Multi-agent systems

Mohammad Rahmani

MAS are composed of multiple autonomous, **interacting agents** that have **common** or **conflicting goals** and **sensory information**. MAS are generally **decentralized**(decision making), **asynchronous** systems but can sometimes be **centralized or hybrid** (Shoham and Leyton-Brown, 2009). Agents **communicate** with each other, and **make decisions** on their own in order to achieve their individual as well as collective goals. To this end they are equipped with a 'brain'(Ciprich et al., 2008).

1 Open MAS

In open multi-agent systems (MASs), agents are independently designed and act according to their own interests. Norms can be applied to regulate such systems influencing and restricting the behavior of their agents but not directly interfering with their autonomy (Santos et al., 2018).

2 Interaction

Complexity Considering agent interaction complexity leads to three classes of MAS: Ngobye et al. (2010) based on :

Types

- no direct interaction
- simple interaction
- complex conditional interaction

In addition to the complexity of interactions, MAS can exhibit different types of interactions based on agent (Rizk et al. (2018)[43], [44])

- goals
- resources
- skills
- Positive: Help each other accomplish their goals

  - **collective**: Agents are unaware of other agents' existence but share a common goal and each agent contributes to its completion, as in robot formation control and foraging.
  - **collaborative**: In collaborative interaction, agents do not have common goals but help each other accomplish their individual goals.
  - **Cooperative**: Cooperative interaction is similar to collective interaction except that agents are aware of other agents' existence. Examples include search and rescue, exploration, and object displacement. It is considered by some one of the more challenging interactions due to the need for high correlation and synchronization between agents and time sensitivity of agents' actions, especially in robotics (Rizk et al., 2018)
  - **Coordinative**: Agents within an environment work together to **minimize interference** and complete their individual goals; MRS(Multi Robotic Systems) path planning is one example.

- **Negative**: Agents impede each other. Negative interaction can be either
  - **conflicting** where agents do not have enough resources to complete their goals and fight for external resources.
  - **competitive** where agents have conflicting goals.

2.1 Communication

The communication section of Yan et al. (2013) offers a taxonomy of existing communication types. Communication is a mode of interaction between agents (Yan et al., 2013). Arai et al. (2002) the communication section, survey.

Communication types

- **Explicit**: Explicit communication refers to the means for the direct exchange of information between the robots, which can be made in the form of **unicast** or **broadcast** intentional messages Yan et al. (2013).
- **Implicit**: Implicit communication refers to the way in which the robot gets information about other robots in the system through the environment.Implicit communication can also be divided into two categories: active implicit communication (e.g., interaction via the environment) and passive implicit communication (e.g., interaction via sensing) Yan et al. (2013)

Another communication Type

- Direct
- Indirect

Another

- Spars

Another

- Local

Another

- Communication Free systems

Agent-communication language `https://en.wikipedia.org/wiki/Agent_Communications_Language`

Communication channel

- Discreet

From centralization point of view

- Centralized
- Decentralized

2.1.1 Natural language

Human NL to action: As the first stage in Section **??** is reception of requests in NL so that the machine can form word embedding, then in this part some related works are presented. Doubleday et al. (2016) describes extensions to Language Communication with Autonomous Systems (LCAS) implementation for the control of autonomous systems by NL, to enable such systems to handle incoming language requests regarding actions. Chai (2018) gives a brief introduction to interactive task learning where humans can teach physical agents new tasks through NL communication and action demonstration.

Human verbal interaction with an IA: Although the main subject of this idea is IA-IA interaction but verbal human interaction can also provide us with a lot of applicable information. A survey over such verbal and none verbal methods is presented in Mavridis (2015).

3 Task allocation

4 Task decomposition

5 Coordination

Is a probable result of some interaction. I believe coordination might be a result of interaction.

6 Self-organization

7 Conflict resolution

Surveys Santos et al. (2018) presented a literature review on normative conflict resolution in Open MAS. W. Alshabi (2007) presents another survey with this regard.

Individuals Ferrera et al. (2014) presents an scalable, decentralized and reactive approach for collision avoidance is presented. Jiang et al. (2019) formulate the problem of multi-robot planning with conflicts and synergies (mrpcs), and develop a multi-robot planning framework, called iterative inter-dependent planning (iidp), for representing and solving mrpcs problems. Kutzke and Bays (2019) presents a schedule deconfliction algorithm to enhance cooperation among heterogeneous teams of autonomous agents in situations where communication is severely limited. The proposed solution, which they call the Generous Agent Algorithm (GAA), encodes human-like negotiations during conflict resolution. Individual agents propose modified schedules, vacating their most costly task from their schedules, while simultaneously augmenting their schedules with another agent's vacated task.

