UNIVERSITAT
ROVIRA i VIRGILI

Introduction to Multi-Agent Systems

Coordination design

Authors:

Becerra Tomé, Alberto

Campàs Gené, Carla

Eduardo Bercowsky Rama, Andrés

González Béjar, Javier

Monfort Grau, Marc

Universitat Rovira i Virgili

Master's degree in Artificial Intelligence

Tarragona, December 2023

Contents

1 Overview

The activity has many implications concerning the communication of the agents involved in the environment. This includes communication between the different types of agents, for example, to enable the explorer to communicate the best possible path to take to each agent.

This activity is a Cooperative Distributed Problem Solving (CDPS) project, this means our agents will be built as benevolent agents rather than self-interested agents. These agents will be built to help each other achieve a common goal: e.g. the best interest of the agents is our best interest.

If our agents were to be placed in a competitive platform, where other teams had different agents within this platform and whoever got more treasure won the game, we would have to create a self-interested agent. This would allow us to apply the competitive angle to our agents.

To select the best possibilities for communication, coordination, and negotiation for agents we will initially explore the different types of communication categories, once these have been explored we will reason the need for each of these in our platform and select the communication that best fits our goal.

Through the following sections, we will address the steps involved with task and result sharing.

1.1 Problem Decomposition

The problem decomposition involves dividing the larger problem at hand into smaller tasks amongst the distributed agents.

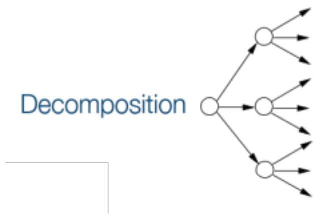

Figure 1: Problem Decomposition

Problem decomposition is a hierarchical process where the problem is divided into subproblems and the subproblems are again divided until we get a granularity that is ideal for our agents to be able to handle as single instructions in the system.

Following our initial proposal for implementation, we will decompose the problem using the explorer agents. Each of these will be able to communicate their solution while communicating with the other explorer systems.

1.2 Sub-problem Solution

The sub-problems are then to be solved to maximize the capabilities of all of our agents. We have to keep in mind a few different aspects to make sure that the solution we obtain makes sense for all the agents and optimizes the work to be done. We do this by maximizing the coherence metric in our system.

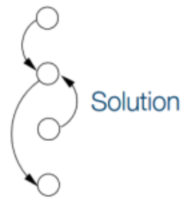

Figure 2: Problem Decomposition

Following our initial proposal for implementation, the explorer agent will trace the best possible paths with other agents. Each of these will be able to communicate their solution while communicating with the other explorer systems.

These solutions have to be consistent and make sure that each agent to whom we give instructions has a common goal. If these agents overlap in what treasure they are obtaining it will make our system inefficient and complicate our system. Therefore, there has to be constant communication between our agents to make sure that the plan of each of these agents makes sense.

1.3 Answer Synthesis

There are two activities to keep in mind when speaking about solution synthesis:

- **Task Sharing:** The sub-problems should be shared to the other agents in the form of tasks. In our system, the explorer will share the task with the collector and tanker agents.

- **Result Sharing:** Once the collectors have collected their items they have to share the status back to the explorers or nearing agents to make sure that these know the most current status of the treasures.

These two communication techniques will need to be achieved to make sure that the problem solution has been achieved. This is the inverse of decomposing the problem: We solve the problem one sub-problem at a time and join these recursively to solve the larger problem.

2 Coordination Techniques

Coordination in multi-agent systems refers to the ability of multiple agents to work together in a way that achieves overall system goals or objectives. In a multi-agent system, individual agents are autonomous entities with their own goals, capabilities, and possibly different perspectives or interests. Effective coordination ensures that these agents can collaborate, share information, and make decisions in a manner that optimises the overall system performance.

First, we will be explaining the theory of different coordination techniques studied. Then we will introduce the ones we plan to implement in our system, to help us solve the treasure hunt problem efficiently.

2.1 Partial Global Planning (PGP) / Generalized Partial Global Planning (GPGP)

Partial Global Planning (PGP) and its advanced version, Generalized Partial Global Planning (GPGP), represent innovative approaches in the domain of distributed planning within dynamic and unpredictable domains. These techniques emphasize the integration of planning and execution processes, differing from the traditional linear sequence of Planning, Scheduling, and Action.

2.1.1 Implementation Steps

The PGP/GPGP methodology unfolds in a series of structured steps:

1. **Local Plan Formulation:** Each agent independently develops a local plan based on its objectives, capabilities, and understanding of the environment. This plan reflects the agent's strategy for task execution and interaction with other agents.

