability

Note that metrics for robustness, adaptability, and scalability had not previously been defined for swarm systems. Appendix A provides a full definition of these metrics and demonstrates how they can be used in swarm use cases. We expect similar metrics can be used outside of swarm systems (e.g. robustness and adaptability), for example in the context of soft robotics or animal systems.

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.

Soft robots

Goal

We evaluate object awareness of a system using the tactile sensing capacity in a grasping task.

Figure 4: Adapted from: American Institute of Physics (2013), Tech Xplore [13].

Definitions

Dimension of awareness: Object awareness

Capacity: Tactile sensing

Task: The soft manipulator should be able to identify and recognize different objects. The manipulator can then select an appropriate grasping approach for gripping the object.

Performance metrics:

- Stability
- Speed
- Energy
- Safety

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.

Networks

Goal:
We evaluate the metacognition of a system using the confidence prediction capacity in an opt-out task designed for artificial neural networks.

Definitions:
Dimension: Meta-cognition

Capacity: Confidence prediction

The system should be able to independently determine whether or not it “feels” confident in predicting the target answer (e.g. a class) for a given input. The system can either reject the prediction (i.e., refusing to predict) or it can choose to predict.

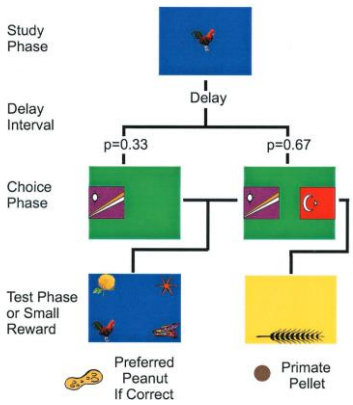

Figure 5: Image credit: Hampton (2001) [14].

Task: Opt-out task for artificial neural networks

A network is trained to classify a fixed set of images / time-series. In particular, the definition of network in this example goes beyond the classical sense of implementation on a microcontroller but is inclusive of implementation via mechanical means, e.g. non-linear oscillators and archetypes.

The network has the opportunity to “reject” the classification, thus refusing to predict. At the end of the training phase, the network is evaluated against a set of examples from the same classes seen during training. We expect the network to perform better when we allow it to reject, than when it is always forced to predict.

Performance metrics:

- Accuracy
  - Accuracy of predictions when the network is forced to always predict
  - Accuracy of predictions when the network is allowed to reject
- Change in accuracy
  - Change in accuracy for different rejection thresholds

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.

Animals: Rhesus monkeys

Goal

We evaluate temporal awareness of a system using the working memory capacity in a match-to-sample task.

Definitions

Dimension: Temporal awareness

Capacity: Working memory

In a delayed matching-to-sample task, the animal is presented with a stimulus and after a delay period (during which the stimulus is not present), it has to pick from two comparison stimuli which was the one that matched the stimuli seen before the delay. Depending on the success this is followed by either reward or punishment (better versus lesser good food).

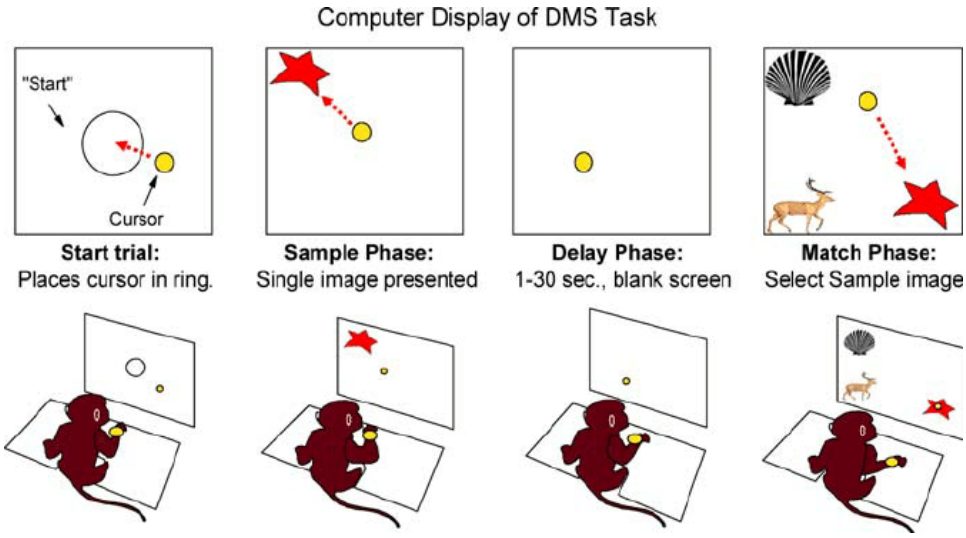

Figure 6: Image credit: Porrino, L. J. et al. (2005) [15].

An opt-out can be introduced if needed, moves it to the meta-level (second-order awareness) see e.g. https://www.pnas.org/doi/full/10.1073/pnas.071600998

Task: Delayed Match-to-sample

Performance:

- Robustness
- Adaptability
- Autonomy
- Speed

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.

Discussion

In this framework, we have presented:

- The terminology required for examining dimensions of awareness.
- The implementation process to apply the framework.
- A performance measure which is evaluated via metrics to be defined.
- Use-case examples demonstrating how the framework may be applied for selected dimensions of awareness and their associated capacities across four systems: swarm robots, soft robots, networks, and animals (rhesus monkeys).

The following is a brief discussion of the considerations made in forming the framework and how it may be extended in future work.

Comparisons of awareness

The proposed framework is formulated to enable comparisons to be made between various systems, along various dimensions of awareness. The dimensions considered in the use-cases are spatial, temporal, object awareness and metacognition. Another dimension that could be considered in the future might be situational or context awareness. By emphasising flexibility in the association of capacities to dimensions, and the choice of task on which to evaluate the system performance, it is made possible to do such inter-system comparisons. The dimensions as offered are not all operating on the same level, metacognitive awareness especially may be considered a second-order dimension of awareness. As metacognition entails a system's awareness of its own 'thought' processes, this means it is predicated on the system already possessing some prior awareness.

It may also be possible to examine the "collective awareness" of a system, loosely defined here as the sum of performance scores with respect to a dimension of awareness across individuals in a system. This may go towards examining the "emergence" of awareness in collectives, where the overall collective awareness is greater than that of any individual.

Comparing between varied systems

In order to demonstrate the way in which the framework can be applied across very different (heterogeneous) systems, we have provided four use-case examples. These examples illustrate different dimensions of awareness and different associated capacities and tasks. It needs to be taken into account that although these systems might be rather different, the performance metrics need to be stable across all systems to ultimately allow for comparisons. The proposed framework therefore makes it possible to evaluate performance with the same set of metrics - to be specified for each use-case.

The framework as currently formulated assumes the possibility of allowing for a delta in capacity, i.e. the option to test the system with and without the relevant capacity. For the human designed systems this will be achievable, however it needs to be noted that this is unlikely to apply to living systems, which form the theoretical basis for the framework. In biological systems isolating the various capacities under consideration from each other, 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.

investigating them independently is likely to be difficult to achieve, especially considering that awareness as a concept anticipates the possible interrelatedness (or at least interplay) of various capacities (and along various dimensions). Moreover, 'turning off' a capacity, although possible in artificial systems, would require knowing and isolating the relevant mechanisms (e.g. neural correlates) - which for most biological systems (especially non-human animals) is not currently achievable.

Aggregation of performance scores

Aggregation over tasks

In the framework, we assume that a single selected task is sufficient to evaluate performance with respect to a single capacity. In the case where a single task is not enough and multiple tasks are required to evaluate a capacity, multiple scores will be produced for each performance metric. These scores may then be aggregated by taking the mean score for each metric, for example, to produce a final performance score.

Aggregation over metrics

The first step in the process of implementation is selecting a dimension of awareness for examination. The final output are performance scores across metrics, evaluated with respect to the capacities associated with the dimension. In an extension to the framework, we may complete the process by aggregating scores over metrics to produce a single score for the initially selected dimension of awareness. This is an intuitive step to take but non-trivial, as the method of aggregation will need to be carefully considered. For example, we may produce an aggregate by taking a weighted sum of scores - but how should we determine the weights? If we are able to compute an aggregate that "makes sense" then this would allow us to directly compare the dimensions of awareness within a system. Should such a single score for a selected dimension become available, and if this can be done for all the dimensions under consideration, then we would have the system's awareness profile. Such an awareness profile would give a general overview of the systems awareness, and again allow for comparison between various systems (possibly both biological and artificial ones). A similar approach has been employed for frameworks of animal consciousness (e.g. Birch et al., 2020).

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.

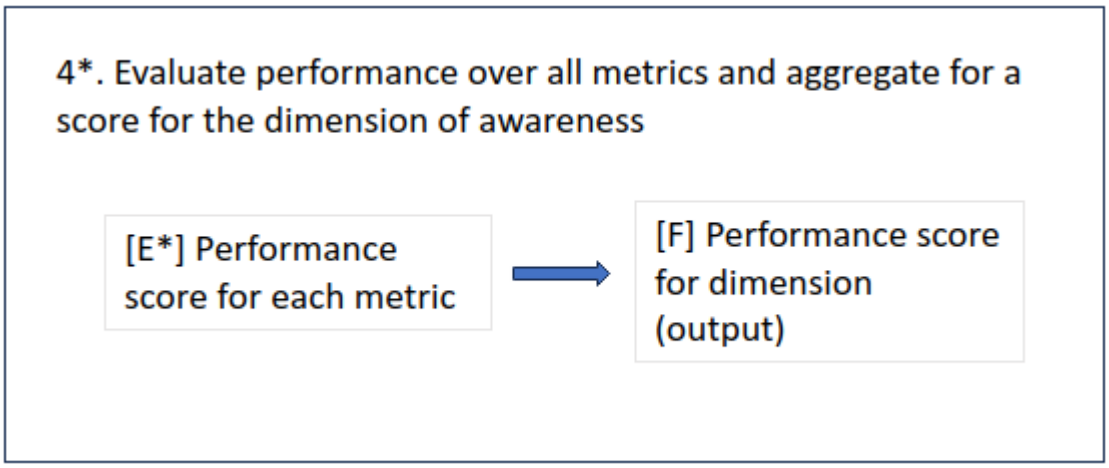

Figure 7: Extension to the process: a final step where we aggregate performance scores across metrics would allow us to compute a single score for each dimension of awareness in the system. Selection of aggregation method is non-trivial.

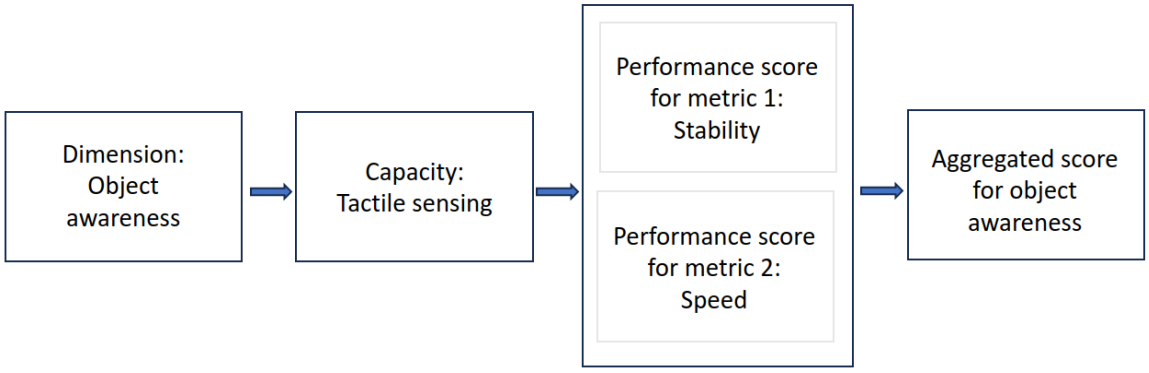

Figure 8: Example of aggregation of performance metrics for single output score for object awareness.

Next Steps

As a first step, we aim to use this framework to measure the impact of different awareness capacities on performance, focussing on the swarm scenario. The swarm use-case is a logical starting point for application of the framework as it allows for flexible testing of a variety of capacities associated with the dimensions of awareness of interest. In addition, we have well-defined metrics for evaluating various properties of swarms (see appendix A) [16]. Finally, swarms have emergent properties, aligning therefore with the overall aim of WP5 to study emergent awareness. The framework will be used to automatically optimise for performance of a swarm logistics task using dimensions of Spatial Awareness. In particular we will consider systems with different implementations of Distributed Spatial Awareness (DSA) including hard-coded capacities such as Gaussian Belief Propagation, 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.

automatically designed ones using artificial evolution. The result will be both an optimised behaviour, but also an optimised awareness selection process to achieve this behaviour.

Conclusion

Measuring awareness is critical to understand, and design for it. The different dimensions of awareness implemented on a system have the potential to improve the performance of the system. Another potential benefit would be improved interfacing with society, by enabling better understanding and increased control.

In this deliverable we present a framework to assess the performance gain provided by different dimensions of awareness, and apply the framework to 4 use-cases relevant to the EMERGE. In the future we will apply this metric to both quantify awareness in our systems, and evolve solutions of systems based on awareness requirements. As a first step we provide a rich library of performance metrics which can be used to assess the awareness of swarm systems (scalability, robustness, and adaptability).

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] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, “Swarm robotics: a review from the swarm engineering perspective,” Swarm Intelligence, vol. 7, pp. 1–41, 2013.

[2] J. Harwell and M. Gini, “Swarm engineering through quantitative measurement of swarm robotic principles in a 10,000 robot swarm,” arXiv preprint arXiv:1907.03880, 2019.

[3] E. S ̧ahin, “Swarm robotics: From sources of inspiration to domains of application,” in Swarm Robotics: SAB 2004 International Workshop, Santa 1 Monica, CA, USA, July 17, 2004, Revised Selected Papers 1. Springer, 2005, pp. 10–20.

[4] M. Schranz, G. A. Di Caro, T. Schmickl, W. Elmenreich, F. Arvin, A. S ̧ekercio ̆glu, and M. Sende, “Swarm intelligence and cyber-physical systems: Concepts, challenges and future trends,” Swarm and Evolutionary Computation, vol. 60, p. 100762, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210650220304156

[5] M. Dorigo, G. Theraulaz, and V. Trianni, “Swarm robotics: Past, present, and future [point of view],” Proceedings of the IEEE, vol. 109, no. 7, pp. 1152–1165, 2021.