8 Swarm intelligence

Swarm intelligence describes the behavior of **decentralized cooperative agents**, whether natural or artificial, working toward a common global goal. **Self-organized and distributed behavior of locally aware and locally interacting agents are pillars of swarm intelligence**. Systems modeled in this fashion generally consist of many autonomous but homogeneous agents implementing simple rules with agent interactions restricted to local neighborhoods.

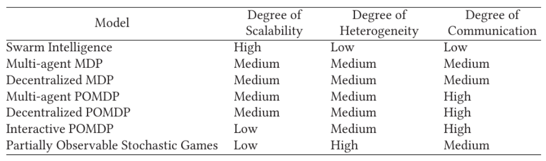
9 Decision making/ Planning

MAS planning and control, also known as decision making, is a main module in MAS. It determines the **sequence of actions**, or **policy**, that agents should perform to complete their assigned task once the complex tasks have been decomposed to sub-tasks and allocated to cooperating groups of agents. Rizk et al. (2019)

Pros and cons of each strategy is presented in Figure 1

communication vs heterogeneity is presented in Figure 2

Table 3. Comparison of Decision-making Models



Model	Degree of Scalability	Degree of Heterogeneity	Degree of Communication
Swarm Intelligence	High	Low	Low
Multi-agent MDP	Medium	Medium	Medium
Decentralized MDP	Medium	Medium	Medium
Multi-agent POMDP	Medium	Medium	High
Decentralized POMDP	Medium	Medium	High
Interactive POMDP	Low	Medium	High
Partially Observable Stochastic Games	Low	High	Medium

Figure 1:

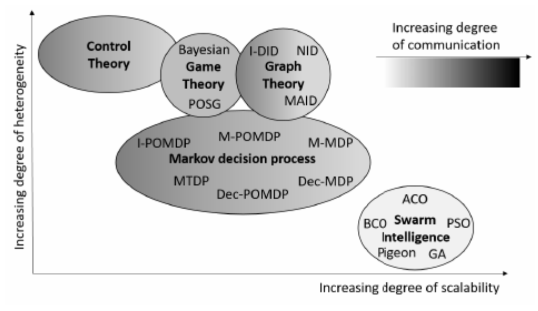

Increasing degree of heterogeneity
Control Theory
Bayesian Game Theory POSG
I-DID NID Graph Theory MAID
Increasing degree of communication
I-POMDP
M-POMDP
M-MDP
Markov decision process
MTDP
Dec-POMDP
Dec-MDP
ACO
BCO Swarm Intelligence PSO
Pigeon GA
Increasing degree of scalability

Fig. 2. Comparison of decision making model frameworks.

Figure 2:

Centrality : Learning policies in MAS decision making:

- Centralized: learns policies for all agents in the system
- Decentralized: each agent learns its own policies in parallel to other agents

Evaluation see artificial intelligence decision making and also this one in challenges

Credit assignment and Communication factor

- Credit assignment: How to distribute rewards among cooperating agents, is one problem that arises and should be appropriately handled to achieve optimal performance.
- Communication, whether direct or indirect, is another issue in cooperative decision making that should be considered

9.1 Reinforcement learning RL

Multi-agent RL allows cooperative MAS to complete tasks with minimal communication overhead by using the global immediate reward instead of the individual agent immediate reward in the Q-learning algorithm to solve repeated games Rizk et al. (2019)[234]. RL is used in Game theory and Markov Decision Process(MDP), mostly as a solution to the problems.

9.2 Markov Decision Process(MDP)

9.2.1 M-MDP

Multiagent MDP (M-MDP) extends MDP to MAS by assuming a **joint action space** with a **team reward model** and **fully observable environment**. A central learner learns a vector of actions that should be performed by the agents and the reward is common to all agents Rizk et al. (2018)[37].
The worst-case complexity of finite horizon M-MDP is **P-complete** Rizk et al. (2018)[68] which is solvable in **polynomial time** by a **Turing machine**, an abstract model of computing devices. As the number of agents increases, the joint state and action spaces' dimensionalities increase exponentially.
To ease the computational burden, independence is assumed to make objective functions factorable.
Solving the problem iteratively also reduces the computational complexity.
Distributed implementations of the central learner have been developed for factorable objective functions (Rizk et al., 2018)-¿[69].