2. **Plan Exchange:** Agents exchange their local plans with one another. This step is crucial for ensuring transparency and collaboration, as it allows agents to understand each other's intentions and operational strategies.

3. **Partial Global Plan Generation:** Based on the exchanged local plans, agents collectively generate partial global plans. These are comprehensive strategies that encompass the collective actions and interactions of multiple agents, aiming to align individual efforts towards a common goal.

4. **Optimization of Global Plans:** The final step involves refining and optimizing the partial global plans. This process seeks to enhance efficiency, reduce resource consumption, and address potential conflicts or redundancies between individual plans.

2.1.2 Advantages and Disadvantages

Advantages:

- **Flexibility in Dynamic Environments:** PGP/GPGP excels in environments where conditions and requirements rapidly change. Its inherent flexibility allows for swift adjustments to plans in response to new information or altered circumstances.

- **Enhanced Collaboration:** By facilitating the exchange of plans and fostering a collaborative planning process, these techniques promote a higher degree of cooperation among agents, leading to more coherent and unified action.

- **Decentralisation of Planning:** PGP/GPGP reduces the reliance on a central planning authority, distributing the planning process among individual agents. This decentralisation can lead to more scalable and robust solutions.

Disadvantages:

- **Complexity of Data Management:** The management of multiple, interrelated plans can become complex, particularly when dealing with a large number of agents. This complexity requires sophisticated data structures and algorithms.

- **High Cognitive Load on Agents:** Implementing PGP/GPGP demands a significant level of reasoning and decision-making capacity from agents. This requirement can be challenging, especially in systems where agents have limited computational resources.

- **Communication Overhead:** The frequent exchange of plans among agents can lead to increased communication overhead, potentially impacting system efficiency and responsiveness.

2.2 Coalition Formation

Coalition formation is a strategic process in multi-agent systems where groups of agents collaborate to achieve specific objectives. These coalitions are essential when tasks are too complex or resource-intensive for a single agent, or when collaborative effort can lead to increased efficiency or better outcomes.

2.2.1 Types of Coalitions

Coalitions in multi-agent systems can be broadly categorised into two types: external and internal.

External Coalitions: In external coalitions, formation is driven by an external agent or authority. This agent dictates the composition of the coalition based on the skills and cost-effectiveness of available agents. The process is centralised, making it relatively straightforward and cost-effective to form coalitions. However, this approach may lack flexibility, as it does not account for dynamic changes in the environment or in agent capabilities.

Internal Coalitions In contrast, internal coalitions are formed through the interactions of agents themselves. Agents communicate and negotiate to identify synergies and form effective alliances. This decentralised approach fosters adaptability and autonomy but at the cost of higher complexity in coalition formation. The process is more resource-intensive, as agents need to continuously communicate and evaluate potential partnerships.

2.2.2 Advantages and Disadvantages

Advantages The primary advantage of coalition formation is the synergistic potential it offers. Agents in a coalition can pool resources, share expertise, and collaborate to achieve objectives that would be unattainable individually. This collective effort can lead to improved efficiency, better resource utilisation, and enhanced problem-solving capabilities.

Disadvantages However, coalition formation comes with inherent challenges. The most significant is the overhead involved in establishing and maintaining coalitions. Agents must invest time and resources in communication, negotiation, and coordination. The computational complexity of determining the most effective coalition composition, especially in dynamic environments, can be substantial. Additionally, there may be conflicts of interest among agents, requiring complex negotiation mechanisms to resolve.

2.2.3 Computational Considerations

From a computational perspective, forming coalitions involves solving optimization problems under constraints. Agents must consider various factors like task requirements, individual capabilities, communication costs, and potential rewards. The process often involves complex algorithms to evaluate the numerous possible coalition structures and their potential effectiveness. Real-time communication between agents for negotiation and coordination further adds to the computational load.

2.2.4 Dynamic Coalition Adjustment

In dynamic environments, the ability to adapt coalition structures in response to changing conditions is crucial. Agents need mechanisms to reassess their alliances and reconfigure coalitions as tasks evolve or new information becomes available. This adaptive approach ensures that coalitions remain effective and relevant, even in rapidly changing scenarios.

2.3 Contract Net

Contract Net protocol is widely used for the allocation of tasks in autonomous agents, involving five stages: Recognition, Announcement, Bidding, Awarding, and Expediting.

1. The protocol initialises when an agent recognises it has a task it needs assistance with, maybe because it lacks capability of achieving a solution or because or due to considerations like solution quality or deadlines.

