



# Engineering Distributed Collective Intelligence in Cyber-Physical Swarms

Gianluca Aguzzi, Claudio Savaglio

## ► To cite this version:

Gianluca Aguzzi, Claudio Savaglio. Engineering Distributed Collective Intelligence in Cyber-Physical Swarms. DISCOLI workshop on DISTRibuted COLlective Intelligence collocated at the 20th International Conference on Distributed Computing in Smart Systems and the Internet of Things, Apr 2024, Abu Dabi, United Arab Emirates. <hal-04666835>

**HAL Id: hal-04666835**

**<https://hal.science/hal-04666835v1>**

Submitted on 2 Aug 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

# Engineering Distributed Collective Intelligence in Cyber-Physical Swarms

1<sup>st</sup> Gianluca Aguzzi

University of Bologna

Cesena, Italy

gianluca.aguzzi@unibo.it

2<sup>nd</sup> Claudio Savaglio

University of Calabria

Calabria, Italy

csavaglio@dimes.unical.it

**Abstract**—Cyber-physical swarms represent a paradigm shift in distributed systems, mirroring characteristics akin to natural swarms, such as self-organization, scalability, and fault tolerance. This paper delves into these complex systems, characterized by vast networks of cyber-physical entities with limited environmental awareness, yet capable of exhibiting emergent collective behaviors. These systems encompass a diverse array of scenarios, ranging from swarm robotics to the interconnectivity in smart cities, as well as the collaboration among augmented humans. The engineering of such systems presents unique challenges, primarily due to their intricate complexity and the *spontaneous* nature of their collective behaviors.

This paper aims to dissect these challenges, offering a clear delineation of potential approaches. We present a comprehensive analysis, shedding light on the intricacies of engineering cyber-physical swarms and discussing modern solutions in engineering collective applications for such systems.

**Index Terms**—distributed collective intelligence, large-scale IoT, collective computing, aggregate computing

## I. INTRODUCTION

The emergence of cyber-physical swarms (CPSWs) marks a significant evolution in the realm of distributed systems. The central premise of CPSWs lies in the integration of digital and physical components to create a network of entities capable of *collaborative* and *intelligent* behavior beyond the capabilities of individual components. These systems, characterized by their ability to mimic the self-organizing, scalable, and resilient properties of natural swarms, are rapidly transforming how we approach problems in a myriad of fields ranging from robotics to urban planning. In this paper, we delve into the engineering complexities and the innovative potential of CPSWs. Our focus is on the unique challenge of orchestrating numerous cyber-physical entities, each with limited individual capabilities and awareness, to exhibit sophisticated, emergent behaviors. This challenge is not just technical but also conceptual, as it requires a paradigm shift in how we perceive and design distributed systems. In particular, we aim to address the following research questions:

**RQ1** What are the main characteristics of CPSWs and how these difference to other distributed systems?

**RQ2** What are the main approaches to engineering CPSWs?

**RQ3** What are the main challenges in engineering CPSWs?

In essence, this paper is a comprehensive exploration of the engineering principles, challenges, and applications of

cyber-physical swarms. It serves as a guide for researchers and practitioners alike, offering a window into the future of distributed systems where *collective intelligence*, *scalability*, and *resilience* are not just desirable but essential features. The paper is organized as follows: Section II provides a motivating example, Section III discusses the main characteristics of CPSWs, Section V explores the engineering of CPSWs, Section IV discusses related concepts, and Section VI concludes the paper.

## II. MOTIVATING EXAMPLE: WILD-FIRE MONITORING IN EXTENSIVE FORESTS

Canada is renowned for its lush forests, which covers approximately one-third of the nation's landmass, making it one of the countries with the largest forested areas globally. However, the increasing impact of climate change has led to a surge in wildfires, causing widespread devastation to both local flora and fauna. Consider for instance the 2023 wildfires [1], which burned over 43 million acres, that is about 5% of the entire forest area of Canada.