9.2.2 DEC-MDP

Decentralized MDP (dec-MDP) assumes an independent action space with local reward and jointly fully observable environments (Rizk et al., 2018)[37]. In other words, individual agents view a partially observable environment but the aggregate observations of all agents in the MAS make the environment fully observable.

Finite-Horizon DEC-MDP was proven to be worst case NEXP-complete (solvable in exponential time using a nondeterministic Turning machine), when three or more agents are considered Rizk et al. (2018)[70].
Since actions and rewards are local, this approach falls under the concurrent learning class of MAS learning.
Assuming agent observations and transitions are independent, the model is known as **TI dec-MDP** and its complexity is NP-complete, meaning a solution can be found in polynomial time by a nondeterministic Turing machine. This model can be further simplified by assuming independent rewards to obtain a **P-complete** complexity in the worst case.

9.2.3 Partially observable MDP(POMDP)

Multiagent POMDP extends M-MDP to partially observable environments, and is **PSAPCE-complete** which means the algorithm's **memory requirements are polynomial function of the input size**. Like M-MDP, it is a team learning approach that has a central learner, and employed Bayesian RL framework to learn policies Rizk et al. (2018)[99].

Dec-POMDP Generalizes POMDP to MAS where rewards are common and based on joint actions but observations are individualistic Rizk et al. (2018)[9]. The goal is to maximize the reward of the entire system as agents collaborate to achieve a common task. Communication among agents can be explicit (Dec-POMDP-COM) or implicit (Dec-POMDP). This model is NEXP-complete Rizk et al. (2018)[70]. Approximate solutions have been proposed based on bounded policy iteration Rizk et al. (2018)[90], Q-value function methods [91], multiagent A* Rizk et al. (2018)[81], genetic algorithms Rizk et al. (2018)[82], DP Rizk et al. (2018)[92]–[94], and a Bayesian learning, stick-breaking policy algorithm Rizk et al. (2018)[95]. A set of approximate inferences and heuristics including bootstrapping were used to find approximate solutions to Dec-POMDP in Rizk et al. (2018)[96].

Networked distributed POMDP (ND-POMDP) assumes **local interaction among agents** to reduce the computational cost of finding policies Rizk et al. (2018)[100]. ND-POMDP is a factored Dec-POMDP model where observations and transitions are independent and rewards are divided among neighboring agents. Its worst case computational complexity is NEXP-complete. Algorithms used to find policies for this model include multiagent RL Rizk et al. (2018)[101], DP Rizk et al. (2018)[102], and distributed constrained optimization Rizk et al. (2018)[100].

Interactive POMDP (I-POMDP) Interactive POMDP (I-POMDP), a concurrent learning approach, generalizes POMDP to MAS by modeling other agents in the system while maintaining a belief of the system state Rizk et al. (2018)[103].

Finitely nested I-POMDP is PSPACE-complete Rizk et al. (2018)[98] and approximate solutions have been proposed based on particle filters Rizk et al. (2018)[104], value iteration Rizk et al. (2018)[105], and Monte Carlo sampling methods Rizk et al. (2018)[106].

9.2.4 Insight

MDP and its variants have been widely adopted in many **complex** MAS decision making problems, despite the very **restrictive** Markovian assumption. Even though these models do **not scale well**, they are able to handle agent **heterogeneity**. Deep learning has allowed the extension of MDPs from the discrete space to the **continuous space**, which is more suitable for **robotic MAS**.

9.3 Game Theory

9.3.1 Partially Observable Stochastic Games(POSG)

(See decision-making) POSG have been used to model learning sequential decision making in cooperative MAS (Rizk et al., 2018)-¿[115]. POSG have also been used to model cooperative MAS decision making in partially observable Markovian environments (Rizk et al., 2018)-¿[115], (Rizk et al., 2018)-¿[117]. However, this model's solution is intractable as the number of agents increases. Therefore, an approximate solution was computed based on Bayesian games to achieve decentralized control in robot teams with limited communication. The algorithm was validated on the two-robot tag problem, two-agent lady, and tiger problem and multiple access broadcast channel problems.