2. This agent sends call-for-proposals to other agents or contractors, specifying task description, constraints and meta-task information.

3. Contractors have two options: they can send a proposal if they're interested or reject the offer if they're not. When submitting a proposal, they include all the necessary information for the agent to make a decision.

4. The agent picks the proposal that fits its needs the most and then sends an acceptance message to the chosen contractor. Simultaneously, rejection messages are sent to the other contractors to let them know about the decision.

5. After completing the contract, the contractor notifies the agent with an informed message. If there are specific results to share, they are included in this message. Alternatively, if the contractor is unable to fulfil its commitment, it informs the agent through a cancel message.

This protocol offers advantages when it comes to handling diverse viewpoints and distributing control, allowing fine-grained resource allocation decisions. However, it comes with several drawbacks, including the risk of network overload due to message exchanges, challenges in task assignment when numerous tasks are present, limitations in the agent's ability to precisely express preferences, and a lack of built-in sanctions for agent task failures communicated through cancel messages.

Some of these reasons are why we don't believe using this specific protocol would be the best idea. Instead, we suggest a simpler approach in which the explorer sends out requests to the fuzzy agents with certain traits, picking the best one based on a method of combining everyone's input. We will later develop this idea.

2.4 Voting

In a horizontal structure devoid of central coordinators, the voting method undergoes adaptation to encourage collaborative decision-making among agents. This protocol entails agents collectively determining the optimal action from a set of options, with the most voted action subsequently selected.

Within this framework, voting primarily serves to establish the order in which treasures are to be collected. Agents, informed by explorers about available treasures, participate in a voting process among collector agents to ascertain the initial treasure for collection.

The voting process in this context lacks a central coordinator, and instead, collector agents collectively initiate and engage in the voting process. Each agent independently decides and votes based on considerations such as proximity to treasures and current availability, with a preference for dense and unoccupied areas.

Prior to voting, all agents are informed about the ongoing process and the available options. Each agent then independently decides and communicates its choice to the group of agents, fostering a decentralised approach to decision-making.

A Plurality protocol is employed in this voting process, allowing each agent to assign one vote to its preferred alternative. The alternative garnering the highest number of votes is selected as the decision. It is assumed that agents truthfully vote based on their actual positions within the system.

In instances where a tie occurs between two or more options, agents can employ strategies such as random selection or seek a second opinion from other agents.

The final decision and execution of the chosen treasure collection involve agents collectively consolidating the results, determining the final decision, and proceeding with subsequent actions. The selected treasure is then assigned to a collector agent for collection.

This adapted voting method adheres to a fully decentralised approach, allowing agents to autonomously participate in the voting process without reliance on central authorities. It facilitates effective decision-making in the dynamic context of treasure collection within a multi-agent system with a horizontal structure.

2.5 Auction

Definition (McAfee and McMillan, JEL 1987): A market institution with an explicit set of rules determining resource allocation and prices on the basis of bids from the market participants.

Agents that receive the announcement decide for themselves whether they wish to bid for the task. Factors:

- agent must decide whether it is capable of expediting task;
- agent must determine quality constraints and price information (if relevant).

In a horizontal hierarchy where there is no central coordinator, the process of coordinating actions among agents can be distributed across the network of agents themselves. Each agent has a level of autonomy and can communicate directly with other relevant agents without relying on a central coordinator. Here's how the scenario might be adapted in a horizontal hierarchy:

When an explorer discovers a treasure and needs a collector to retrieve it, the explorer broadcasts a message to all nearby agents, including potential collectors. This message contains information about the discovered treasure, its location, and the type of treasure. Agents in the vicinity, including potential collectors, receive this message.

If the explorer can open the chest independently, it proceeds with its journey. However, if the assistance of other agents is required, the explorer waits at the treasure's location for a collector to come into the appropriate range. The collector, upon receiving the message, assesses its ability to assist and, if willing, collaborates with the explorer to open the treasure.

To coordinate the collection process, the explorer communicates its need for a collector directly to potential collectors in the area. Collectors can then submit their bids or express their willingness to assist. The explorer evaluates these bids and collaborates with the selected collector.

Similarly, the tanker's role in transporting the gathered treasure can be decentralised. The explorer communicates its need for a tanker to transport the treasure directly to nearby tankers. Tankers submit their bids or express their willingness to transport the treasure, and the explorer selects the appropriate tanker.

In a horizontal hierarchy, communication and coordination occur directly between agents, fostering a more decentralised and collaborative approach without the need for a central coordinator.