To address this issue, there is a growing consensus on the need for *proactive* and *localized* interventions. Specialized monitoring systems are essential for keeping tabs on vulnerable regions, especially given the impracticality of relying solely on human observation due to the vastness of these areas. One innovative solution might be the deployment of a sophisticated network of environmental sensors, strategically placed to monitor temperature, humidity, and other fire-prone conditions. These sensors might be complemented by a swarm of drones equipped with advanced imaging and data collection capabilities.

The system operates through ongoing collaboration between ground-based sensors and aerial drones. The drones provide a bird's-eye view of the landscape, allowing for real-time monitoring of a wide array of nodes across large geographical expanses. This integrated approach enables the early detection of potential fire hazards, thereby facilitating the pre-emptive mobilization of emergency services.

However, the implementation of such a comprehensive monitoring system comes with its own set of challenges. Each sensor or drone has a limited operational range and can only provide a *partial* view of the overall system. Therefore, robust data fusion algorithms are necessary to merge information

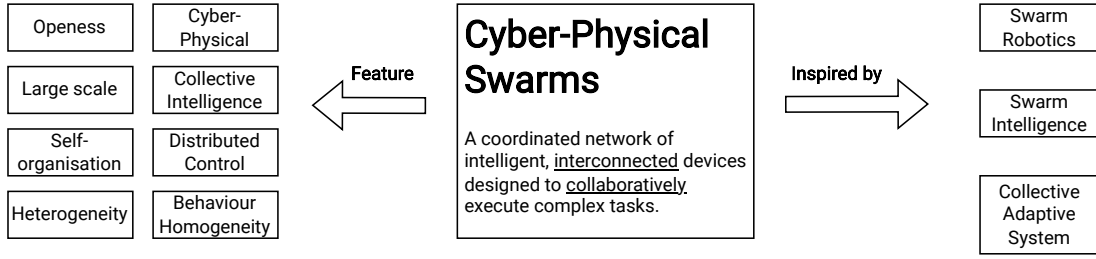


Fig. 1. Overview of Cyber-Physical Swarms

from multiple sources into a cohesive and actionable overview. Environmental conditions are also highly dynamic, requiring the system to *adapt* in real-time to changing variables such as wind speed, temperature fluctuations, and precipitation levels.

Given that drones have limited battery life and need to be recharged, the system must be capable of *self-organizing* to ensure uninterrupted monitoring. This is particularly challenging considering that the number of deployed devices could easily exceed *thousands* of units across the extensive area of interest.

Moreover, the system must be highly responsive to emergency conditions. This necessitates that each device, whether a sensor or a drone, should be equipped with *edge intelligence capabilities* [2] for performing local analyses. These local analyses can then be used to trigger collective alarms, ensuring immediate action is taken to mitigate the risk of wildfires.

### III. CHARACTERISTICS – RQ1

Drawing from the motivating example, several distinct characteristics emerge that set CPSWs apart from other large-scale distributed systems, like Collective Adaptive System and Swarm robotics. These unique traits (highlighted in Figure 1) not only define the essence of CPSWs but also pose specific challenges in their design and implementation. Each of these characteristics will be discussed in detail below.

#### A. Scale

CPSWs are *inherently* scalable, often comprising hundreds or even thousands of interconnected nodes. The behavior scale independence ensures that the same programmatic logic can be applied across networks of varying sizes, making them highly adaptable to different application scenarios. In doing this, is essential to capture the right collective abstraction that allows both to help the developers to think in terms of the collective behavior and to effectively design the collective behavior itself.

#### B. Device heterogeneity

While the nodes in a CPSW may possess varying hardware capabilities and attributes, they are engineered to operate cohesively. Interoperability stands as a fundamental requirement for these systems, particularly because they often incorporate a diverse array of devices manufactured by different vendors. In today’s IoT landscape, semantic interoperability presents a significant challenge. Devices from various vendors might employ divergent communication protocols or data formats,

complicating the task of achieving seamless interaction. Such diversity introduces an additional layer of complexity in system design, necessitating the identification of a common framework that enables effective *communication* and *collaboration* among the devices.