9.3.2 Application of RL

(First see RL application in Decision making and the full text is being taken from Rizk et al. (2019)) Multi-agent RL allows cooperative MAS to complete tasks with minimal communication overhead by using the global immediate reward instead of the individual agent immediate reward in the Q-learning algorithm to solve repeated games Rizk et al. (2019)[234]. Validation on box pushing and sensor distribution demonstrated the superior performance of this algorithm compared to other approached.
Sparse interaction to negotiate equilibrium sets and transfer knowledge in multi-agent RL reduced computational complexity and led to better coordination and scalability, as shown by simulations on grid world games and robots shelving items in a warehouse Rizk et al. (2019)[240].
Information sharing has been modeled as a MDP to reduce communication overhead without affecting performance Rizk et al. (2019)[12].

9.3.3 Bayesian games

(See decision-making -¿ Bayesian games)

9.3.4 insight

While game theoretic approaches had been mainly used in **competitive MAS**, some models have gained popularity in **cooperative MAS** due to the agents' capabilities of modeling other agents in the game. This property can be useful in robotic systems where robots are unable to communicate with others. However, this restricts the number of agents in the system due to increasing computational costs. Game theory's systematic mathematical approach has been an attractive quality for many applications but combining it with some heuristic approaches such as deep learning might lead to improved performance in robotic applications and others.

- Advantages

  –

- Disadvantages

  –

9.4 Swarm intelligence

Swarm intelligence describes the behavior of decentralized(see decision making) cooperative agents, whether natural or artificial, working toward a common global goal.Self-organized and distributed behavior of locally aware and locally interacting agents are pillars of swarm intelligence. Systems modeled in this fashion generally consist of many **autonomous** but **homogeneous agents** implementing **simple rules** with **agent interactions restricted to local neighborhoods**.

9.4.1 inspired by biology

Bee colony behavior Many algorithms have been inspired by bee colony behavior. Bee colony optimization is based on **direct communication** among agents performing a series of moves for a certain duration based on the strength or fitness of the solution, also known as "waggle dancing." This recruits other agents to the most fit solution.

Navigation is based on path integration where agents continuously update a vector indicating the position of the start location.

Ant colony optimization (ACO) , inspired by ant colony behavior, is a class of algorithms that rely on **indirect communication. Navigation** is based on depositing pheromones along the trail. A more fit solution results in stronger pheromones on the trail that lead to recruiting more agents.

PSO(particle swarm optimization) is inspired by **flocks of bird** and **schools of fish** (Rizk et al., 2018)[124]. Agents **navigate** the environment searching for better solutions using principles from birds' movements. A **pigeon inspired optimization** algorithm relied on the **magnetic field**, **sun** and **landmarks** to **achieve path planning** (Rizk et al., 2018)[125].
Distributed implementations of ACO (Rizk et al., 2018)[126], [127], and **PSO** (Rizk et al., 2018)[128] have been developed to **speedup convergence**.

9.4.2 Insight

While such systems exhibit desirable properties like **robustness**, **flexibility**, **scalability**, low **complexity**, **inherent parallelism**, and **fault tolerance** (Rizk et al., 2018)[11], [129], they have important limitations. Most swarm systems consist of **identical agents**, leading to their limitations according to Rizk et al. (2018)[129]. The agents must be **homogeneous** or can be **divided**

into a small number of homogeneous clusters following simple rules to make decisions.
However, there are many applications, such as search and rescue operations, that require heterogeneous, complex agents working toward a common goal.

9.5 Graph theory

Decision making in MAS have been modeled as graphs with nodes representing agents and edges representing interactions and information flow among agents Rizk et al. (2018)[130].

9.5.1 Influence diagram(ID)

Decision nodes (rectangles) represent choices available to the agent and utility nodes (diamonds) compute the utility of these choices. The action with the highest utility is chosen.

Multiagent IDs (MAIDs) generalizes IDs to MAS by generating decision rules that depend on decision rules made by other agents Rizk et al. (2018)[134]. This is graphically represented by connecting decision nodes that depend on each other; a directed relevance graph is thus produced.

MAIDs represent games with imperfect information graphically and are an alternative to the normal and extensive forms of game representation Rizk et al. (2018)[133]. They can be converted either to extensive form games or to IDs and then solved.

Network of IDs (NIDs) A network of IDs (NIDs) is built on top of MAIDs to account for uncertainties in other agents' decision making and hierarchy of beliefs Rizk et al. (2018)[135]. This formalism can represent irrational behavior and distinguishes between different agent models in the systems, i.e., it does not treat all other agents identically. (Both MAID and NID are applicable to episodic decision making only.)

Acyclic NIDs can be solved using a bottom up approach by converting each block to a MAID and solving it. Duplicates are included to account for beliefs about others' strategies.

Cyclic NIDs are converted to acyclic NIDs and solved.