[6] M. K. Heinrich, M. Wahby, M. Dorigo, and H. Hamann, “Swarm robotics,” 2022.

[7] J. Bjerknes and A. F. Winfield, On Fault Tolerance and Scalability of Swarm Robotic Systems, 01 2013, vol. 83, pp. 431–444.

[8] J. P. Hecker and M. E. Moses, “Beyond pheromones: evolving error-tolerant, flexible, and scalable ant-inspired robot swarms,” Swarm Intelligence, vol. 9, pp. 43–70, 2015.

[9] C. Prehofer and C. Bettstetter, “Self-organization in communication networks: principles and design paradigms,” IEEE Communications Magazine, vol. 43, no. 7, pp. 78–85, 2005.

[10] H. Hamann and A. Reina, “Scalability in computing and robotics,” IEEE Transactions on Computers, vol. 71, no. 6, pp. 1453–1465, 2022.

[11] Birch, J., Schnell, A. K., & Clayton, N. S. (2020). Dimensions of Animal Consciousness. Trends in Cognitive Sciences, 24(10), 789–801. https://doi.org/10.1016/j.tics.2020.07.007.

[12] Machamer, P., Darden, L., & Craver, C. (2000). Thinking about Mechanisms. Philosophy of Science, 67(1), 1-25. doi:10.1086/392759.

[13] American Institute of Physics (2021). Flexible tentacle like robotic manipulators inspired by nature. Tech Xplore, https://techxplore.com/news/2021-12-flexible-tentaclelike-robotic-nature.html.

[14] Hampton (2001), R. R. Rhesus monkeys know when they remember. Proc Natl Acad Sci USA 98, 5359–5362.

[15] Porrino, L. J., Daunais, J. B., Rogers, G. A., Hampson, R. E., & Deadwyler, S. A. (2005). Facilitation of task performance and removal of the effects of sleep deprivation by an ampakine (CX717) in nonhuman primates. PLoS biology, 3(9), e299.

[16] Milner, E. (2023). Swarm performance indicators and algorithms for logistics use cases, PhD Thesis.

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.

Appendix A - Swarm Performance Indicators

Swarms have distributed control and so are assumed to inherently have superior robustness, scalability and adaptability compared to centralised multi-agent systems. However, these features have generally only been defined qualitatively and there is a lack of quantitative metrics and experiments to measure the claimed parameters. Swarm Performance Indicators are defined here as Key Performance Indicators for swarm features but can be applied to multi-agent systems with centralised control as well. These swarm features are metrics for robustness, fault tolerance, adaptability and scalability. We introduce Swarm Performance Indicators here towards use within the proposed framework as metrics for evaluating awareness.

Introduction

Swarm systems are a subset of multi-agent systems, which have distributed rather than centralised controllers. Distributed control means that the agent's behaviour is based on local sensing and communication. Distributed control does not use globally available information and does not have a leader who gives instructions, such as planning path way-points to reach. The benefits of distributed control in a group of agents are: redundancy [1]; ability to scale to high numbers without increasing the complexity of control [2]; and the ability to self-organise [3]. These benefits are extrapolated by researchers to incorrectly state their swarm systems are inherently robust to failures, scalable to large numbers and adaptable to environmental changes. However, a number of researchers have pointed out the issues with assuming these traits are automatic in swarms [1, 4] without further evidence or description. For example, what types of failures is a system robust to? How much performance can a system lose under faults before it is deemed not to be robust? It is not enough to call a system robust without first defining the meaning and to what the system is tolerant to and what its limits are therein [5]. Similarly, the extent of scalability is rarely discussed when asserting it for a given swarm platform or behaviour. Scalability can be defined as an ability to scale up and down in size without detriment to the performance of the swarm [6], or as the superlinearity of swarm performance with increased swarm size [7]. Finally, adaptability to changes in the environment or task is asserted as a given in a swarm system, due to self-organisation. It is true that self-organisation can cause good Adaptability to changes in the surroundings, but the extent to which this is true and how to measure it is not defined.

To bring swarm robotic solutions into real-world use, it is necessary to create metrics for their
benefits of robustness/fault tolerance, scalability and adaptability, which we term Swarm Performance Indicators (SPIs). These metrics will allow swarms to be meaningfully compared to non-swarm systems, which would otherwise be compared with Key Performance Indicators (KPIs) alone. KPIs, such as time taken, operating costs, throughput, speed of delivery and so on, are used in traditional industries to quantify and measure critical success factors. They are important for identifying gaps between actual and desired overall performance [8]. Until now, robustness, scalability and adaptability have commonly only been described qualitatively, with little mathematical proof or measurement and no way of knowing the 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.

of these features. Using KPIs alone would miss these key factors that make swarms so useful. It is difficult to make a business case for swarms compared to other systems, without measures for these factors to describe what makes them beneficially different. It is difficult to compare swarms to other swarms without quantitative measures to describe these traits. To say that two swarms are both robust is not as helpful as to say that one is more robust than the other and to give evidence for where and why this is the case. Metrics for these traits can be used to compare across swarm solutions to improve their robustness, scalability and adaptability. If we have a measurement to attribute to these traits then we can gauge improvement in them as we try to design better swarms. The SPIs suggested in this work aim to solve this. To examine their efficacy they are applied to two use cases which use swarm algorithms. The SPIs measure robustness, fault tolerance, scalability and adaptability in a system as it performs a task.

Literature review

The Swarm Performance Indicator traits that are described here are as follows:

- Ability to cope with faulty agents in the system
- Ability to cope with changing swarm size
- Ability to cope with changing external parameters (environmental or task parameters).

The aim of the following literature review is to define the state-of-the-art in metrics to describe these traits. The literature review will justify the need for these metrics and corroborate the logic which is used to develop them from qualitative into quantitative definitions. It is also necessary to define names for these traits so that they can be more easily referred to, but there is some dispute in the literature about the terminology to be used. This is explored in this review.

Research gap

Swarm Performance Indicators (SPIs) are here created as a form of Key Performance Indicators (KPIs) to describe the beneficial features of swarm behaviours. The distributed control of swarms gives agents redundancy and means they use only local sensing and communication [6]. These features of distributed control are often said to inherently give robustness to faults, scalability to large group sizes and adaptability to changing environmental conditions and tasks. However, it is incorrect to state that these traits are inherent in all distributed systems or swarms, without proof or evidence. These traits will have limits which must be clearly defined if swarm robotics is to move from the laboratory into real, commercial usage [1].

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.

Terminology

These typical benefits of distributed control that are described as Swarm Performance Indicators here, have been named using different terminology across the literature. The following is a discussion of the different terms and conclusions about the terminology that will be used in this paper.

Ability to cope with agent faults

The first swarm behaviour trait that requires a universal term is:

(1) The ability to cope with faulty agents in the system

In the literature this has been defined as either robustness [6, 9, 10, 11] or as fault/error tolerance [4, 2, 5, 12], although the two words are used interchangeably by some of these sources. Robustness is defined as change in performance due to damage or failures in the literature [10, 6, 12, 5, 9, 11, 13]. With the exception of Dorigo et al. (2021) who define robustness as the response to changing environmental conditions, and Sahin (2004) who gives robustness to describe both individual failures and disturbances in the environment. Harwell and Gini (2019) directly contradict this inclusion of environmental factors in the term robustness by stating it is "in response to internal, as opposed to environmental stimuli" [9]. Winfield and Nembrini (2006) say it is not surprising that there is a lack of precision in the use of the term ‘robustness’ because there are so many reasons that a swarm could be robust, from mechanical reliability due to simple hardware to distributed control leading to no single point of failure [1].

(1) is often described by performance loss in the literature [9, 4, 10, 6] but how much performance loss is allowed before the system is considered to have "poor" robustness/fault tolerance differs. Harwell and Gini (2019) state that "performance should remain the same" [9], Sahin (2004) only requires that the swarm should "operate, although at a lower performance" [14] and Dorigo et al. (2021) allows for a "graceful degradation of performance in the presence of system faults" [4]. The metric given in this research therefore has a scalar value, rather than a purely yes/no result. The value is inversely proportional to the performance loss. This result can be used for all three definitions, if the user wanted to take one as a preference over the others. For example, if the user did not allow for any performance loss (using Harwell and Gini (2019)), then they would be looking for a higher metric value than a user who was allowing for a lower performance, following Sahin (2004). However, below 0 the metric is no longer gracefully degrading in performance, and the system could be deemed to be not tolerant to faults, according to the definitions given here. This "graceful degradation" is defined in the work presented as the point at which the performance loss surpasses the agent loss due to faults.

Winfield and Nembrini (2006) discusses how fault tolerance measurements and assurances must be stated alongside the failure mode to which the system is fault tolerant. They analyse fault tolerance by individual fault type and effect. With the metrics described in this work, when stating measures for performance change under faults, a particular failure mode is considered and is clearly stated alongside the metric value for it to be valid.

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.

Bjerknes and Winfield (2013) discuss how self-repair can be used in the discussion of fault tolerance, where the swarm agents are able to recover performance that had previously been lost to faulty agents. They describe the failure mode of malicious agents as acting as "anchors" to the working agents, who must self-repair, to not have their performance degraded by the faulty agents [5]. This has an increased negative effect on performance due to faults, where they do more than remove their own performance contribution but also degrade the performance of surrounding working robots. However, what is missing from this discussion of faults is the contribution that the faulty agents could still be making to the positive performance. It is not necessarily true that the performance under faults is always worse than if the swarm were scaled down to remove these agents altogether. It is proposed here that if the performance under faults was better than the performance of the swarm when those agents are removed, then the system is tolerant to those errors in some significant way. Harwell and Gini (2019) discuss this in part by including the robustness to fluctuating population size due to robot failures as an axis in their definition of overall robustness of a swarm [9].

There are two factors to consider when discussing how well a system copes with agent faults: how graceful the degradation in performance is; how much the performance improves over the scaled down, equivalent swarm performance. These are therefore split into two metrics for this topic. Both terms are used to mean the same thing in the literature, often interchangeably in the same text, but when referring to these two separate metrics, as given in this work, they should be defined as:

- If m agents out of N total have failed by Failure Mode X, then good **Robustness** occurs when the percentage change in performance is less than the percentage change in agents (m/N).
- If m agents out of N total have failed by Failure Mode X, then good **Fault Tolerance** occurs when the percentage change in performance due to faults is less than the percentage change in performance of a working swarm with N − m agents.

Ability to cope with changing swarm size

The second swarm behaviour trait that requires a name to be used going forward is:

***(2)** The ability to cope with changing swarm size*

This is defined by many sources to be scalability [4, 6, 10, 2, 7, 12, 9, 11]. How much change in performance is allowed for a system to be deemed scalable differs by source. Dorgio et al. (2021) and Şahin (2004) only require the performance to remain the same ("function properly" - [4]) under differing group sizes, for a scalable system. Hamann and Reina (2022) state that "a scalable system has increasing performance with increasing swarm size", a higher requirement. Hamann and Reina (2022), and other sources [2, 9], continue on their definition of scalability in swarms to include, but not require, superlinear scalability or speedup as occurring when additional agents provide more performance per agent than the previous. If PN is the initial performance with N agents, then for superlinear scalability, PN+1 > PN + PN /N. The metric given for scalability by this work defines scalability as being a positive change in performance due to a change in group size but it also has a scalar value, which is proportional to the change in performance per change 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.

group size. Therefore, reading the metric value will indicate both scalability by ̧Sahin (2004) and the other definitions, but also indicate when the swarm has superlinear scalability, according to Hamann and Reina (2022). The definition of the term is therefore:

- A system is **Scalable** when the performance change is positive with increasing swarm size. The system is **Superlinearly Scalable** when the performance change per agent is superlinear.

Ability to cope with changing external parameters

The final swarm behaviour trait is:

> ***(3)*** *The ability to cope with changing external parameters (environmental or task parameters).*

This does not include the ability to be applied to completely different tasks or environments, only parameter changes within the original task or environment. A common term for **(3)** is particularly difficult to find, compared to the other two traits discussed. Brambilla et al. (2013) would define this in its most simple terms as *flexibility*. However, contrary definitions from S ̧ahin (2004) and Dorigo et al. (2021) would define the term *flexibility* only by the ability to switch tasks with no mention of environmental factors [10, 4] and Brambilla et al. (2013) also includes this in their broad definition for flexibility [6]. Both Harwell and Gini (2019) and Hecker and Moses (2015) define **(3)** as *flexibility*, focusing on differing external environmental factors without mention of differing tasks. Harwell and Gini (2019) go further to define *adaptability* as an axis of their flexibility metric, where adaptability is "ability to minimise performance losses under adverse conditions and proportionally exploit beneficial deviations from ideal conditions" [9]. However, Dorigo et al. (2021) define the ability to "continue to work efficiently in environmental conditions different from those considered at design time" [4] as *robustness*. This is a confusing term to use because they define this as capacity to work with changing environmental conditions but most others found in the literature would define robustness as to do with robot failures [10, 9, 6, 5, 11, 13]. Dorigo et al. (2021) would define *adaptivity* or *adaptability* (both terms are used in their work), as the ability to change behaviour to new operating conditions such as obstacles or changing atmospheric conditions [4]. Other literature [11, 13] also use the term *adaptability* to describe the response to changing external factors.

The term *flexibility* has been defined for both flexibility to work in different environmental changes and flexibility to work on different tasks altogether. It could be a confusing term if used going forward, as changing task is not discussed in the SPI used here. Therefore, the term *adaptability* has been used to avoid confusion, particularly as definitions from Hecker and Moses (2015) and Dorigo et al. (2021) of *adaptability* most closely matches **(3)**.

- A system is **Adaptable** to changes in external parameter, x, if any performance lost is less than the proportional change in parameter, x. Where x can be an environmental factor (e.g. number of obstacles, brightness of light in the space) or a task parameter (e.g. size of warehouse space, speed of agents).