#### C. Behaviour homogeneity

Despite the diverse hardware configurations, nodes within a CPSW display *consistent* behavior at the local level. They run the same local *algorithms*, thereby achieving the system’s collective objectives. However, it is important to clarify that “homogeneous behavior” does not imply that all nodes *do the same thing at the same time*. Indeed, given the system’s inherently distributed and extensive: the input received by one node differs from that of another far from it, leading to variations in local behavior.

#### D. Distributed control

Centralized control mechanisms are often impractical in CPSWs due to their scale and complexity. In a centralized system, a single controller manages the behavior of all agents, necessitating high computational power and introducing a single point of failure. This becomes increasingly challenging when dealing with a CPSW that comprises a large scale of highly distributed agents. Any failure or delay in the central controller could lead to system-wide disruptions.

In contrast, distributed control algorithms decentralize the decision-making process, allowing nodes to make local decisions based on their immediate environment and information from neighboring nodes.

#### E. Self-organization

One feature of CPSWs is their ability to *self-organize*. Nodes can dynamically adapt to changing conditions, reconfiguring themselves to maintain system *integrity* and *performance*. This is particularly important in scenarios where the environment is unpredictable or hostile. To be more precise, self-organization is defined as a dynamical and adaptive process where systems acquire and maintain structure themselves, without external control. It can be also denoted as a property “to arrange several elements into a purposeful sequential or spatial (or both) order or structure”. Therefore, self-organization is an autonomous, robust, and flexible process that seeks an increase of order [3].

### F. Openness

CPSWs are typically *open* systems, meaning that the total node count is not predetermined. New nodes can join or leave the network dynamically, requiring the system to be *flexible* enough to accommodate these changes without compromising its functionality.

### G. Collective intelligence

The nodes in a CPSW work collaboratively to achieve common goals, displaying a form of *collective intelligence* [4]. This is facilitated by advanced algorithms that enable the system to learn from its environment and adapt its behavior accordingly. Through decentralized decision-making processes, each node contributes its unique computational resources and sensory data, leading to a more robust and resilient system. This collective approach allows for enhanced efficiency, as the system can dynamically allocate tasks and resources based on real-time needs and constraints. The *collective intelligence* exhibited by the nodes also enables fault tolerance and self-healing capabilities. If one node encounters a failure or is compromised, the remaining nodes can recalibrate and reorganize to continue functioning effectively.

Therefore, collective intelligence in CPSW is not merely a by-product but a crucial feature that substantially enhances the system's *adaptability*, *resilience*, and overall performance.

### H. Cyber-physical interactions

The term “cyber-physical” aptly describes the integration of computational and physical elements in CPSW. Each node is usually connected to a physical entity via sensors and can influence its environment through actuators. This seamless integration is crucial for applications that require real-time interaction with the physical world, such as industrial automation, healthcare monitoring, and autonomous vehicles.

In summary, the term “cyber-physical” encapsulates a sophisticated blend of computational intelligence and physical interaction. The seamless integration of these elements requires a modern computing paradigm, making CPSW highly versatile and robust.

## IV. RELATED PARADIGMS

This section will discuss some of the related concepts that

	MAS	Swarm Robotics	CAS	CPSW
Scale	Dozens	Hundreds	Thousands	Thousands
Device heterogeneity	Yes	No	Yes	Yes
Behaviour homogeneity	No	Yes	No	Yes
Distributed Control	Yes/No	Yes/No	Yes	Yes
Self-organization	No	Yes/No	Yes	Yes
Openness	Yes/No	No	Yes	Yes
Cyber-Physical	Yes/No	Yes	Yes/No	Yes

TABLE I

SUMMARIZED COMPARISON BETWEEN MULTI-AGENT SYSTEMS (MAS), SWARM ROBOTICS, COLLECTIVE ADAPTIVE SYSTEMS (CAS), AND CYBER-PHYSICAL SWARMS (CPSW)

are closely aligned with CPSW summarized in Figure 2 and Table I.