Interactive DIDs (D stands for dynamic see decision making)were proposed in Rizk et al. (2018)[133] as an MAS extension of DIDs and can be viewed as computational counterparts of I-POMDP. Models of other agents are *clustered to reduce computational complexity* but lead to *approximate solutions.*

9.6 Insight

Advantages Graph theory models the interaction of agents, allowing them to exchange information and make decision accordingly.

Disadvantages The computational complexity of this approach increases exponentially in densely connected graphs with many nodes (agents). The main benefits of graph theory in MAS come from combining it with other approaches such as MDPs and control theory (discussed next) to extend these approaches to MAS. Furthermore, exploiting special structures such as **sparsely connected dense subgraphs** are a common approach to reduce computational cost and improve performance.

9.7 Control Theory

As automation problems became more complex, researchers extended control theory to MAS by developing distributed controllers.

9.7.1 Distributed Cooperative Control

Distributed controllers are designed by **combining** concepts from **control** and **graph theory**.

Specifically, interactions among agents are modeled using graph theory and the control problem is decomposed among the agents to obtain a distributed controller.

The amount of communication among agents is dependent on the design of the distributed controller and can vary based on the nature and complexity of the task (whether it is easy decomposable), the optimality of the control algorithm and other factors.

Since many controllers are based on **optimization algorithms**, distributed optimization is an integral part of distributed control Rizk et al. (2018)[136]. Unlike other approaches, distributed cooperative controllers designed using control and graph theory can be mathematical validated to prove optimality, stability, robustness, and convergence, to name a few properties. Distributed controllers have been applied to various control problems. For example, a **Lyapunov-based** voltage and frequency controller was designed for micro-grid systems that only requires local **communication among neighbors** Rizk et al. (2018)[137] and a secondary voltage distributed controller based on input–output feedback linearization that requires sparse communication Rizk et al. (2018)[138].
A Lyapunov-based distributed lead-follower control system was developed that scaled to large MAS when the interaction topology is an undirected graph Rizk et al. (2018)[139]. Distributed consensus tracking was achieved by designing: distributed adaptive controllers in weakly connected, directed graphs Rizk et al. (2018)[140], a distributed optimal control algorithm Rizk et al. (2018)[141], and nonlinear distributed impulsive control (control signals are given as impulses instead of continuously) with delayed impulses in undirected graphs Rizk et al.

(2018)[142]. Stochastic sampling in leader-follower consensus problems has been shown to improve scalability of MAS Rizk et al. (2018)[143]. Distributed impulsive control has also been applied to heterogeneous MAS synchronization problems Rizk et al. (2018)[144]. Rizk et al. (2018)[145] proposed distributed output regularization using adaptive control in MAS with a switching topology. Other applications include formation control Rizk et al. (2018)[146] and navigation Rizk et al. (2018)[147] in MRS.

9.7.2 insight

Advantages Adopts systematic mathematical approaches to develop controllers,

Disadvantages some systems are simply too complex and intractable for such methods. For example, most algorithms assume linear systems. Therefore, data-driven methods such as those in distributed artificial intelligence are necessary to automate certain complex tasks in real-world environments.

Overall Nevertheless, distributed cooperative controllers are necessary in some applications where <u>sufficient data</u> is not available or <u>mathematically proven optimal controllers are crucial</u> like in <u>aviation</u> or <u>military domains</u>.

9.8 Application

9.8.1 Robotics

see robotics decision making section

9.8.2 Repeated Coalition Formation

9.8.3 Intelligent Transport Networks

9.8.4 Wireless Sensor Networks

9.8.5 Intrusion Detection

9.8.6 Other Applications

9.9 Challenges

9.9.1 Scalability

Decision making algorithms should be scalable, especially in heterogeneous MAS, to accomplish more complex tasks. Scalability relies on

- homogeneity
- level of interaction

Swarm intelligence can scale to large MAS since agents are homogeneous and interaction is minimal and restricted to the agent's neighborhood.

MDP and Game theoretic **MDP** variants and **game theoretic** models do **not** scale well since the complexity of the algorithm increases exponentially due to the model formulation that results in exponentially large state spaces. Using the **graph theoretic** formulation for large MAS results in **densely connected graphs** which are **computationally expensive**.

9.9.2 Computational Complexity

Causes:

- **real-time decision making** in some applications or the
- **lack of enough computational resources** of agents.
- Agent **interactions** MAS increases the computational cost per agent as the number of agents increases, especially in methods that extend single agent models to MAS if careful consideration of interaction cost is not performed.
- **Tightly coordinated** tasks also increase the computational burden due to the large amount of communication and data exchange among agents.