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 Performance Indicators

These Swarm Performance Indicators are metrics to describe traits seen in swarm behaviours, with their terminology reasoned using current literature on the topic. The metrics defined here are intended to improve the measurability of swarm behaviours. When we want to know how a system is fairing under faults or changes to the external parameters or changing swarm size, what the user is asking is how are these factors affecting the performance of the system. One cannot ask only how performance has differed as the difference from one condition to another (e.g. there is an increase in time taken of 10 seconds) because this is not enough context. First, it is important to know how much the performance has changed in proportion to the usual performance of the system. This is most easily described by a percentage change in performance. For example, 'there has been a 10% loss in performance due to faults in the system' is more informative at a glance than 'the performance has gone from 100s to 90s when faults are present'. Secondly, how much the system has altered to cause this performance change is an important factor in determining the effect of the change on the system. If we know that the system has lost 10% of its original performance due to faults, then it is necessary to know more details about the magnitude of these faults. For example, if the system lost 10% of performance when 1% of agents failed, then this is a worse result than if the same performance loss occurred when 80% of agents have failed. The 10% looks bad when the agent loss percentage is low and it looks good when the agent loss percentage is high.

Scalability metric

The swarm trait that the Scalability metric, S, describes is:

The ability to cope with changing swarm size

This was defined following a literature review as:

A system is *Scalable* when the performance change is positive with changing swarm size. The system is *superlinearly scalable* when the performance change per agent is superlinear.

To measure Scalability as intended here, the performance should be measured for a given system with N agents and then again with $N + m$ agents. The conditions and all other parameters between the two swarm size tests should be identical. The density of agents in the space will change between the two systems and the size of the space should not be adjusted to keep density constant. As a result, density and the available space for agent movement are factors which contribute to the Scalability metric result, rather than factors to be eliminated from the discussion. This is intentional and intended to test the scalability of a real swarm in real-life usage, where the available space cannot necessarily change and is a factor in how scalable the swarm behaviour is. For example, if the user has a 5 m x 5 m space to use their swarm, then they may find that the swarm performance is much worse going from 50 to 60 agents because they do not have enough space to move around anymore. In theory, with adjustable space size, the swarm may perform better with 60 than 50 agents but this information is not useful to a user who only has a limited space and wants to know if they can scale up their swarm to improve performance. For this reason, when quoting the Scalability metric result it is important to also state the conditions which this result was found such as the size of the swarm space and the radius of the agents.

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.

Definition for scalability

In the literature, scalability is defined as being acceptably true when no performance is lost by increasing swarm size [4], [10], which is Condition 1:

$$P_{N+m} \geq P_N$$

and as being good (superlinear) when the new performance is increased by more than the performance per agent [2, 9]. This is, Condition 2:

$$P_{N+m} > P_N + \frac{m * P_N}{N}$$

Derivation of Scalability metric, S

In a scalable system that moves from N agents to N + m agents, the performance at the two swarm sizes follows Condition 1: $P_{N+m} > P_N$. In a superlinear scalable system going from N to N+m agents, the performance of the larger system satisfies Condition 2: $P_{N+m} > P_N + (m * P_N / N)$. Therefore, Scalability, S has the following derivation where the original swarm performance P_N with N agents is changed to P_{N+m} by the addition of m agents:

$$P_{N+m} > P_N + \frac{m * P_N}{N}$$

$$P_{N+m} - P_N > \frac{m * P_N}{N}$$

$$\frac{P_{N+m} - P_N}{P_N} > \frac{m}{N}$$

$$\frac{\frac{P_{N+m} - P_N}{P_N}}{\frac{m}{N}} > 1$$

$$S = \frac{\frac{P_{N+m} - P_N}{P_N}}{\frac{m}{N}}$$

Such that a scalable system is **S > 0**, which satisfies Condition 1 and a superlinear scalable system satisfies Condition 2 when **S > 1**. Simplified, this is,

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.

$$\%\Delta P = \frac{P_{N+m} - P_N}{P_N} \quad (1)$$

$$\%\Delta N = \frac{m}{N} \quad (2)$$

$$S = \frac{\%\Delta P}{\%\Delta N} \quad (3)$$

Fault Tolerance and Robustness metrics

The swarm trait that these metrics are describing is:

The ability to cope with faulty agents in the system

The effect of faults on the system has been described in multiple ways within in the literature (see Section 5.2). To include the different elements of the effect of faults, two different metrics are proposed: Fault Tolerance and Robustness.

To formally test the effect of faults on a system and use these metrics as they were intended, it is necessary to set-up experiments in the following way. Two performances should be measured: the original performance without faults; and the performance where faults have occurred. The performance with faults should have identical parameters and set-up to the case without faults. Ideally, the faults would be simulated by the user for better control over these parameters. Only one failure mode should be tested at a time so that the metric is clearly specific to that set of parameters, failure mode and number of failures. This supports research into fault tolerance metrics by Bjerknes and Winfield (2013).

Fault Tolerance metric

It is important to examine if the faulty agents are acting as an anchor [5] or are still being useful to the swarm. As an anchor, faulty agents cause their working neighbours to perform less well despite having no fault of their own. Conversely, a faulty agent may still be able to contribute to the system depending on the extent of their fault. For example, if their wheels have failed causing them to be static obstacles, they may still be able to communicate sensory information, which may help the task more than if they were just removed from the system when they failed. This is examined by this metric, which compares the performance of the equivalent, scaled down swarm performance, P_S, to the failed swarm. For P_S, m agents have been removed from the swarm space. In the failed swarm, m agents have failed and remain in the swarm space. If the failed swarm performs better than the scaled down swarm, then there is an indication that the swarm is tolerant to the fault X with m failed agents because it performs better than if those agents are simply removed altogether.

Definition of Fault Tolerance

The definition for Fault Tolerance that was reasoned in the literature review, is:

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.

If m agents out of N total have failed by Failure Mode X, then good *Fault Tolerance* occurs when the performance change under faults ($\%\Delta P_F$) is better than the performance ($\%\Delta P_S$) of a working swarm with N-m agents.

Therefore, in a fault tolerant system,

$$\%\Delta P_F > \%\Delta P_S$$

Where $\%\Delta P_S$ and $\%\Delta P_F$ are calculated as follows where P_{N-m} is the performance with $N - m$ agents and P_F is the performance with m faulty agents out of N total. P_o is the original performance with no failures and N working agents.

$$\%\Delta P_S = \frac{P_{N-m} - P_o}{P_o}$$

$$\%\Delta P_F = \frac{P_F - P_o}{P_o}$$

Derivation of Fault Tolerance metric, FT

A fault tolerant system is $\%\Delta P_F > \%\Delta P_S$. Therefore, Fault Tolerance, FT can be measured as,

$$\%\Delta P_F - \%\Delta P_S > 0$$

$$FT = \%\Delta P_F - \%\Delta P_S \quad (4)$$

Where **FT > 0** is a fault tolerant system.

Robustness metric

For a system to be counted as robust it is permissible for some performance to be lost when some agents are faulty [10, 4]. How much loss of performance is acceptable has not been defined [5] but a reasonable suggestion is given here as a metric for robustness. Dorigo et al. (2021) describes a "graceful degradation" in performance due to faults. Therefore, in this metric, we determine that a system is robust to a fault if that faulty agent does not cause a negative reaction in its neighbouring agents. This is similar to the anchor effect described in [5]. A system is robust if the faulty agent only removes their own performance contribution. The definition of Robustness is therefore as follows.

Definition of Robustness

The definition for Robustness that was reasoned in Section 5.2 is:

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.

If *m* agents out of *N* total have failed by Failure Mode X, then good **Robustness** occurs when the percentage change in performance is less than the percentage change in agents (*m*/*N*).

A system is robust if the performance P_F with *m* faulty agents out of *N* total is,

$$P_F > P_o - \frac{m * P_o}{N}$$

where P_o is the performance without faults. Otherwise, if the faulty agents cause more performance loss than agent loss then the system is not robust.

Derivation of Robustness metric, R

Following the definition, Equation 5, given for Robustness, the metric R can be derived as follows

:

$$P_F > P_o - \frac{m * P_o}{N} \tag{5}$$

$$P_F - P_o > -\frac{m * P_o}{N}$$

$$\frac{P_F - P_o}{P_o} > \frac{-m}{N}$$

$$\frac{P_F - P_o}{P_o} - \frac{-m}{N} > 0$$

$$\frac{P_F - P_o}{P_o} + \frac{m}{N} > 0$$

Which can be simplified to,

$$\%\Delta P + \%\Delta N > 0$$

$$\%\Delta P = \frac{P_F - P_o}{P_o} \tag{6}$$

$$\%\Delta N = \frac{m}{N} \tag{7}$$

Such that,

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.

$$R = \%\Delta P + \%\Delta N \quad (8)$$

Where **R > 0** is robust.

Adaptability metric

The swarm trait this metric describes is,

The ability to cope with changing external parameters.

This was defined following a literature review as:

A system is *adaptable* to changes in external parameter, *x*, if any performance lost %Δ*P* is less than the proportional change in parameter *x*, %Δ*x*.

To measure this metric as it is intended in this work, it is necessary to test the following two cases of the given system. The two cases should have identical parameters and conditions but with one particular parameter of interest different between the two. This parameter could be either environmental (e.g. number of obstacles, level of brightness) or task based (e.g. number of targets to find, task time limit). The term *adaptability* has been used here to define this trait but it should be noted that there is not a unifying term in the literature for this behaviour, see Section 5.2. The exact boundaries of the term's meaning as it is used for the metric are given in the following definition.

Definition of Adaptability

A system is adaptable to changes in parameter x if the performance improves from the original performance P_o. This is Condition 1,

$$P_{new} > P_o$$

If this is not true and performance is lost when moving from x_o to x_{new}, then the system is only adaptable if the change in performance is proportional to the modulus of the change in x. The modulus is used because it is irrelevant if parameter x increases or decreases. This is Condition 2,

$$P_{new} > P_o - P_o * |\frac{x_{new} - x_o}{x_o}|$$

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.

Derivation of Adaptability metric, A

A system is adaptable when,

$$P_{new} > P_o - P_o * |\frac{x_{new} - x_o}{x_o}|$$

$$\frac{P_{new} - P_o}{P_o} > -|\frac{x_{new} - x_o}{x_o}|$$

$$\frac{\frac{P_{new} - P_o}{P_o}}{|\frac{x_{new} - x_o}{x_o}|} > -1$$

Which simplifies to,

$$\%\Delta P = \frac{P_{new} - P_o}{P_o} \tag{9}$$

$$\%\Delta x = \frac{x_{new} - x_o}{x_o} \tag{10}$$

Which gives A as,

$$A = \frac{\%\Delta P}{|\%\Delta x|} + 1 \tag{11}$$

The system is adaptable if **A > 0** because Condition 2 is satisfied. The Adaptability is very good when **A > 1** as Condition 1 is satisfied, meaning that performance has been gained.

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.

Summary of metrics

Scalability	**The ability to cope with changing swarm size.**	
	A system is scalable if, $P_{N+m} > P_N$ A system is superlinearly scalable if, $P_{N+m} > P_N + \frac{m * P_N}{N}$	$S = \frac{\%\Delta P}{\%\Delta N}$ $S > 0$ is scalable $S > 1$ is superlinearly scalable
	P_{N+m} is performance with $N+m$ agents and P_N is with N agents. $\%\Delta P$ is percentage change in performance when moving from m to $N+m$ agents. $\%\Delta N$ is percentage change in swarm size.	
Adaptability	**The ability to cope with changing external (environment or task) parameters.**	
	A system is adaptable if, $P_{new} > P_o - P_o * \lvert\frac{x_{new} - x_o}{x_o}\rvert$ when moving parameter changes from x_o to x_{new}	$A = \frac{\%\Delta P}{\lvert\%\Delta x\rvert} + 1$ $A > 0$ is adaptable
	P_{new} is the performance with the parameter x_{new} P_o is the original performance with the original parameter x_o $\%\Delta P$ is the percentage change in performance from original to new. $\%\Delta x$ is the percentage change in the parameter of interest, x	
Fault Tolerance	**The ability to cope with faulty agents in the system.**	
	A system is fault tolerant if, $\%\Delta P_F > \%\Delta P_S$	$FT = \%\Delta P_F - \%\Delta P_S$ $FT > 0$ is fault tolerant .
	$\%\Delta P_F$ is percentage change in performance under m faults $\%\Delta P_S$ is percentage change in performance with a working swarm of $N-m$ agents.	
Robustness	A system is robust if, $P_F > P_o - \frac{m * P_o}{N}$	$R = \%\Delta P + \%\Delta N$ $R > 0$ is robust.
	P_F is performance under m faults out of N agents. P_o is performance without faults, with N agents. $\%\Delta P = \frac{P_F - P_o}{P_o}$ $\%\Delta N = \frac{m}{N}$	

Table 1: Summary of the Swarm Performance Indicators

Results of metric testing

The Swarm Performance Indicators are tested here on two use cases which use distributed control algorithms. All experiments are simulated in a 2D Python simulator. The first use case is a logistics use case where the swarm is tasked with retrieving items from a storage space. The second is a decision making algorithm. The aim of the task is for the swarm to make a collective decision about which is the best of two sites, A and B, of various qualities. These use cases were chosen for testing because they would benefit from swarm trait specifications to understand how best to apply them to real-world use. In both use cases the area is entirely

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.

unmapped and the agents only have local communication and sensory information. In both cases, the control of the swarm is entirely distributed, with each agent only sensing their local environment and only communicating with neighbours in their limited sensory range.

How the results will be structured

Scalability

To test the scalability of the use case systems from N to $N + m$ agents, the arena size remained the same throughout. The scalability is measured in two ways, both using the Equation 3 as the metric for scalability. The first is *incremental scalability*, which is the scalability for each jump in swarm size measured e.g. S_{5to10} is scalability of going from 5 to 10 agents, then S is measured again for the system increasing from 10 to 15 agents, S_{10to15}. Then the scalability is measured for *increasing swarm sizes from x agents*. Here, each increase gives the scalability S_{xtoN} when the scalability is increased from x agents to N agents, tested for a group of N values while keeping x the same.

Fault tolerance and Robustness

To test the fault tolerance (FT, Equation 4) and Robustness (R, Equation 8), different failure modes are tested. For each use case, a series of failure modes are chosen and tested that are specific to the use case. Then a number of agents are failed, increasing from 1. For each failure mode, all other parameters are kept the same to compare them.

When discussing fault tolerance or robustness it is important to note that the metric value is a guide and further depth of analysis and justification should always be taken and given alongside the metric value. The classification is Boolean, which can be reductionist e.g. 'not Robust' if $R < 0$, but *what* the system is robust to should always be clearly stated alongside the metric, to allow for complexities and to clarify the data. For example, when stating if a system is robust or not it should be accompanied by a clear description of the experimental set-up, the parameters tested and the failure mode used. The user should state e.g. 'System A is robust up to an agent loss of y%, to failure mode B, in the experimental set-up C using control algorithm D'. Here, the agent loss that it is robust to is the highest number of agents that can fail by B while $R > 0$.