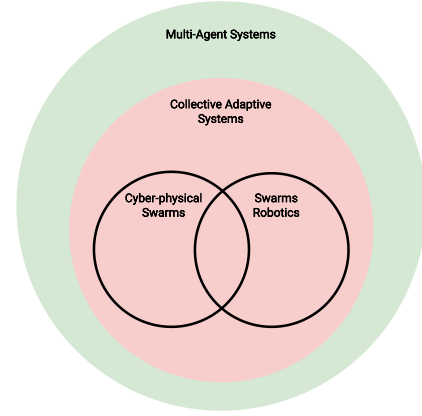


Fig. 2. Taxonomy of CPSW with respect to related systems

### A. Multi-agent systems

In the realm of computational systems, a CPSW can be closely related to a Multi-Agent System (MAS), with a specific emphasis on being a *many*-agent system. In a MAS, multiple autonomous agents interact with each other to achieve specific objectives. These agents are capable of sensing their environment, making decisions based on their observations, and then acting upon those decisions. The key difference in a CPSW is the scale and the integration of physical components, such as sensors and actuators, which allow for real-time interaction with the environment.

In a CPSW, agents are not just *virtual entities* but are tethered to physical components, making them cyber-physical agents. These agents continuously engage in a cycle of *repeated* sensing, computation, communication, and actuation to achieve collective behaviors.

Moreover, the agents in a CPSW are designed to work collaboratively to achieve collective goals, rather than pursuing individual objectives. This is in contrast to some MAS where agents might have conflicting goals. In a CPSW, the focus is on achieving a form of collective intelligence through distributed computing and decision-making. Agents share information and coordinate their actions to solve complex problems that are beyond the capabilities of any single agent.

In summary, while CPSW shares similarities with MAS in terms of multi-agent interaction and decision-making, they extend the concept by incorporating large-scale, cyber-physical integration, and a focus on collective intelligence.

### B. Swarm intelligence

It is a branch of collective intelligence with an interdisciplinary field that draws inspiration from the collective behaviors observed in social animals, such as ants, birds, and fish, to develop computational algorithms and systems. The methodology employed is fundamentally bottom-up. Designers and researchers study the behaviors of individual animals in their natural habitats to understand the rules or heuristics they follow. These individual behaviors are then modeled computationally to observe how they contribute to

the emergence of collective intelligence in a swarm. One seminal concept that emerged from this line of inquiry is *stigmergy* [5]. Stigmergy is a form of indirect communication and coordination where agents in a swarm interact with each other by modifying a shared environment, rather than through direct communication.

In recent years, the focus of swarm intelligence has evolved to concentrate more on algorithmic aspects rather than swarm robotic (see next section). Researchers have started to leverage the principles of collective behavior to develop optimization algorithms that can solve complex computational problems. Some of the most well-known algorithms that have emerged from swarm intelligence research include ant colony optimization (ACO) [6], particle swarm optimization (PSO) [7], and flock of starling optimization (FSO) [8]. Each of these algorithms has its own set of rules and heuristics, modeled after the specific animal behaviors they are inspired by, and they have been applied successfully in various domains such as network design, resource allocation, and data clustering.

While these optimization algorithms offer valuable methodologies and have broad applicability, these differ from the notion of CPSWs. Indeed, the focus of such systems is on harnessing the *principles* of swarm intelligence to develop *artificial* systems that can achieve similar collective behaviors through mechanisms of *self-organization*. Unlike traditional swarm intelligence algorithms that often aim to *mimic* nature, CPSW engineering draws *inspiration* from natural systems to understand the fundamental principles that make these systems *robust*, *scalable*, and *adaptable*. The ultimate goal is to exploit these principles to design and implement artificial swarm systems that can operate effectively and efficiently in complex, dynamic, and potentially hostile environments.