9.9.3 Dynamic Environments

The environment's dynamic and **unpredictable** nature makes it difficult to **foresee**, **design** and **test** an agent that can handle all these situations. Therefore, decision making algorithms should **generalize** well to situations that have not been learned or tested.

They should be able to adapt to the **dynamic environment** and various **uncertainties** it might encounter and should be robust to **noisy** and **incomplete information** generated by sensors, and **nondeterministic** actions.

POMDP, **IDs**, **POSG**, and **Bayesian games** are better suited to handle uncertainties than MDP and its variants that assume fully observable environments, because they account for **partially observable environments**, incomplete and **imperfect information** in their algorithms. **Agent failures** are also a source of uncertainty in MAS that hinder the completion of tasks.

Unlike other models, **swarm intelligence** models are **better** suited to handle **agent failures** due to the homogeneous nature of agents and minimal interaction necessary. However, this is still an issue that needs to be considered whenever MAS are designed.

9.9.4 System Heterogeneity

Advantageous of heterogeneous systems:

- can deal better with **environment diversity**
- can deal better with **complex tasks**

Disadvantages

- makes cooperative decision making more complex
- agents need to model other agents when capability uncertainty exists
- agent capabilities should be compatible
- agents should have or achieve a common language to communicate and interact

Swarm intelligence simplifies modeling by assuming all agents are **homogeneous**.

Graph theoretic models, **POSG** and its subclasses can handle heterogeneous MAS if the **state** and **observation spaces** are designed appropriately.

I-POMDP and **I-DID** inherently model other agents, making them better than other graph and game theoretic models in dealing with MAS heterogeneity.

9.9.5 Big Data

Algorithms that model and generate representations of such big data like convolutional neural networks (deep learning) produce **computationally expensive** models that are not suitable for computationally limited agents or decision making algorithms whose computational cost grows exponentially with the **dimensionality** of the data.

Yet, allowing agents to access these models through the **cloud** has its own complications with respect to **cloud accessibility**, **bandwidth constraints**, **representation compatibility**, **privacy**, and **security**.

9.9.6 Evaluation Standards

Decision-making algorithms are generally evaluated based on policy optimality and their time and space complexity. Evaluation standards are necessary in MAS decision making to compare proposed algorithms and assess the state-of-the-art.

General metrics

- solution optimality
- algorithm completeness
- algorithm time
- space complexity

Braubach et al. Rizk et al. (2018)[257] developed abstract metrics that would be specialized for MAS applications, and include

- function (e.g., restrictions)
- usability (e.g., simplicity)
- operating ability (e.g., performance)
- pragmatic metrics (e.g., installation)

Lass et al. Rizk et al. (2018)[258] distinguished between two metric categories:

- effectiveness (e.g., success, failure, and 90% accuracy)
- performance (e.g., resource consumption and time complexity)

that could be applied to four MAS levels:

- agent
- framework
- platform
- host

They presented a framework to select appropriate metrics for a given application and performed a case study on a distributed constrained optimization problem. Di Bitonto et al. Rizk et al. (2018)[259] developed a hierarchical metric system where both *interagent (communication and cooperation)* and *intra-agent metrics* measured

- environment complexity
- agent rationality
- autonomy
- reactivity
- adaptability

This system was tested on a knowledge management problem for the automotive industry with two agents only.

Marir et al. Rizk et al. (2018)[260] proposed an evaluation platform that included metrics like average of communication load and validated the platform on an auctioning problem.

Nevertheless, standards to evaluate and compare the performance of MAS on real-world environments are still underdeveloped. Existing metrics have been tested on a hand-full of narrow-scoped scenarios that did not necessarily include robot agents.

9.10 Human in the loop

In MRS The term human-in-the-loop refers to a system architecture that requires robots or agents to interact with humans. Humans' involvement can range from giving agents instructions to executing actions alongside other robots and tele-operating vehicles. The benefits of such systems include expanding the scope of tasks MRS can execute without achieving full autonomy, complementing the skills of MRS with those of humans to efficiently execute certain tasks, giving humans more control over the system and improving system adaptability and resistance to environmental stochasticity. However, this system is also faced with many challenges from communication to responsiveness and fault tolerance. Communicating with agents via text requires the agents to perform natural language processing to parse messages and text generation to send messages. Communicating via speech imposes additional complexity by requiring the integrations of speech recognition and synthesis modules. While these fields have seen significant strides in recent years with the emergence of deep learning and big data, embodied by systems like Siri, Amazon Alexa, and others Rizk et al. (2019)[129], accuracy, computational complexity, and responsiveness are still issues that need improvements before incorporating them into more complex systems with coordination, synchronization, and time constraints Rizk et al. (2019)[89]. The responsiveness of the system will suffer when incorporating humans in the loop due to the additional communication overhead and human performance variability. Finally, such systems may be more error prone and could be affected by human fatigue and distractions. Therefore, determining whether human-in-the-loop is an asset or liability in a given scenario is key to choosing the right system architecture that would lead to the successful completion of tasks.