In our horizontal structure, the rules governing participation in an auction can be adapted to the decentralised nature of interactions among agents. Here's how the rules might be modified:

1. **Admission Rule:** In a horizontal structure, the admission rule can be adapted to allow agents of the same type as the initiator (who calls for the auction) to participate. Agents within proximity or with relevant capabilities can autonomously decide to participate in an auction called by another agent.

2. **Interaction Rule:** Agents, instead of sending their proposals directly to a central coordinator, can communicate directly with the agent who initiated the auction. This fosters a peer-to-peer interaction model, where agents negotiate and submit their proposals directly to each other.

3. **Validation Rule:** The validation rule can still be maintained, but agents can independently assess their own availability and lack of major occupation at the time of bidding. They don't need a central coordinator to validate their eligibility; rather, agents autonomously ensure they meet the criteria before participating.

4. **Outcome Decision:** In a horizontal structure, the decision-making process for the outcome of the auction is distributed. The initiator of the auction evaluates the proposals received directly from participating agents and makes a decision based on the best bid. There is no central coordinator; instead, the decision-making authority is distributed across the network.

5. **System Flexibility:** Including auctions in our system still provides flexibility, but in a more decentralised manner. Each auction is initiated by an agent and involves direct interactions between interested agents. This approach reduces the time needed to make decisions in specific scenarios, as agents can autonomously engage in auctions without waiting for a central coordinator.

6. **Auction Scope:** In our horizontal structure, each auction involves only one instance at a time, aligning with the decentralised nature of our system. Agents autonomously engage in one-on-one auctions, enhancing the simplicity of assigning activities to specific agents based on their attributes and capabilities.

2.6 Organisational Structures

Organisational structures are a way to coordinate and communicate among agents in a multi-agent system. They can be thought of as a set of rules and norms that govern the behaviour of the agents. These rules and norms may be explicit or implicit, and they may be enforced by the agents themselves or by an external entity.

Organisational structures can provide a number of benefits for multi-agent systems. They can help to ensure that the agents are working towards a common goal, that they are communicating effectively, and that they are resolving conflicts in a fair and efficient way.

There are many different types of organisational structures, but they can be broadly categorised into two groups: centralised and decentralised.

- **Centralised organisational structures**: have a single agent that is in charge of making all of the decisions. This can be effective for simple tasks, but it can be difficult to scale to large systems.

- **Decentralised organisational structures**: have multiple agents that make decisions independently. This can be more difficult to manage, but it can be more flexible.

3 Agent Coordination

For this system we will have to consider which coordination and communication techniques to use throughout its implementation. For this, we will go through each situation and task assigned to each agent and figure out the best communication technique.

In the previous documentation we expressed the agents we would be using to organise our system. In this case we will only use the base agents explorer, tanker and collector in order to complete the task at hand. We won't be adding any extra agents to compute any more logic. As we had previously defined, we will have an organisational hierarchy with a decentralised organisation structure. Each of the explorer agents will have the power of making decisions about where each of the coordination agents should be going. The other agents will have to ask the explorer agents for a task when they require one.

The structure for our communication agents will be based on Belief, Intent and Desires (BDI). It is based on the idea that agents are capable of reasoning about their beliefs, desires, and intentions, and that these mental states can be used to guide their behaviour.

- **Beliefs:** represent an agent's knowledge about the world. They can be either true or false, and they can be acquired from a variety of sources, such as perception, communication, and reasoning.

- **Desires:** represent an agent's goals or objectives. They can be either simple or complex, and they can be either primary or derived. A primary desire is a goal that is desired for its own sake, while a derived desire is a goal that is desired in order to achieve another goal.

- **Intentions:** represent an agent's commitments to achieving its desires. They are plans or actions that the agent is committed to carrying out. An intention is said to be active when the agent is currently working on carrying it out, and it is said to be inactive when it is no longer being pursued.

To do this, we will be using the *bdi4jade* library. This library provides the necessary functions to create a general agent, this agent will identify it's type and add the desired behaviours into it. With this structure, we can also centralise the communication types and the information required for each incoming and outgoing communication in one agent. This will make it easier to handle communications between agents and the normalisation of these between each agent type.

The initial task will be discovering the agents within the map, for this purpose we will assign an initial one shot behaviour that sends a message to all the agents in broadcast in order to announce the agent's AID and information (agent type and capacities) to the other agents in range. For this purpose, we will require a larger range than the rest of the messages involved in this study. In order to minimise this we can implement two initial behaviours: The fist one sends the agents AID and information. Once the agent has gone through a determined number of milliseconds without receiving an AID (this will depend on tests performed once the agents are up and running) the second behaviour is triggered. Since we know the number of agents in our system, we can implement this as a ground truth and expect our AID list to have 9 agents (including ourselves). If this number isn't reached we can ask those agents in range for their agent list in order to detect all agents in our platform.