### C. Swarm robotics

Historically, the field of swarm robotics has its roots in the early approaches to swarm intelligence. Over time, it has evolved to become the *engineering* arm of swarm intelligence, focusing on the *practical* aspects of building and maintaining swarm systems. The overarching goal of *swarm engineering* is to establish a rigorous methodology for the entire lifecycle of a swarm robotics system, from conceptualization to operation and maintenance [9].

In traditional swarm robotics, the primary focus is on *robots* that are *autonomous*, *situated*, and operate under *no central control*. These robots are designed to interact with each other and their environment to achieve collective goals. CPSWs extend this paradigm to include other “swarm-like” systems that may not necessarily involve robots. Examples include crowds of people in public spaces, large-scale IoT networks, and smart city infrastructures. These systems share many similarities with swarm robotics, such as the need for autonomy, localized decision-making, and collective behavior, but they also present unique challenges and opportunities.

One of the most promising avenues for extending the principles of swarm robotics to these other domains is the emerging field of *automatic design* [10]. In automatic design

approaches, the control logic for the agents in the swarm is not manually programmed but is instead derived through optimization techniques such as *genetic algorithms* or *multi-agent reinforcement learning*. These methods aim to optimize a *global* utility function that captures the overall objectives of the swarm. This is particularly appealing for CPSW, where the complexity and scale of the system make manual programming impractical.

### D. Collective adaptive systems

Collective adaptive systems (CAS) are a broad class of systems composed of agents (potentially heterogeneous) capable of adapting to environmental conditions while striving to achieve a collective goal through the emergence of individual node cooperation. Typically, individual units are simple and draw strong inspiration from natural systems. In these systems, which are adaptive by nature, there is interest in incorporating *self-\** properties (in fact, these CAS are sometimes discussed as collective self-adaptive systems). These properties include *self-healing*, *self-optimization*, and *self-configuration*. Self-healing refers to the system’s ability to recover from failures without human intervention. Self-optimization means the system can improve its performance over time based on feedback. Self-configuration allows the system to adapt to changing conditions without requiring manual adjustments. These *self-\** properties contribute to the system’s overall *resilience* and *efficiency*.

CPSWs can be understood as a specialized subset of the broader CAS. This specialization arises from two key distinguishing characteristics: i) the involved entities are not merely digital but also have a *cyber-physical* nature, integrating both computational and physical elements, and ii) despite the heterogeneous nature of the devices within the system, their behavior manifests uniformly. These unique attributes have a profound impact on the system’s design methodology, which is why this thesis specifically focuses on this subclass of systems.

## V. ENGINEERING COLLECTIVE INTELLIGENCE – RQ2

To address the characteristics discussed, researchers over the years have proposed various approaches that can be divided into two major areas: *automatic design* and *manual design*, with a novel approach that combines the two so-called *hybrid design*. In the following, we will give an overview of the main approaches in each area, and then we will discuss the challenges that are still open in the field.

a) *Automatic design*: regarding automatic design, traditional machine learning and reinforcement learning [11] approaches (both single-agent and multi-agent) have been found to be ineffective as they do not take into account scale and face issues of *non-stationarity* when many agents learn concurrently. A modern approach to such systems falls under the branch known as “many-agent reinforcement learning” [12], where the aspect of *scale* is directly tackled through simplifications in the models trained. Initial work in this area was based on the concept of *swarming agents* [13], where the goal was to find a “homogeneous” controller for the

entire system. More modern approaches utilize ideas from game theory and mean-field approximations. In these, a *game* is structured where one agent plays against “all” others. In spatial approaches, this global agent is approximated by the average response of its neighbors. In this direction, mean-field reinforcement learning [14] has been introduced. Like many other large-scale learning systems, it leverages a centralized solution for training while creating distributed policies [15]. It is important to note that these approaches are still in their infancy and have not yet been applied to CPSWs. The primary limitation of these methods lies in their inability to scale with the complexity of collective applications, coupled with challenges in transferring the experience gained across various scenarios. Indeed, an ideal approach would be one that simplifies the composition of various collective behaviors to create more complex systems. This aligns with the principles of manual design.