Adaptability

To test the Adaptability, different examples are used for parameter x in A, Equation 11. The extent of the Adaptability of the system to changes in the chosen parameter is measured by varying x.

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.

Use case example: Logistics scenario

This use case is analogous to a logistics scenario such as a warehouse full of items for delivery upon incoming order. Here the task is to deliver an individual and specific item to a delivery area. There are multiple unique and randomly spaced items with no inventory list or area map. The area of interest - which the agents must find when they find the requested item - is the delivery area. Here, in a real, extended version of this scenario, the items are collected by human pickers who would pack the item for shipping.

Experimental set-up

The **performance** is measured as the number of items delivered in the time limit. Each experiment is run for 100 trials and the performance for that given set of parameters is the average of these trials. Figure 9 is a screen-grab of the same simulated experiment at two different times, 4 s and 28 s. The earlier time point shows the items in their starting positions in the inlet area (although one item has already been moved out by an agent). The inlet area is always 20% of the warehouse width wide and 100% of the warehouse depth ('height' on the diagram) long. It is located at the far left hand side of the warehouse as viewed from above, as it is in Figure 9. The items always start the experiment randomly distributed in the inlet area, and the agents start in the not-inlet area (i.e. anywhere in the warehouse space except the inlet area). Also shown in Figure 9 is the requested item in black, where all other items appear in red. Items are delivered when they are carried to a point where warehouse width is greater than 800 cm (into the delivery area) which is indicated by the dotted line in the image. Finally, the timer values, which the agents display as an indication to other agents of how recently they have been in the delivery area, are also shown alongside their respective agent as a number between 0.0 and 500.0 (< 0.0 are not shown or communicated to neighbours). Unless otherwise stated, the parameters for each experiment set-up are given in Table 2.

The task the agents are performing is delivery of a requested item. All items have a unique identity number that the agents can read when they are within a certain range of the item. The only global information given to the agents is the label of the requested item for delivery, which they are all searching for simultaneously. This is updated to every agent when it is delivered and a new item is requested.

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.

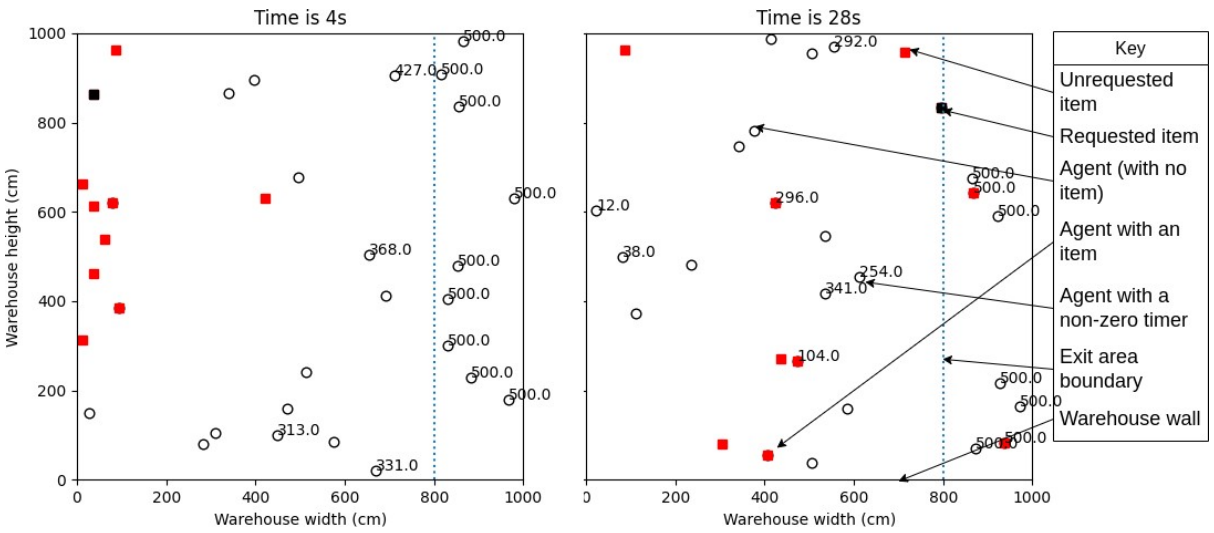

Figure 9: Screen grabs taken from the swarm simulation of the logistics use case at time stamps 4 s and 28 s from the start of the experiment. Experiment and warehouse elements are labelled in the Key.

Parameter name	Value	Unit
Warehouse width	10	m
Warehouse depth	10	m
Time limit	500	s
Number of items	60	items
Delivery area width	2.0	m
Delivery area depth	10	m
Inlet area width	1.0	m
Inlet area depth	10	m
Agent speed	1.0	m/s
Collision avoidance	0.35	m
Item detect range	0.75	m
Communication range	5.0	m
Agent diameter	0.25	m
Item diameter	0.25	m
Time step	0.02	s

Table 2: Experiment parameters for the logistics use case. The agents and warehouse dimensions are based on the Toshiba DOTS [15].

Algorithm

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 agents in this case study are based on the Toshiba DOTS, developed at the Bristol Robotics Laboratory [15] (although it should be noted that the DOTS have more hardware and capabilities than the agents simulated in these experiments). Each agent has the following (simulated) hardware: holonomic wheel configuration; lifting mechanism for items; camera and IR sensor to detect items and obstacles; sensor to communicate with other agents. Using this onboard equipment, each agent can perform the following behaviours every time step: Random Walk; Collision avoidance; Item detection, pick up and put down; Periodic reshuffling of items; Delivery area detection, timer broadcast and item delivery; Swarm Diffusion-Taxis algorithm [16].

The **random walk** behaviour is updated every 1 second (once every 50 time steps). The agents move forward at a constant speed of 1 m/s for 1 second and then change to a new random direction of movement, adding $-0.5^c < \alpha_{random} < 0.5^c$ to the current direction of travel, α. This is modelled as an instantaneous change in direction. The agents follow this random walk unless something comes into their sensory range. If they come within 0.35 m (centre-centre) of an obstacle then their **collision avoidance** behaviour is triggered. If the obstacle is a wall then the agent adds a quarter turn to their heading direction ($\alpha = \alpha + \pi/2$) until they are moving away from the wall. If the obstacle is another agent or an item (which they want to avoid because they have an item currently) then the agent will move in the opposite direction from that obstacle. If there are multiple obstacles to avoid then the distances and directions to each obstacle sum to a vector, which the agent moves along to avoid them. **Item detection, pick up and put down:** When an agent that is not currently carrying an item comes into sensory range of an item (0.75 m from agent centre) then the agent will pick up the item. The items are on table-like carriers, which raise them off the ground on stilts. The agents can navigate underneath an item they have found and lift it up from beneath to carry it around. This is based on how items are stored and collected in the Toshiba test-bed, which simulates warehouse scenarios [15]. The items are **periodically reshuffled** by the agents which will carry items around and put them down again somewhere else if they are not the requested item. They generate a random number between 0 and 100 every time step and if it is below 2 then they drop their item where they are and leave it behind for another agent to pick up. This reshuffling avoids two deadlock cases: (1) the requested item is trapped behind unrequested items; (2) all the agents are carrying unrequested items and are unable to deliver them or pick up new items. **Delivery area detection, timer broadcast and item delivery:** When an agent arrives in the delivery area it receives a (simulated) signal from a beacon there to say it is in the delivery area. It can then deliver the item it is carrying if it is the requested item. It will also broadcast a time value when it has been in the delivery area, which follows the **Swarm Diffusion-Taxis algorithm** [16]. In the SDT algorithm, the timer value that the agent broadcasts to its local neighbours is maximum (500) when the agent is in the delivery area. When it is outside the delivery area, the timer value decays by 1 every time steps until it is NaN after 10 seconds. When the agent has the requested item, it will read the timer values of neighbours within its communication range (5.0 m) and move towards the agent with the highest timer value. It will re-do this step every time step.

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.

Scalability results

The performance was measured for different swarm sizes and the results for performance and Scalability are given in Figure 10 and expanded in this Section.

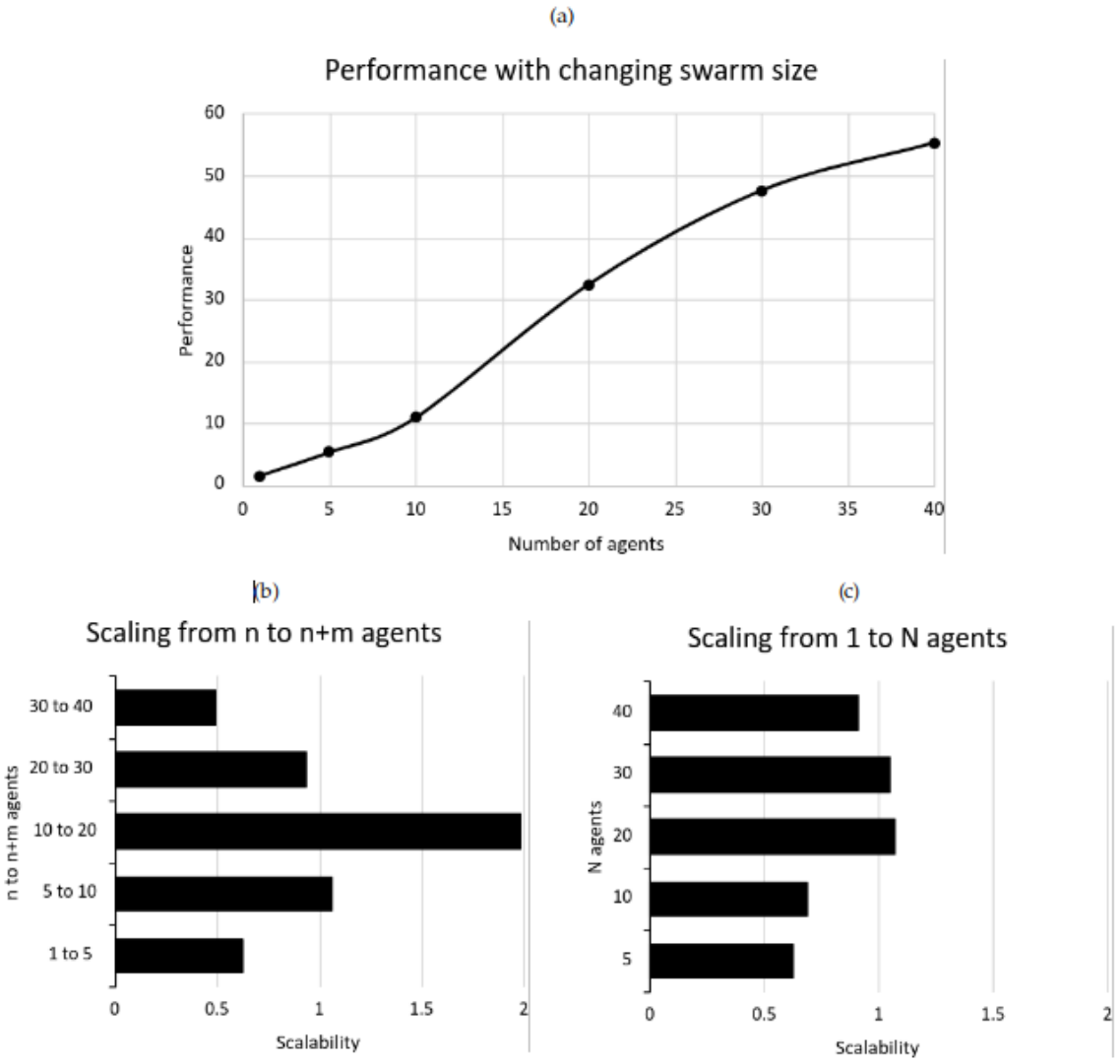

Figure 10: Scalability, S (Equation 3) and performance (number of items delivered in time limit) for logistics use case ***(a)*** *Performance data for various swarm sizes 1-40 agents* ***(b)*** *Scalability measured for incremental changes in swarm size* ***(c)*** *Scalability measured from 1 agent to N agents.*

Incremental Scalability

Equation 3 was used to measure the Scalability of incremental increases in swarm size. For Equations 1, 2 and 3, m = 5 for 1 to 10 agents and m = 10 for 10 to 40 agents. The results are

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.

given in Figure 10(b). The Scalability values are all S > 0, which indicates that they are all scalable. This means that there is no decrease in performance due to an increase in agents, at any agent number tested. This can be confirmed by observing the performance curve in Figure 10(a). The results are superlinearly scalable, S > 1, for 5 to 20 agents, which indicates that the performance (percentage change) increases more than the number of agents added (percentage change), which is a superlinear increase in performance. Considering the performance values in Figure 10(a), this matches the shape of the curve because either side of this region, the gradient is less steep.

Scalability from 1 agent

Scalability is also measured for the change in performance from 1 agent. In every case of S, m is increasing, $N = 1$, $P_N = P_1$, $P_{N+m} = P_{1+m}$. The results for Scalability are given in Figure 10(c). The Scalability values are all scalable, with S > 0. This makes sense because all the performances in Figure 10(a) increase from P_1. 1 to 20 and 1 to 30 agents are superlinear Scalability ranges, S > 1. From this, it can be concluded that the maximum super scalable range is 1 to 30 agents and the maximum scalable range is 1 to 40 agents (the maximum tested swarm size).

Specification for Scalability

The user can use this information in the following ways. If the **Incremental Scalability** results were included in a specification then the user could look up the Scalability for a given range of agent numbers. For example, if the user was working with this set-up with 10 agents and they wanted to improve the performance then they could look up Figure 10(b). From this graph they could read that scaling from 10 to 20 agents will give them a super scalable result, meaning that it will be a performance per agent increase that is more than the cost per agent increase. Whereas, if they had a swarm of 20 agents and they looked up what the Scalability was for moving to 30 agents they would find that 0 < S < 1, which is a scalable but not superlinearly scalable result. They would therefore know that they would get a performance increase but it would not be an increase with good efficiency. How they would proceed would depend if they valued performance or efficiency more and what resources they had available to them.