9.10.1 Transfer Learning

(In MRS) The highly stochastic nature of real-world environments and the variability in agent capabilities within heterogeneous MRS imply that a given MRS will rarely encounter two identical scenarios. However, MRS will encounter many similar scenarios. Therefore, leveraging previous experiences to improve current decision making would significantly improve the performance, adaptability, and robustness of MRS. Transfer learning is a learning paradigm that allows agents to jump-start their learning by transferring knowledge from previous experiences to current reinforcement learning problems instead of learning from scratch during every new scenario Rizk et al. (2019)[213]. Transfer learning has been applied to MAS Rizk et al. (2019)[151, 212] and MRS Rizk et al. (2019)[50] but has yet to be tested in end-to-end systems deployed in the real world on complex tasks. Furthermore, transfer learning for RL has been mainly tested on benchmark problems and gaming environments but not in real-world environments. Therefore, this field of research, which started gaining traction less than two decades ago Rizk et al. (2019)[213], still has many open problems that must be addressed before successful integra tion into MRS in the real world can be realized.

9.10.2 Lack of a Unified Framework

In MRS - Significant contributions have been made to the various modules in Figure Rizk et al. (2019)2. However, one reason that has hindered the successful deployment of fully automated MRS is the fact that most of the work has mainly dealt with these modules independently. Taking a more holistic approach by viewing these research fields as connected within a larger field is necessary to take steps toward successful MRS deployments. Furthermore, feedback connections between the different modules should be incorporated into the workflow to further improve its efficacy. For example, task decomposition should be influenced by the capabilities of the agents and coalitions, as well as the environment's state estimated by the perception module and vice versa. End-to-end simulations and real-world experiments will also help in improving the formulated models by identifying major weaknesses preventing successful deployment.

9.10.3 Other Challenges

- Task complexity poses a challenge for decision making algorithms because they do not have the capability of recognizing what tasks can be decomposed into simpler tasks that they can complete. Adding this capability to decision making algorithms in MAS in addition to dynamically recognizing what tasks require tight coordination and what tasks can be accomplished with minimal interaction among agents will increase the scope of automated tasks.

- Learning algorithms for decision making and perceiving agents should be autonomous. Reducing the number of manually tunable hyper parameters that require human intervention will allow algorithms to generalize better to unknown environments.

- (In MRS) Communication constraints and connectivity uncertainty further complicate things for cooperative MRS, especially for tightly coordinated problems. While connecting MRS to the cloud also allows us to reduce the computational load on these mobile devices and improve their performance Rizk et al. (2019)[222], the existence and stability of this connection is uncertain and may sometimes cripple the system instead of improving its performance.

- (In MRS) The time sensitivity of certain tasks and limited hardware resources of robots requires the development of efficient algorithms for decision making, perception, coalition formation, and task decomposition and allocation.

- (In MRS) Evaluation standards are needed to effectively compare the performance of MRS, as they are still underdeveloped.

9.11 Swarm intelligence

- Multi-objective particle swarm optimization (PSO) variants (see Rizk et al. (2018)-¿[17].)
- algorithms based on bees Rizk et al. (2018)-¿[18]
- meta-heuristic algorithms Rizk et al. (2018)-¿[19]
- artificial bee colony variants and applications Rizk et al. (2018)-¿[20]