The next step is assigning tasks to all the agents. The explorer, collector and tanker agents will all start with an initial explorer behaviour. These will be tasked to walk around the map and perform corresponding actions.

3.1 Explorer

The explorer will quickly move around the map trying to find the treasures around the map and get a comprehensive idea of all the nodes within the map. In order to discover the map efficiently we will have an initial exploration period to make sure that the subsequent treasures it finds can be properly fulfilled by tankers and collectors. In this initial behaviour, it will quickly discover the map and try to open the treasures it goes finding. If the explorer finds a treasure and is able to open it, it will be assigned the task of auctioning off this treasure. However, auctions will not happen until the explorer behaviour has finished.

During the explorer behaviour, the explorer will also periodically share the map instance with the rest of the explorers in range. The period in which the explorers will share their maps will be investigated once an initial system is up and running. This will make it simpler to full fill the map instance with all the nodes. The explorer will therefore take the map and get the most updated version of the map by checking the timestamp assigned to each node. It will also join nodes that it had not previously discovered.

The condition for the explorer behaviour to finish will be investigated once an initial system is up and running. When the explorer behaviour has finished, the explorers are assigned with auctioning off the treasures it has been able to open. There will be two auctions, one for the collector and one for the tanker. Initially, the explorer will be tasked with finding a collector for a specific treasure (given the treasure type). The explorer will start an auction and receive the position of the collectors that are within range. It will then assign the task of collecting the treasure to the collector that is closest to the given treasure. In the case there are no available collectors in range, the explorer will move and re-start the auction elsewhere. Once the explorer has been able to locate a collector for a given treasure, it will auction off the treasure to the tankers, again based on minimum distance. In this case, we could have considered many of the protocols explained in the previous sections, however, they would be difficult to adapt to our system due to the constraint that only the explorer has the full map and visibility of the area.

Once an auction is assigned to either a collector or a tanker the explorer will return the path towards the position the agent has to take. In the case of the tanker, it will also share the AID of the collector that will be collecting the treasure in that position. The fact that the explorer has the capacity to walk about and re-start the auction means that the range of interaction can be less in these stages as it will try again in the future.

Once the explorer has assigned each of the treasure it has opened to a *collector, tanker* pair, it has then fulfilled the task.

3.2 Collector

The collector will be initially tasked with an explorer behaviour. In this initial stage, when the collector finds a treasure it can open by itself (or is able to take fully), it can take this treasure without the need of waiting for an explorer. In this case, it will broadcast a message to explorers telling them it has taken this treasure. If the treasure cannot be taken or opened, it will go about with the explorer behaviour.

The collector will wait to receive an auction from the explorers. Once these auctions are received, they will move to the assigned slot and take the treasure that is stored there and offload to the assigned tanker. The collector will have to communicate with the tanker once this one arrives to know who and where to offload the treasure into. Once the treasure is offloaded it will be freed up again to take more treasure and repeat the action. It will then announce to the tanker that all the treasure has been taken.

Once the collector receives no more auctions, it has fulfilled it's task.

3.3 Tanker

The tanker will be initially tasked with an explorer behaviour. This means it will move around the map randomly. It will also be subscribed to auction messages from explorers. Once an auction is won by a tanker, it will be tasked with arriving to the location given a path. Once the tanker arrives to a specific location, it will announce itself to the collector assigned with taken the treasure and wait for the collector to announce that all the treasure has been taken.

Once the tanker receives no more auctions, it has fulfilled it's task.

References

[1] Glavic, Mevludin (2006). *Agents and Multi-Agent Systems: A Short Introduction for Power Engineers*. Retrieved 17:36, October 16, 2020, from https://people.montefiore.uliege.be/glavic/MAS-Intro_Tech_report.pdf.

[2] Rocha, Jorge (2017, 28 June). *Introductory Chapter: Multi-Agent Systems*. Retrieved 21:05, October 16, 2020, from https://www.intechopen.com/books/multi-agent-systems/introductory-chapter-multi-agent-systems.

[3] Boissier, Olivier. *An Introduction to Multi-Agent Systems*. Retrieved 20:05, October 16, 2020, from https://www.intechopen.com/books/multi-agent-systems/introductory-chapter-multi-agent-systems.