b) *Manual desing*: in these approaches, two different areas can be identified. There are *micro approaches* and *macro approaches*. In the foster, the goal is to create collective behavior through emergence by programming individual devices. This approach, typically inspired by natural behaviors and then emulating them, has proven inadequate for scaling in complexity, resulting in ad-hoc solutions for each behavior [16]. Moreover, the abstraction gap between the solution (i.e., the collective behavior) and the implementation (i.e., the individual controller being developed) was extremely high, leading to significant cognitive effort. Therefore, a modern approach involves the use of *macroprogramming*, which shifts the focus to programming the “collective” directly, captured by a certain *programming abstraction*. Various approaches have followed this direction, originating in wireless sensor networks. Some are more practical (like Buzz [17], which allows the use of *stigmergy* [18] and the concept of a *swarm*), others more theoretical (like TOTA [19], where tuples were used to create intelligent agents). One of the most modern approaches is *aggregate computing* [20]. The idea of this paradigm is to manage complexity directly at the language level with a top-down approach. Indeed, the programmer must describe collective behavior through the combination of (*computational*) *fields*. A computational field can be seen as a spatiotemporal structure that maps a certain value for each device. This paradigm allows to create of *scale-independent* collective behaviors, as the same programmatic logic can be applied across networks of varying sizes. Moreover, the abstraction gap is reduced, as the programmer can focus on the collective behavior (i.e., the combination of fields), rather than the individual controller. Furthermore, the program are self-organizing by design, therefore the resulting system is more robust and resilient.

Aggregate computing has already been used in many CPSWs scenarios, such as swarm robotics [21], crowd engineering [20], and WSNs [21]. However, in such scenarios, there are still some open issues, for instance how to handle highly dynamic environments, where the collective behavior must be able to adapt to changing conditions. For this reason, hybrid

approaches are being explored.

c) *Hybrid design*: a modern approach involves combining manual and automatic methods to achieve: i) comprehensibility of the high-level behavior to be generated, ii) the ability to generate more complex behaviors, and iii) mitigation of the problems of sub-symbolic techniques by “guiding” learning. In this direction, various works have been done using aggregate computing as the reference language (given its characteristics that allow managing scale, etc.) and many-agent reinforcement learning as it is capable of learning policies in such systems [22]. This combination can occur at different levels, at the platform level to improve execution efficiency [23], at the language level to program only complex parts of the program (via program sketching [24]), or use AC to enhance the learning process with graph neural networks [25]. This is a new line of research that still needs to be explored but has already yielded some interesting initial results.

### A. Challenges – RQ3

a) *Inherent complexity*: the inherent complexity of CPSWs poses a significant challenge in their design and implementation. The scale of these systems, coupled with the heterogeneity of the devices and the need for self-organization, makes it difficult to develop effective algorithms and control mechanisms.

b) *Dynamic environments*: CPSWs often operate in dynamic environments, where conditions can change rapidly and unpredictably. This necessitates the development of algorithms and control mechanisms that can adapt to changing conditions in real-time. Moreover, the system must be capable of self-organizing to maintain its integrity and performance in the face of these changes.

c) *Scalability*: the scalability of CPSWs is a critical challenge, particularly in terms of developing algorithms and control mechanisms that can operate effectively across networks of varying sizes. Moreover, the system must be capable of accommodating new nodes dynamically, without compromising its functionality.

d) *Interoperability*: the heterogeneity of devices within a CPSW introduces significant challenges in terms of interoperability. Devices from different vendors may employ divergent communication protocols or data formats, making it difficult to achieve seamless interaction. This necessitates the development of a common framework that enables effective communication and collaboration among the devices.