If the **Scalability from 1** results were included in a specification then this could be used to decide the best swarm size for the user to use, depending on how many agents they have available or have the resources for. If a user had unlimited agents and the same experimental set-up as is used for the results in Figure 10(c), then they could look at this graph and see that they would get the most efficient performance per agent increase (from 1 agent) at 20 agents. From the performance data (Figure 10(a)) they can see that this is not the best performance possible. But if the user values efficiency then 20 agents would be their best choice. Either Incremental Scalability or the Scalability from 1 can also be used to compare one swarm's Scalability to another. For example, if both swarms have N agents and perform the same task in the same set-up then the swarm with the highest S_{1toN} number would be the most scalable.

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.

Fault Tolerance and Robustness results

The performance of the system under 4 different failure modes is measured and given in Figure 11(a). In part (a) of this Figure, the scaled down swarm performance P(SD) and the proportional change P(%N) in performance are also given alongside the performances under faults 1-4. P(SD) is very similar to P(%N), which means that the results for Fault Tolerance (FT) and Robustness (R) are likely to be very similar to each other. A swarm of 25 agents was used for these experiments. The failure modes tested were as follows:

- **Failure Mode 1 (FM1):** Malicious agents. The delivery area timer value is wrong. For failed agents the timer is always set to 500, the maximum value, so that they look to their neighbours as if they are always in the delivery area.
- **Failure Mode 2 (FM2):** Failed agents cannot pick up items and they will instead treat them as obstacles.
- **Failure mode 3 (FM3):** Delivery area timer value always set to 0. Failed agents cannot broadcast how recently they have been in the delivery area but they can still detect the delivery area when they are in it to deliver items they carry.
- **Failure mode 4 (FM4):** Failed agents cannot deliver items. In the reshuffling behaviour, items are not dropped if they are the requested item. This means that with this failure mode, the requested item will never be passed on to a working agent if it is carried by a failed agent in this failure mode. Therefore, if an agent has failed in this way and is carrying the requested item then the scenario will deadlock and no more items will be delivered.

Fault Tolerance results

The results for Fault Tolerance at each failure mode (FM) are given in Figure 11(b), alongside the performance data in Figure 11(a). The system is fault tolerant to FM3 up to and including 10 agent failures. Looking at the performance data for FM3, this is clearly true as the average performance increases for some of the results with increasing faulty agents and is always high. The system is fault tolerant to FM1 for 1, 2, 4-10 failed agents. The result where it is not fault tolerant , m = 3 agents, could be accounted for by the general variation in performances for P(FM1) and P(SD), shown in the performance data.

The system is fault tolerant to FM2 for 1-2, 4, 6-8 agent failures. The magnitude of the FT results for FM2 are small compared to the other failure modes at almost all numbers of faulty agents tested, because the performance trend is close to the scaled down performance (seen in Figure 11(a)).

The system is not fault tolerant to FM4 at any number of faulty agents. The larger magnitude of FT (with negative sign) indicates the catastrophic failure caused by this FM from 2 failed agents and up, seen in the performance data for FM4.

Robustness results

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.

As expected, the values for Fault Tolerance, FT (Figure 11(b)) and Robustness, R (Figure 11(c)) are very similar in this case, as was predicted given the performance data for the scaled down swarm. The system is robust to all 4 failure modes tested when 1 agent has failed, as all have results R > 0. The system remains robust to both FM1 and FM3 up to and including 10 agent failures. The system is robust to FM2 up to 5 agent failures and not robust for 6-10 agent failures. Finally, the system is not robust to FM4 for 2 and above agent failures, which also reflects the catastrophic failure caused by this FM.

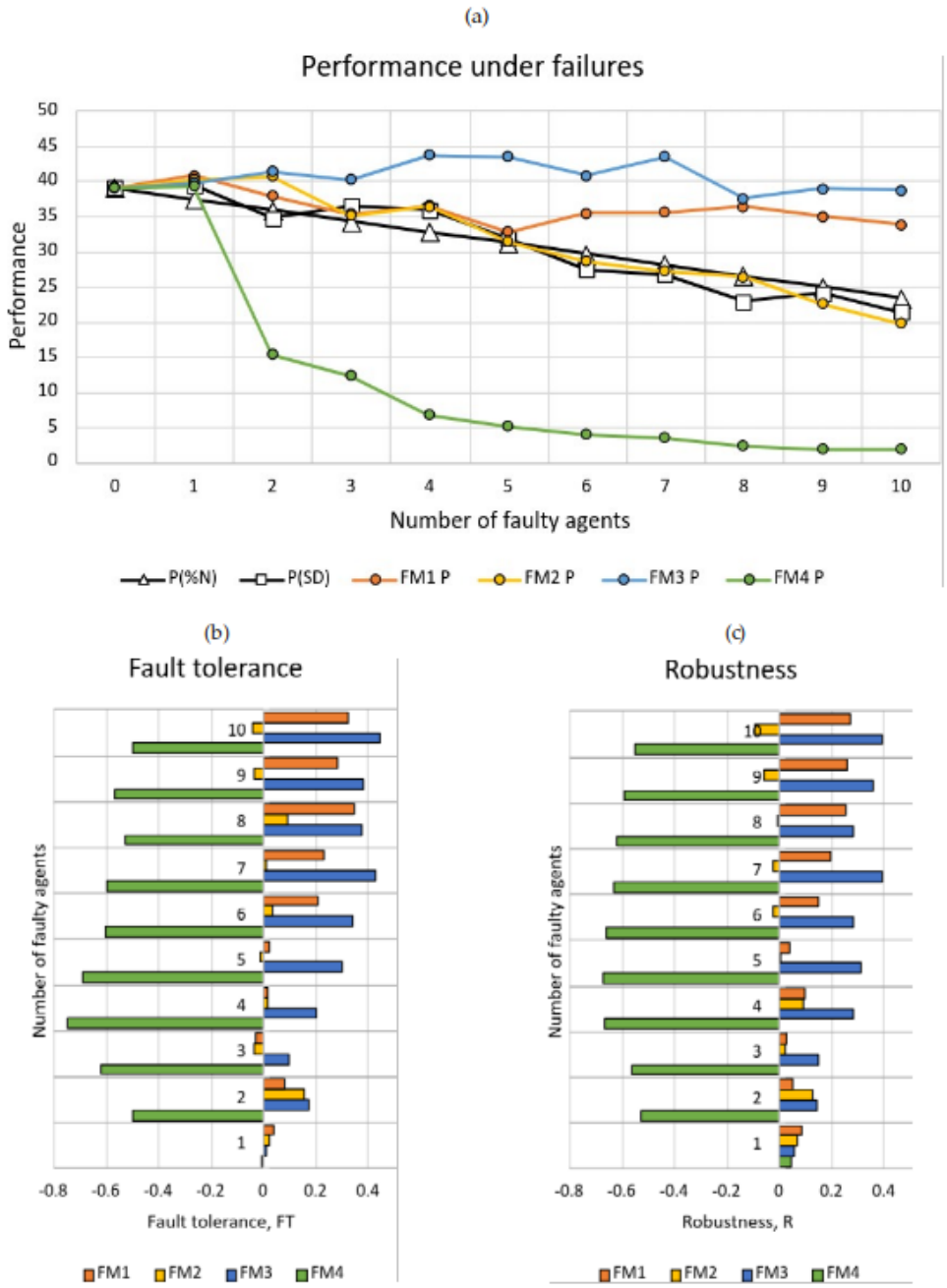

Figure 11: Performance (items delivered in time limit) with different numbers of faulty agents, m, for the logistics use case and Fault Tolerance, FT (Equation 4) and Robustness, R (Equation 8) results. ***(a)*** *Performance data with m agents failed by each failure mode (FM 1-4) and an equivalent scaled down swarm with 25 − m agents (P(SD)) and P(%N), which is the performance if it is reduced by the same*

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.

proportion as agents lost. If FM-P is above P(SD) at a given m then it is a fault tolerant result, if it is above P(%N) then this is a robust result ***(b)*** *FT results* ***(c)*** *R results.*

Adaptability results

The Adaptability between two conditions was measured for three different types of condition change. These were: warehouse width; delivery area width; and number of items. These are used as parameter *x* in Equation 5.11. For all three parameter experiments, the swarm size used was 25 agents and the rest of the parameters are given in Table 2 unless otherwise stated.

Adaptability to changing warehouse width

Warehouse width was varied over a series of experiments to measure how adaptable the system was to changing warehouse size. This information is useful to a user hoping to use the system with their given swarm size and number of items, when they need to make decisions about what space to give the agents. It is unclear without testing what size space would be optimum for a given system and how much performance might be lost or gained by increasing or decreasing that space. The available space may change in a modular, pop-up storage space and how adaptable the system is to this change is important to know.

The depth of the warehouse was kept at 10 m in each experiment and the delivery area width is always 20% of the warehouse width. Performance data is given in Figure 12(a), where the performances measured all decrease as warehouse width increases from 500 cm. Therefore none of the results for the widths tested are adaptable, in Figure 12(b). It is difficult to tell from the performance graph alone how the proportional changes with performance compare to proportional changes in the parameter, warehouse width, so looking at the Adaptability metric results is necessary for further analysis. The system is adaptable (A > 0) from 500cm to 2000cm (the maximum width tested). The system is most adaptable from 500 cm to 1000 cm, which has the highest A value. The values for 500-1500 cm and 500-2000 cm are very similar Adaptability values.

The performance at 2000 cm is 8 items compared to 47 items at 500 cm. This is a big drop in performance but the system is still defined as "adaptable" by this given metric because it measures the proportional changes in performance and width. In this case, the performance changes proportionally to the change in width, which is defined as adaptable following the reasoning and definition given previously.

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.

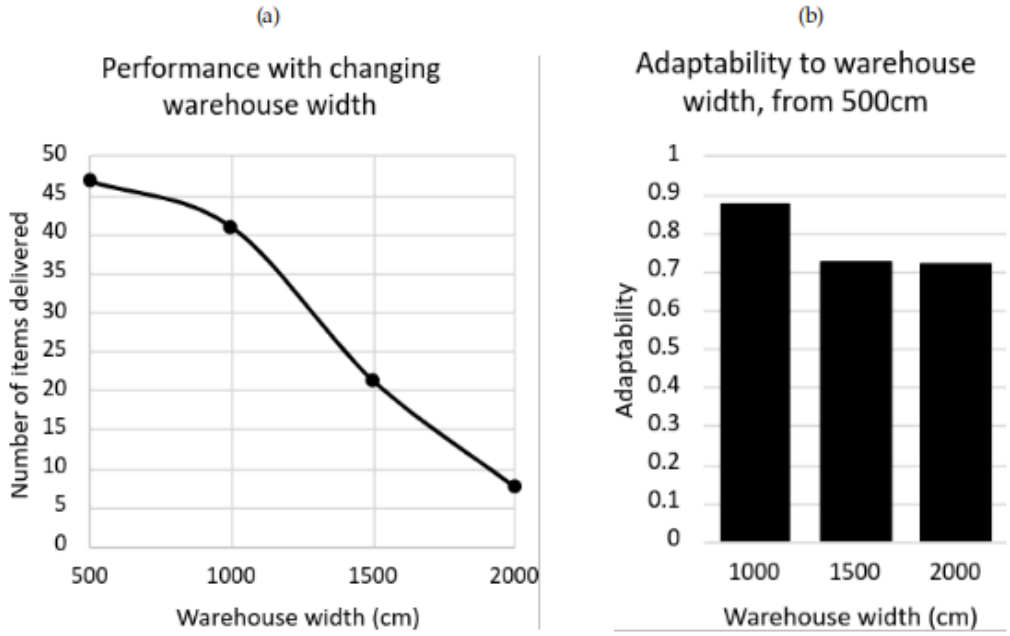

Figure 12: Performance (number of items delivered in time limit) and Adaptability, A (Equation 11) to various warehouse widths. ***(a)*** *Performance data for warehouse widths 500-2000 cm* ***(b)*** *Adaptability metric, A, results, comparing performances from (a) to x_o = 500 cm.*

Adaptability to changing delivery area width

The size of the delivery area affects the system in the logistics scenario in two ways. The first is that it is more likely that a random walker will come across the delivery area if it is larger, which means that the performance is likely to increase with increased delivery area size. However, a smaller delivery area is more useful to a user of the system because the delivery to the picker- packer is more efficient and the system is more useful. It also means that the storage space can be larger and more items can be stored. Additionally, the signal source used to indicate the delivery area cannot necessarily have a far reach. This experiment set tests how adaptable the system is to changing delivery area width, particularly decreases in delivery area width.

The Adaptability to changing delivery area size in a 10 m x 10 m warehouse is tested here. The depth of the delivery area is kept constant at 10 m (the depth of the warehouse). The performance data for different delivery areas is given in Figure 13(a). The Adaptability metric results are given in Figure 13(b) where A is measured by taking the original delivery area width as 200 cm, so that in each case x_0 = 200 cm. The performance increases when delivery area width is increased 200 cm to 250 cm and decreases when delivery area width decreases for all results, 200 cm to 50 cm. The performance increases are reflected in the Adaptability value for 200 cm to 250 cm, which is A > 1. All other results reflect the decrease in performance (A < 1) but all are still found to be adaptable, A > 0, down to and including delivery area width = 50 cm, compared to 200 cm.

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.