9.11.1 Biologically Inspired Algorithms

Swarm intelligence was inspired by many social insects and animals including ants, bees, wasps, termites, bats, fish, and birds. In some ways, swarm intelligence is similar to RL; both are iterative algorithms that use a reinforcement signal to learn a solution Rizk et al. (2018)[121]. However, the reinforcement signal modifies the behavior of the agent differently in both algorithms. Many algorithms have been inspired by bee colony behavior. Bee colony optimization Rizk et al. (2018)[122] is based on direct communication among agents performing a series of moves for a certain duration based on the strength or fitness of the solution, also known as "waggle dancing." This recruits other agents to the most fit solution. Navigation is based on path integration where agents continuously update a vector indicating the position of the start location. Ant colony optimization (ACO), inspired by ant colony behavior, is a class of algorithms that rely on indirect communication Rizk et al. (2018)[123]. Navigation is based on deposit- ing pheromones along the trail. A more fit solution results in stronger pheromones on the trail that lead to recruiting more agents. PSO is inspired by flocks of bird and schools of fish Rizk et al. (2018)[124]. Agents navigate the environment searching for better solutions using principles from birds' movements. A pigeon inspired optimization algorithm relied on the magnetic field, sun and landmarks to achieve path planning Rizk et al. (2018)[125]. Distributed implementations of ACO Rizk et al. (2018)[126], [127], and PSO Rizk et al. (2018)[128] have been developed to speedup convergence.

9.11.2 Insight

While such systems exhibit desirable properties like robustness, flexibility, scalability, low complexity, inherent parallelism, and fault tolerance Rizk et al. (2018)[11], [129], they have important limitations. Most swarm systems consist of identical agents, leading to their limitations according to Rizk et al. (2018)[129]. The agents must be homogeneous or can be divided into a small number of homogeneous clusters following simple rules to make decisions. However, there are many applications, such as search and rescue operations, that require heterogeneous, complex agents working toward a common goal.

10 Terms and definitions

Homogeneity and heterogeneity Homogeneity and heterogeneity are concepts often used in the sciences and statistics relating to the uniformity in a substance or organism. A material or image that is homogeneous is uniform in composition or character (i.e. color, shape, size, weight, height, distribution, texture, language, income, disease, temperature, radioactivity, architectural design, etc.); one that is heterogeneous is distinctly nonuniform in one of these qualities.

Game theory the study of mathematical models of strategic interaction among rational decision-makers `https://en.wikipedia.org/wiki/Game_theory`

References

Tamio Arai, Enrico Pagello, and Lynne E. Parker. Guest editorial advances in multirobot systems. *IEEE Trans. Robotics and Automation*, 18(5):655–661, 2002.

Joyce Y. Chai. Language to action: Towards interactive task learning with physical agents. page 6. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM, 2018.

Nikola Ciprich, Marie Duzí, Tomás Frydrych, Ondrej Kohut, and Michal Kosinár. The architecture of an intelligent agent in MAS. 2008.

Steve Doubleday, Sean Trott, and Jerome Feldman. Processing natural language about ongoing actions. *CoRR*, abs/1607.06875, 2016.

Eduardo Ferrera, Ángel Rodríguez Castaño, Jesus Capitán, Pedro Marrón, and Anibal Ollero. Multi-robot operation system with conflict resolution. 2014.

Yuqian Jiang, Harel Yedidsion, Shiqi Zhang, Guni Sharon, and Peter Stone. Multi-robot planning with conflicts and synergies. 2019.

D. T. Kutzke and M. J. Bays. Conflict resolution for heterogeneous teams in communication-limited environments: A generous agent approach. In *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, 2019.

Nikolaos Mavridis. A review of verbal and non-verbal human-robot interactive communication. *Robotics Auton. Syst.*, 63:22–35, 2015.

Martin Ngobye, Wouter T. de Groot, and Theo P. van der Weide. Types and priorities of multi-agent system interactions. *Interdisciplinary Description of Complex Systems - scientific journal*, 8(1):49–58, 2010.

Yara Rizk, Mariette Awad, and Edward W. Tunstel. Decision making in multiagent systems: A survey. *IEEE Trans. Cogn. Dev. Syst.*, 10(3):514–529, 2018.

Yara Rizk, Mariette Awad, and Edward W. Tunstel. Cooperative heterogeneous multi-robot systems: A survey. *ACM Comput. Surv.*, 52(2):29:1–29:31, 2019.

Jessica Santos, Jean Zahn, Eduardo Silvestre, Viviane Silva, and Wamberto Vasconcelos. Detection and resolution of normative conflicts in multi-agent systems: A literature survey. 11 2018.

Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems - Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2009.

M. Itmi H. Abdulrab W. Alshabi, S. Ramaswamy. Coordination, cooperation and conflict resolution in multi-agent systems. 2007.

Zhi Yan, Nicolas Jouandeau, and Arab Ali Cherif. A survey and analysis of multi-robot coordination. *International Journal of Advanced Robotic Systems*, 10(12):399, 2013.