## VI. CONCLUSION

In this paper, we have extensively explored the concept of cyber-physical swarms (CPSWs), a novel and rapidly evolving field in distributed systems.

These systems present a paradigm shift in how we perceive and engineer distributed networks. The emergence of CPSWs has opened new horizons in various domains, from swarm robotics to smart city infrastructure and beyond, demonstrating remarkable potential in addressing complex, large-scale problems with unprecedented efficiency and adaptability.

Throughout our discussion, we have identified and delved into the unique characteristics that define CPSWs which not only distinguish CPSWs from traditional distributed systems but also pose specific challenges in their design and implementation. We addressed these challenges by presenting various engineering approaches, including automatic, manual, and hybrid designs, each offering distinct perspectives in tackling the complexity inherent in CPSWs.

Additionally, we have drawn parallels and distinctions between CPSWs and related concepts. This comparative analysis has helped to contextualize CPSWs within the broader landscape of distributed systems, highlighting their unique position and potential for future development.

Cyber-physical swarms are still in their nascent stages, and much research is required to fully harness the potential of these systems. Future work should focus on refining the engineering approaches, developing more robust and scalable algorithms, and exploring new applications where CPSWs can make a significant impact. Moreover, non-functional concerns such as security, privacy, and ethical considerations should be addressed to ensure the responsible development and deployment of CPSWs.

## VII. ACKNOWLEDGMENTS

This work was supported by the Italian PRIN project “CommonWears” (2020HCWWLP).