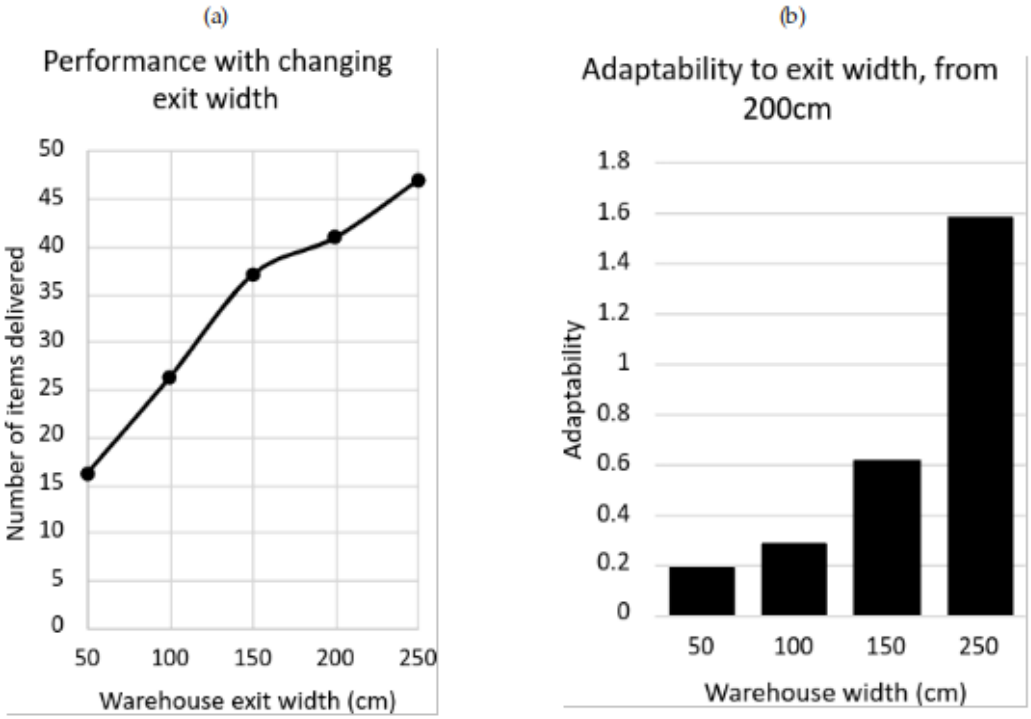

Figure 13: Performance (number of items delivered in time limit) and Adaptability, A (Equation 11) to various delivery area widths. ***(a)*** *Performance data for 50-250 cm delivery area widths* ***(b)*** *Adaptability metric, A, results, comparing delivery area widths from (a) to* x_o *= 200 cm.*

Adaptability to changing number of items

When items are delivered they are removed from the warehouse space and no more items are added to the warehouse during the experiment. This can change the performance of the swarm because if the items are numerous then they can occlude each other and it can take longer to find the requested item. The probabilistic reshuffling behaviour is used to avoid deadlocks caused by trapped items but progress can still be slowed. How adaptable the system is to changes in the number of items is measured here.

The Adaptability is measured between x_0 = 60 and x items. The results for performances at different numbers of items, Figure 14(a), show that the performance decreases as the numbers of items increase from 60. Adaptability metric results from Figure 14(b) show that increases from 60 to 90 items are not adaptable (A < 0) whereas increases from 60 to 100-120 items are adaptable (A > 0). For x=70-90 items, the proportional change in performance is greater than the change in number of items from 60 to x (e.g. $\%\Delta x = (70 - 60)/60 = 0.17$). Whereas, for x = 100 to 120 items, the change in number of items (e.g. $\%\Delta x = (100 - 60)/60 = 0.67$) overtakes the performance change. The gradient of performance change is less severe for this section of results in Figure 14(a). This does not mean that the system is adaptable in the range 60-100 items, as it is not adaptable for 60-70 items. Therefore, this system would be classified as not adaptable to any increase in item number from 60. This is because the first increase in items tested is not adaptable (x = 70 items, A < 0). This does not

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.

mean that the system is unusable at 70 items, it may be that 31 items (P i = 70 = 31 items) is a reasonable performance for the user, but it cannot be said to be adaptable, when compared to 60 items, if the user is using this metric to gauge Adaptability. Since no adaptable range was found from 60 items, it was tested again for smaller gaps, starting with 60-61 items. These results are given as performances in Figure 14(c) and the corresponding Adaptability of the performances in Figure 14(d). None of those tested are adaptable (all are A < 0) meaning there is no adaptable range from 60 items.

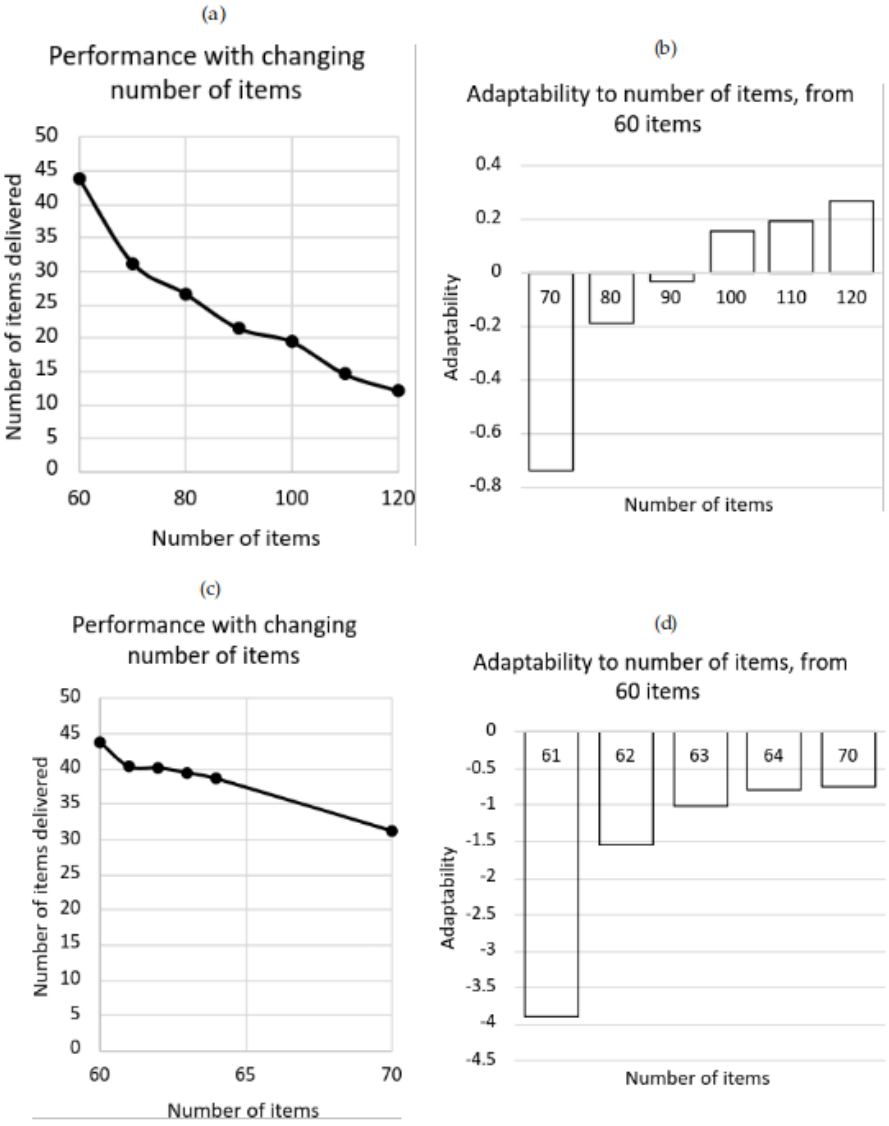

Figure 14: Performance (number of items delivered in time limit) and Adaptability, A (Equation 11), to varying numbers of items. (a) Performance data for 60-120 items (b) Adaptability metric, A, results for 60-120 items, comparing performances (a) to x_o = 60 items (c) Performance data for 60-64, 70 items (d) Adaptability metric, A, results for 60-64, 70 items, comparing performances (c) to performance at x_o = 60 items.

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.

Use case example: Collective decision making

Use case description

In the following, a use case for a collective decision making algorithm with distributed control is considered. The swarm of *N* agents are tasked with making a decision between two sites of different qualities, Site A and Site B, by voting for which Site they think is best at every time step.

Experimental set-up

The **performance** is sum over time of the number of votes for Site A (the higher quality site) every second. This is the Integral $\int_0^T \Sigma V(A)/N\, dt$, at each time step (dt = 0.2s), the sum of the votes for Site A are counted and divided by the total number of agents (ΣV (A) / N). This is summed for the total number of time steps (Total time T = 360s) to give the area under the curve as performance. Each experiment is run for 100 trials and the performance for that given set of parameters is the average of these trials. Figure 15 is a screengrab of the simulated environment with the locations of Site A and Site B shown. Table 3 has the full set of parameters used for each experimental set-up, unless otherwise stated.

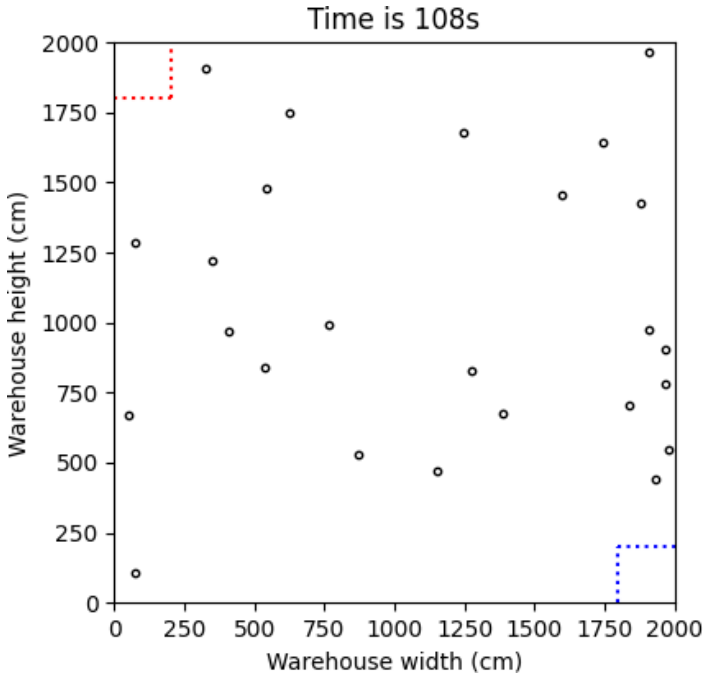

Figure 15: Screen grab of a simulation of the decision making use case. Circles are agents in the swarm. The dotted lined boxes represent the edges of the areas of interest, Site A (bottom, right of image in blue) and Site B (top, left of image in red).

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.

Parameter name	Value	Unit
Arena width	20	m
Arena depth	20	m
Time limit	360	s
Site A size	2.0	m
Site B size	2.0	m
Site A quality	50	
Site B quality	15	
Agent speed	0.25	m/s
Collision avoidance	0.35	m
Communication range	4.0	m
Agent diameter	0.25	m
Item diameter	0.25	m
Time step	0.2	s

Table 3: Experiment parameters for the collective decision making use case. The agents and warehouse dimensions are based on the Toshiba DOTS [15].

Algorithm

While moving around the swarm space the agents perform the same **random walk** and **collision avoidance** behaviours as in the logistics use case. For the decision making algorithm, the agents communicate with local neighbours (within their sensor range, 4 m) every time step. They exchange information about their current vote (for best site) and their level of confidence in that vote, which is represented by an integer. This confidence is equal to the Site quality when they are in that Site, and degrades by 1 every time step that they are out of the site. If their confidence goes to 0 then they change their vote to Undecided

(U). An agent changes their own vote if the majority of their neighbourhood is voting for another site. If the agent does not have the highest confidence of their neighbourhood, then they will change their confidence level (including when not changing their vote) to be equal to the *average confidence of their neighbours* in that time step.

Scalability results

The performance was measured for different swarm sizes and the results for performance and Scalability are given in Figure 16 and expanded in this Section.

Incremental Scalability

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.

Equation 3 was used to measure the Scalability of incremental increases in swarm size. The gap in agent number over which Scalability was measured was always 5 agents, increasing in a total range from 1 to 70 agents. These Scalability results are shown in Figure 16(b). All average performances from 1 to 45 agents were scalable according to the metric given as S > 0. This corresponds to a trend of increasing performances seen in Figure 16(a) for this same region. For agents added beyond this the resulting performance changes are negative, resulting in the increase from 45 to 55 agents being not scalable. There is a slight increase in performance from 55 to 65 agents, resulting in a scalable result S > 0 again, but then it dips to not scalable for the range 65 to 70 agents. The results for increasing swarm size from 5 to 30 agents are superlinearly scalable (S > 1), telling us that the proportional change in performance with change in agent number is superlinear for each increase of 5 agents in this range.

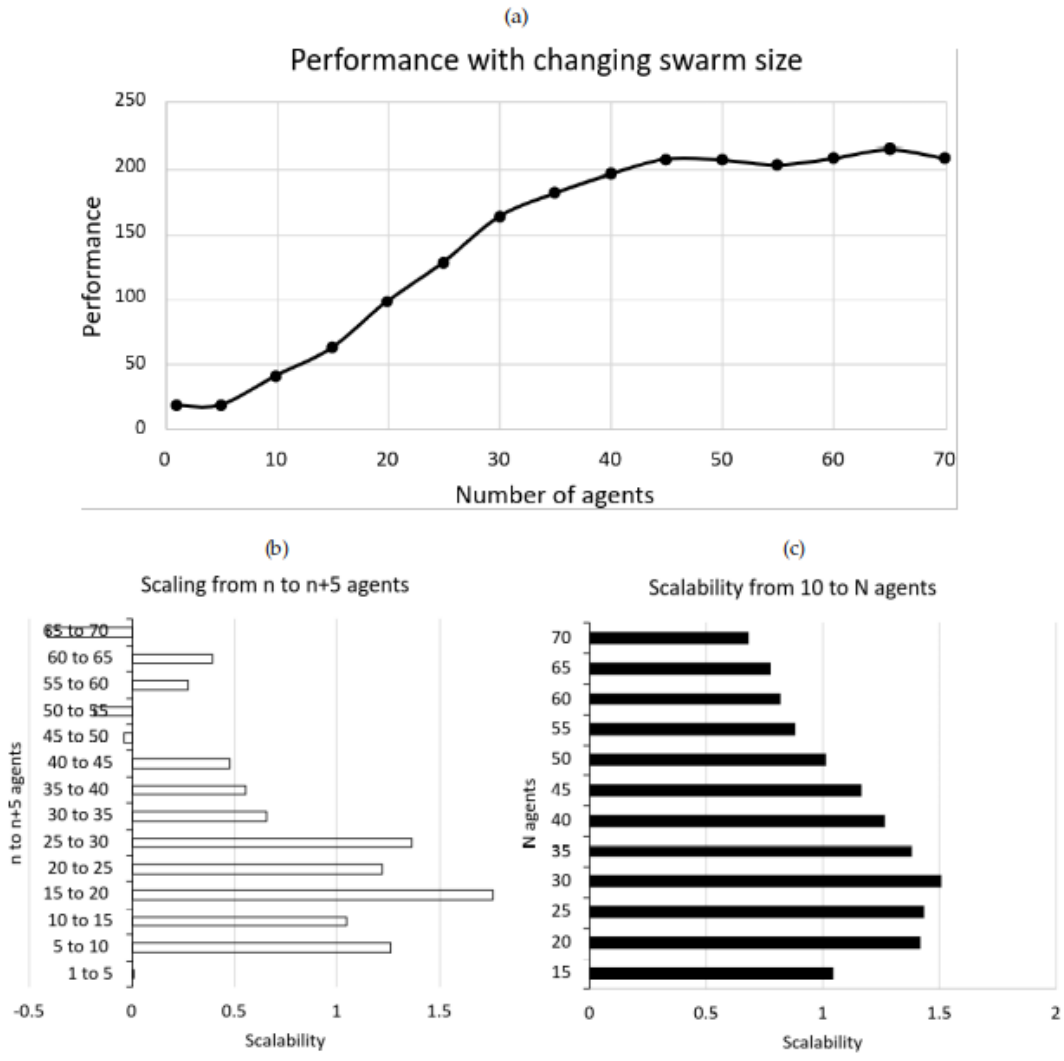

Figure 16: Scalability, S (Equation 3) and performance (integral over time of votes for Site A) for decision making use case. (a) Performance data for various swarm sizes 1-70 agents (b) Scalability measured for incremental changes in swarm size (c) Scalability measured from 10 agents to N agents.

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.

Scalability from 10 agents

In this case, N in Equation 1 and 2 is always N = 10, $P_n = P_{10}$. The results for the Scalability from 10 agents upwards, using performance data from Figure 16(a) are given in Figure 16(c). The results for Scalability from 10 are superlinearly scalable, S > 1, up to 50 robots. The swarm is scalable, S > 0, for all results from 10 to 70 agents.

Specification for Scalability

From the Incremental Scalability results in Figure 16(b), the user could read that the range 5 to 30 agents all have superlinear scalability for each increase of 5 agents in this range. This means that the user would know that they could increase their swarm size in this range for a performance increase that was greater than the current performance per agent. This can help to make decisions about adding agents to the system when cost and efficiency are important to the user. Additionally, increased scalability from 10 agents, Figure 16(c) can tell the user what the most efficient swarm size would be, which here is 30 agents, where S = 1.5 is highest for the swarm sizes tested.

Fault Tolerance and Robustness results

The performance of the system under 2 different failure modes is measured and given in Figure 17(a). In part (a) of this Figure, the scaled down performance P(SD) and the proportional change P(%N) in performance are also given alongside the performances under Failure Modes 1 and 2. A swarm of 25 agents was used for these experiments.

The failed agent's votes are not included in performance. The simulated failure modes were:

- **Failure Mode 1 (FM1)**: Failed agents acted as malicious agents, giving false information to their neighbours. Failure was simulated by a broadcast of a vote for Site B (the lower quality site) with a high confidence of 50. The broadcast was constant and unchanging by either Site A detection or neighbourhood correspondence.
- **Failure Mode 2 (FM2)**: Failed agents cannot detect the Sites A or B. They are still able to communicate with neighbours and give votes and confidence levels based on this communication but they cannot sense the sites if they come across them themselves.

Fault Tolerance results

The results for the Fault Tolerance values for the two failure modes are given in Figure 17(b). The FT results for FM1 are high negative numbers, indicating a catastrophic failure due to this failure mode, even as low as 1 failed agent. Observing the performance data for FM1 in Figure 17(a), this is an accurate result as the performance goes from 128 with no failures to 53 with 1 agent failure. The results for FM2 are very good, with all agent failures

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.

up to 10 (the maximum number tested) being fault tolerant results, FT > 0. This means that the swarm improves upon the performance of the scaled down equivalent swarm, under FM2, in these conditions.

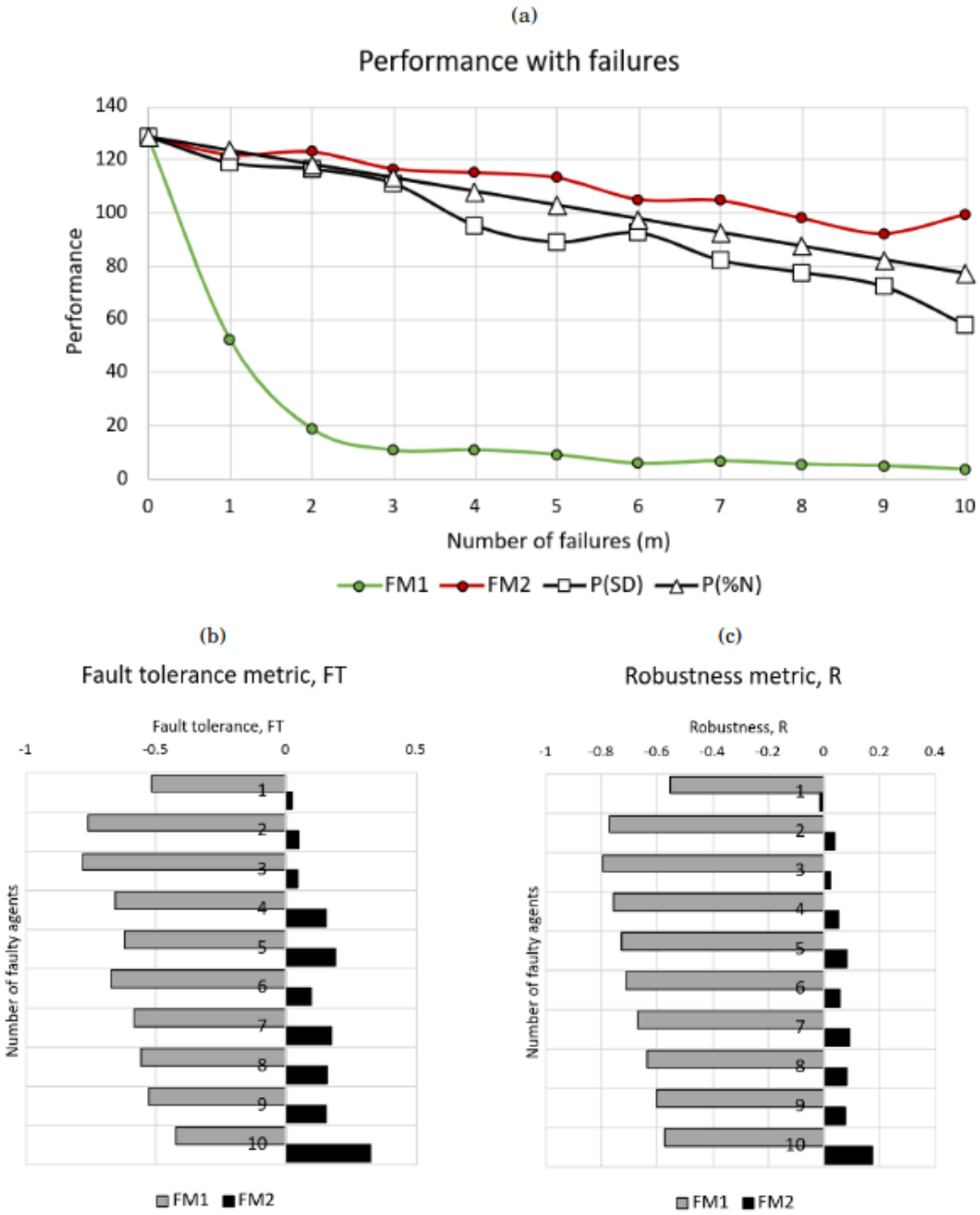

Figure 17: Performance (integral over time of votes for Site A) with different numbers of faulty agents, m, for the decision making use case and Fault Tolerance, FT (Equation 4) and Robustness, R (Equation 8) results. (a) Performance data with m agents failed by each failure mode (FM 1,2) and an equivalent scaled down swarm with 25 − m agents (P(SD)) and P(%N), which is the performance if it reduced by the same proportion as agents lost. If FM-P is above P(SD) at a given m then this is a fault tolerant result, if it is above P(%N) then this is a robust result. (b) FT results (c) R results.

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.

Robustness results

The results for the Robustness values for the two failure modes are given in Figure 17(c). The R values correctly reflect the catastrophic failure that occurred for Failure Mode 1, as all the results for FM1 are R < 0 and therefore not robust. For FM2, all results are R > 0 (robust) except for m=1, which is close to zero but less ($R_{m=1}$ = -0.01).

Adaptability results

Adaptability to changes in Site qualities is measured here by varying the quality of Site A. Site A is always the best site, with a quality that is greater than Site B (which is always 15). How much better Site A is than Site B can make a difference when it comes to making a decision as a group about which is better. Site A quality is varied for these experiments in the range 20-100. A greater disparity in Site qualities is likely to produce a greater performance. However, it is not always possible to choose the quality of two sites in practical use, so it is useful to know how adaptable the system is to different Site qualities.

For each Site A quality tested, 100 trials were tested in an arena with all other parameters unchanged from those given in Table 3. Three different swarm sizes were tested to consider how swarm size affects the Adaptability to changes in Site quality. The swarm sizes tested were 15, 25 and 35 agents. The performance data is given in Figure 18(a). The original Site A quality is taken to be 100 (x_o = 100 in Equation 11). The results for Adaptability are given in Figure 18(b). The system is adaptable, A > 0, for all three swarm sizes when Site A quality drops from 100 to 90 and 100 to 80. The 35 agent swarm has a slight improvement in performance at 90 compared to 100, giving it a very good Adaptability score of A > 1. Swarm sizes 25 and 35 are both adaptable all the way to a Site A quality of 20, from 100. However, the 15 agent swarm is not adaptable to a Site quality change of 100 to 70, or any below 70.

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.

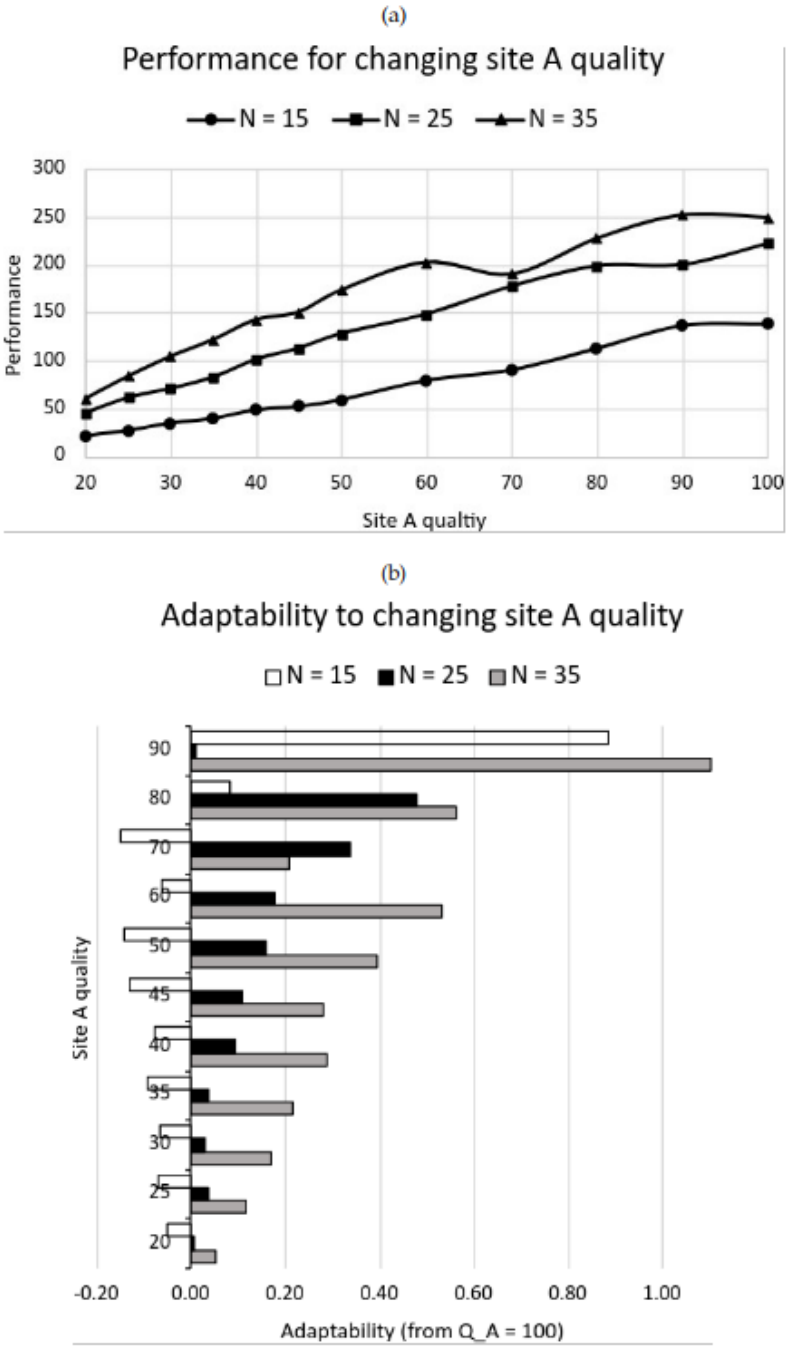

Figure 18: Performance (integral over time of votes for Site A) and Adaptability, A (Equation 11), to varying qualities of Site A. (a) Performance data for qualities 20-100 (b) Adaptability metric, A, results for qualities 20-90, comparing performances from (a) to $x_o = 100$

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.

Discussion

Successful metrics

The Swarm Performance Indicators (SPIs) tested here are metrics for: Scalability; Adaptability; Robustness; and Fault Tolerance. The SPIs were tested on two distributed control use cases to determine if they had these traits successfully and what the limits were of these traits in these use cases. For example, different swarm sizes were compared for their Adaptability to changes in target site quality (Decision Making use case, results in Figure 18(b)). This is useful for determining what swarm size is best when you know the Site quality will vary. When transferring swarms to the real-world, it will be necessary to write specifications for their design and usage and to communicate to the user their limitations and strengths. These SPIs provide quantitative ranges of parameters that can contribute to a specification for this purpose. For example, the Scalability metric has been successfully used to determine a range of swarm sizes to which the logistics use case system is super scalable. Bjerknes and Winfield (2013), among other works, said that it is important for metrics describing the effect of faults to answer questions about what the swarm is tolerant to and to what extent it is robust. These are given by these metrics for Fault Tolerance and Robustness in the results here, with specific failure modes and number of faulty agents given alongside the metrics. Finally, the metrics are generalisable across a range of swarm systems and tasks. They are generalisable because the SPI definitions are based on changes in performance in proportion to parameters that are given in each use case when the metric is applied. The metric can therefore be applied to any system or task because they are based only on performance and parameter data from that system or task and compared within the context of that use case.