## REFERENCES

- [1] Wikipedia contributors, “2023 canadian wildfires — Wikipedia, the free encyclopedia,” [https://en.wikipedia.org/w/index.php?title=2023\\_Canadian\\_wildfires&oldid=1199259781](https://en.wikipedia.org/w/index.php?title=2023_Canadian_wildfires&oldid=1199259781), 2024, [Online; accessed 30-January-2024].
- [2] V. Barbuto, C. Savaglio, M. Chen, and G. Fortino, “Disclosing edge intelligence: A systematic meta-survey,” *Big Data and Cognitive Computing*, vol. 7, no. 1, p. 44, 2023.
- [3] T. De Wolf and T. Holvoet, “Emergence versus self-organisation: Different concepts but promising when combined,” in *International workshop on engineering self-organising applications*. Springer, 2004, pp. 1–15.
- [4] N. T. Nguyen, R. Kowalczyk, S.-M. Chen *et al.*, *Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems: First International Conference, ICCCI 2009, Wrocław, Poland, October 5-7, 2009. Proceedings*. Springer, 2009.
- [5] M. Dorigo, E. Bonabeau, and G. Theraulaz, “Ant algorithms and stigmergy,” *Future Gener. Comput. Syst.*, vol. 16, no. 8, pp. 851–871, 2000. [Online]. Available: [https://doi.org/10.1016/S0167-739X\(00\)00042-X](https://doi.org/10.1016/S0167-739X(00)00042-X)
- [6] M. Dorigo, V. Maniezzo, and A. Colnari, “Ant system: optimization by a colony of cooperating agents,” *IEEE Trans. Syst. Man Cybern. Part B*, vol. 26, no. 1, pp. 29–41, 1996. [Online]. Available: <https://doi.org/10.1109/3477.484436>
- [7] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of International Conference on Neural Networks (ICNN’95)*, Perth, WA, Australia, November 27 - December 1, 1995. IEEE, 1995, pp. 1942–1948. [Online]. Available: <https://doi.org/10.1109/ICNN.1995.488968>
- [8] F. R. Fulginei and A. Salvini, “The flock of starlings optimization: Influence of topological rules on the collective behavior of swarm intelligence,” in *Computational Methods for the Innovative Design of Electrical Devices*, ser. Studies in Computational Intelligence, S. Wiak and E. Napieralska-Juszczak, Eds., 2011, vol. 327, pp. 129–145. [Online]. Available: [https://doi.org/10.1007/978-3-642-16225-1\\_7](https://doi.org/10.1007/978-3-642-16225-1_7)
- [9] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, “Swarm robotics: a review from the swarm engineering perspective,” *Swarm Intell.*, vol. 7, no. 1, pp. 1–41, 2013. [Online]. Available: <https://doi.org/10.1007/s11721-012-0075-2>
- [10] G. Francesca and M. Birattari, “Automatic design of robot swarms: Achievements and challenges,” *Frontiers Robotics AI*, vol. 3, p. 29, 2016. [Online]. Available: <https://doi.org/10.3389/frobt.2016.00029>
- [11] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018.
- [12] Y. Yang, “Many-agent reinforcement learning,” Ph.D. dissertation, UCL (University College London), 2021.
- [13] A. Šošić, W. R. KhudaBukhsh, A. M. Zoubir, and H. Koepl, “Inverse reinforcement learning in swarm systems,” *arXiv preprint arXiv:1602.05450*, 2016.
- [14] H. Gu, X. Guo, X. Wei, and R. Xu, “Mean-field multi-agent reinforcement learning: A decentralized network approach,” *CoRR*, vol. abs/2108.02731, 2021. [Online]. Available: <https://arxiv.org/abs/2108.02731>
- [15] A. Oroojlooyjadid and D. Hajinezhad, “A review of cooperative multi-agent deep reinforcement learning,” *CoRR*, vol. abs/1908.03963, 2019. [Online]. Available: <http://arxiv.org/abs/1908.03963>
- [16] G. Mone, “Rise of the swarm,” *Commun. ACM*, vol. 56, no. 3, pp. 16–17, 2013. [Online]. Available: <https://doi.org/10.1145/2428556.2428562>
- [17] C. Pinciroli and G. Beltrame, “Buzz: A programming language for robot swarms,” *IEEE Softw.*, vol. 33, no. 4, pp. 97–100, 2016. [Online]. Available: <https://doi.org/10.1109/MS.2016.95>
- [18] G. Theraulaz and E. Bonabeau, “A brief history of stigmergy,” *Artificial life*, vol. 5, no. 2, pp. 97–116, 1999.
- [19] M. Mamei, F. Zambonelli, and L. Leonardi, “Tuples on the air: A middleware for context-aware computing in dynamic networks,” in *23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings*. IEEE, 2003, pp. 342–347.
- [20] J. Beal, D. Pianini, and M. Viroli, “Aggregate programming for the internet of things,” *Computer*, vol. 48, no. 9, pp. 22–30, 2015.
- [21] R. Casadei, M. Viroli, G. Audrito, D. Pianini, and F. Damiani, “Engineering collective intelligence at the edge with aggregate processes,” *Engineering Applications of Artificial Intelligence*, vol. 97, p. 104081, 2021.
- [22] G. Aguzzi, R. Casadei, and M. Viroli, “Machine learning for aggregate computing: a research roadmap,” in *2022 IEEE 42nd International Conference on Distributed Computing Systems Workshops (ICDCSW)*. IEEE, 2022, pp. 119–124.
- [23] —, “Addressing collective computations efficiency: Towards a platform-level reinforcement learning approach,” in *2022 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)*. IEEE, 2022, pp. 11–20.
- [24] —, “Towards reinforcement learning-based aggregate computing,” in *International Conference on Coordination Languages and Models*. Springer, 2022, pp. 72–91.
- [25] G. Aguzzi, M. Viroli, and L. Esterle, “Field-informed reinforcement learning of collective tasks with graph neural networks,” in *2023 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)*. IEEE, 2023, pp. 37–46.