Context and Boolean classifications

Boolean classifications of e.g. "robust" or "not robust", are reductionist without further context, as is discussed in the justification for these metrics. Therefore, when these metrics are given as proof of these swarm traits, it is important to include the circumstances in which the system achieved the given score. For example, if stating that the system is scalable according to Equation 3, it is paramount that the user also say what swarm sizes it is scalable to and from, and what the experimental set-up was. This is given in the Use Case 1 and 2 results in this work, such as for the decision making (DM) case study, it could be said that "A swarm following the DM algorithm is scalable in the range 10 to 70 agents with the experimental set-up given in Table 3. The same swarm set-up is superlinearly scalable in the range 10 to 50 agents". Similarly, Fault Tolerance and Robustness are given alongside the failure mode, they are tolerant/robust to, and the number of faulty agents in the system at the time of measurement. Additionally, the Adaptability is measured to a change in a specific and stated parameter with detailed experimental set-ups also given. For each of the metrics the full context is clearly stated and successfully used to describe the effects of faults, scaling swarm size and changing parameters.

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.

Clear definition of terminology of metrics

In the literature there is dispute over what these different effects should be termed and what would describe them. A full description of the current literature is given in Section 5.2 and the metric terminology is justified. However, there will still be some confusion going forward because of varied terminology and inevitable further discussion of an evolving lexicon. Therefore, the metrics are clearly described in this work by the effect that they are measuring, not just as a single word, such as "adaptability" without explanation. These defined effects are shown to be successfully measured in each case but when these metrics are used in other works it should be made clear by the user that they are descriptors of data. For example, 'Fault Tolerance, FT' is, here, short-hand for the proportional difference between the scaled down swarm (N − m agents) performance and the faulty swarm performance (N agents with m faulty agents). When discussing FT values, it should always be made clear that any discussion of "fault tolerance" derives from this measurement. This is particularly important when performance can become confused with these traits. These metrics are not measures of good or bad performance: that can be read straight from performance data and user-set thresholds of acceptable values. Instead, they are measures of the effects on performance due to changes in their circumstances. For example, a fault tolerant result, FT > 0, may have a performance that is unacceptable to the user but that is separate to the information given by the FT metric. Just because something passes as good in terms of the Swarm Performance Indicators, does not mean that it has good performance for the user. This judgement is user dependent, as different performances are acceptable to different users depending on their needs and cannot be generalised here.

Using proportional change

Proportional changes were used in these metrics, e.g. $\% \Delta P$ as opposed to the difference between data points of $\Delta P = P 2 - P 1$. This is beneficial as it removes units from the equation. Without units, the change in any performance parameter can be directly compared to any other factor e.g. number of faulty agents. For example, comparing a loss of 50 seconds to a loss of 3 agents is difficult to digest whereas a loss of 5% of performance compared to 2% loss of agents is directly comparable. Using proportional changes also brings in information about how significant that change is to that particular system, because it is a percentage change from the original performance. For example, a change in performance of 10% compared to 20% is easier to compare directly than a change in 10 seconds compared to 20 seconds. It may be that the original performance was 1000 seconds, in which case 10 seconds would not be significant, or it could have originally been 30 seconds in which case a change of 10 seconds is huge. This makes the metric more generalisable as the change in performance discussed is directly dependent on the original, baseline performance. The issue with using proportional changes is that as the deviation from the original parameter grows, the amount of loss of performance that is acceptable increases with it. For example, according to Equation 11, a system which has a parameter change of $\% \Delta x = 0.01$ is only adaptable if the performance loss is less than 0.01. This threshold could be missed by variation in results alone because the criteria is so tight. This can result in confusing metric results where e.g. a system that was not considered robust to 1 to 5 failures, is considered robust to 5 to 10 failures, according to the metric for R. This happens in the decision making use case, which is found to be robust to Failure Mode 2 for 2-10 agent failures but not robust to 1 failure (see results Figure 17). Only a small amount of

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 loss is deemed acceptable at this low failure rate and the variation in mean performance can be outside of this range, causing the data to score poorly, despite scoring well with higher losses of agents.

Variability in results

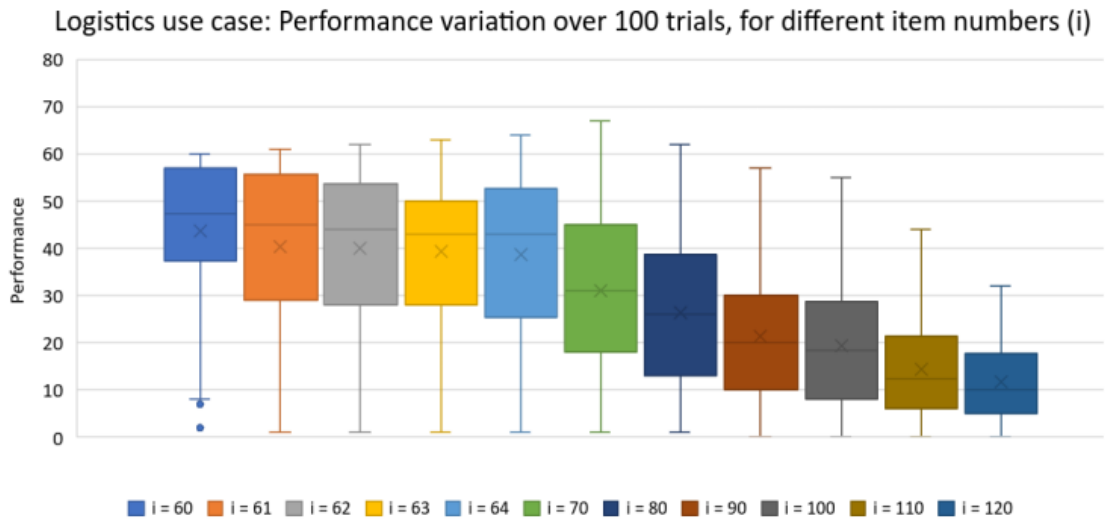

Figure 19: The full set of the performances (number of items delivered in the time limit) for 100 trials in the logistics use case, for varying item numbers (i = 60 to 120 items). This shows the various nature of the results, which make the metrics difficult to measure using average performances.

Using average performance can cause variability in the results to be an issue when measuring these traits. For example, in the logistics use case, when measuring the Adaptability to changes in the number of items, the system appears to be adaptable from 60 to 100 items but not for 60 to 70 items (see Figure 14(b)). When smaller gaps between 60 and 70 items are tested to find what the system is adaptable to within this range, it was found to not be adaptable even to one additional item (see Figure 14(d)). Variability in the results is shown to be a factor in this confusing result, as follows. The set up with 60 items was tested for another 100 trials. This second set of trials gave an average performance of 39.65 items collected, compared to the first set in which 43.81 items were collected in identical conditions. The variation between these two results for 60 items is greater than the variation measured between 60 and 61 (P 61 = 40.24 items). The full set of the performances for 100 trials in the logistics use case, for varying item numbers are given in Figure 19. This shows the various nature of the results, which make Adaptability (and other metrics) difficult to measure using average performances.

The variability in performance data is not included in these metrics, which could mean that a user could be using a system, expecting it to be e.g. adaptable to x, but finding that it is only adaptable in Y% of uses. For example, for the logistics use case, the results for m = 1, Failure Mode 2 (no Site detection) was found to be a fault tolerant result. However, when examining all 100 trials individually, 55% were fault tolerant but 45% were not fault tolerant . This is a high percentage, which means that it is likely a user of this system will often find

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.

that the performance is less than the scaled down swarm with 24 agents, despite having a "good Fault Tolerance" rating when using average performances. When using these metrics it is good to include some discussion of the variability in the average performance used as either an issue or positive justification for the presence of the trait in a given system.

Uses

The SPIs can be useful for fitness functions when developing swarm algorithms in evolutionary models. These metrics provide a way of selecting for adaptability, robustness, fault tolerance and scalability. The SPIs have a Boolean classification of above or below zero but they also scale with the metric value, meaning that e.g. R = 0.7 is better than R = 0.1, by which it provides useful, descriptive information about the system's (e.g.) Robustness beyond a yes/no definition and allows for trade-offs that can be quantitatively described. For example, going from one algorithm to another may improve Scalability but decrease Robustness. The degree of this improvement vs decrease can be measured with these metrics. The metrics are useful when they are helping to quantify a report on the behaviour of a swarm, e.g. to measure improvement to adaptability when developing an algorithm, or to compare two systems for their level of Scalability.

Limitations

Limitations of the Adaptability metric

The changing parameter x is restricted to a numerical value. In the literature, some adaptability definitions apply to a swarm being used for different condition sets. For example, the Adaptability of a swarm to being applied to entirely different use cases or tasks. These full, various condition sets cannot be described by one parameter so cannot be used with this Adaptability metric. Instead, this metric is useful for the case where a user would want to know how adaptable a particular system and set-up is to a change in one particular parameter. This benefits from being generalisable and specific but can lose some definitions of adaptability for the application of swarms to more broad changes in circumstances.

Limitations of the Scalability metric

The *S* metric cannot measure how easy or complicated it is for agents to be added or removed from a swarm in practical use terms. Part of the discussion of scalability in swarms is how easily new agents can be added without requiring complicated processes or pausing the rest of the swarm behaviours. For example, Şahin (2004) describes how a swarm is scalable by saying agents could be "poured" into the group to add them to the system. Therefore, there are some scalability factors that are not encompassed by this metric. These features may be limited to qualitative discussion by describing the process to add agents to the group.

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.

Limitations of the Fault Tolerance and Robustness metrics

One feature of swarms that could be why they are so Robust/Fault tolerant is their self-organisation ability to self-repair. Bjerknes and Winfield (2013) discuss self-repair as a swarm's ability to avoid negative effects from failed agents. They simulate failed robots in the system during a taxis task and monitor how the swarm becomes 'anchored' to the failed agents but then recovers performance and manages to complete the task. This is considered good Robustness/Fault Tolerance. The pattern of performance over the course of the task is used to measure self-repair time. In the FT and R metrics, the pattern of performance over time is not considered and self-repair is not discussed. Self-repair cannot be applied to all swarm algorithms or tasks, so it is left out to make the metric more generalisable but it should be considered when analysing cases where it occurs. Some guidance on how to include or measure self-repair could be included in future work with these metrics.

Conclusions

Swarm Performance Indicators are quantitative metrics for Fault Tolerance, Robustness, Scalability and Adaptability in multi-agent systems. These traits have been given in the literature as being inherent to swarm systems, without measurement, which has created a gap in the research for a method of measurement. The exact definitions for these traits and their terminology is disputed in the literature but their definitions here are justified with a full literature review on the topic in swarm systems. The metrics are used to successfully describe the effects they aim to measure on two use cases, which both use distributed control. It is shown that the metrics can be used towards specifications for swarms and for improving their explainability. The metrics can also be used to compare two systems in terms of these swarm traits or to quantify the improvement (or otherwise) in them when different system parameters, such as swarm size, are changed.

References

[1] A. F. Winfield and J. Nembrini, "Safety in numbers: Fault tolerance in robot swarms," International Journal of Modelling Identification and Control, vol. 1, pp. 30–37, 01 2006.

[2] M. K. Heinrich, M. Wahby, M. Dorigo, and H. Hamann, "Swarm robotics," 2022.

[3] M. Schranz, M. Umlauft, M. Sende, and W. Elmenreich, "Swarm robotic behaviors and current applications," Frontiers in Robotics and AI, vol. 7, p. 36, 2020.

[4] M. Dorigo, G. Theraulaz, and V. Trianni, "Swarm robotics: Past, present, and future [point of view]," Proceedings of the IEEE, vol. 109, no. 7, pp. 1152–1165, 2021.

[5] J. Bjerknes and A. F. Winfield, On Fault Tolerance and Scalability of Swarm Robotic Systems, 01 2013, vol. 83, pp. 431–444.

[6] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," Swarm Intelligence, vol. 7, pp. 1–41, 2013.

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.

[7] H. Hamann and A. Reina, “Scalability in computing and robotics,” IEEE Transactions on Computers, vol. 71, no. 6, pp. 1453–1465, 2022.

[8] L. Zhu, C. Johnsson, M. Varisco, and M. M. Schiraldi, “Key performance indicators for manufacturing operations management – gap analysis between process industrial needs and iso 22400 standard,” Procedia Manufacturing, vol. 25, pp. 82–88, 2018, proceedings of the 8th Swedish Production Symposium (SPS 2018). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2351978918305791

[9] J. Harwell and M. Gini, “Swarm engineering through quantitative measurement of swarm robotic principles in a 10,000 robot swarm,” arXiv preprint arXiv:1907.03880, 2019.

[10] E. S ̧ahin, “Swarm robotics: From sources of inspiration to domains of application,” in Swarm Robotics: SAB 2004 International Workshop, Santa Monica, CA, USA, July 17, 2004, Revised Selected Papers 1. Springer, 2005, pp. 10–20.

[11] M. Schranz, G. A. Di Caro, T. Schmickl, W. Elmenreich, F. Arvin, A. S ̧ekercio ̆glu, and M. Sende, “Swarm intelligence and cyber-physical systems: Concepts, challenges and future trends,” Swarm and Evolutionary Computation, vol. 60, p. 100762, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210650220304156 39

[12] J. P. Hecker and M. E. Moses, “Beyond pheromones: evolving error-tolerant, flexible, and scalable ant-inspired robot swarms,” Swarm Intelligence, vol. 9, pp. 43–70, 2015.

[13] C. Prehofer and C. Bettstetter, “Self-organization in communication networks: principles and design paradigms,” IEEE Communications Magazine, vol. 43, no. 7, pp. 78–85, 2005.

[14] E. S ̧ahin, “Swarm robotics: From sources of inspiration to domains of application,” in International workshop on swarm robotics. Springer, 2004, pp. 10–20.

[15] S. Jones, E. Milner, M. Sooriyabandara, and S. Hauert, “Dots: An open testbed for industrial swarm robotic solutions,” ArXiv, vol. abs/2203.13809, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:247748922

[16] Milner, E., Sooriyabandara, M., & Hauert, S. (2022). *Swarm Diffusion-Taxis: Transport of spatial information for cooperative gradient-based navigation*. Paper presented at AROB-ISBC-SWARM 2022.

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.