

Distributed Artificial Intelligence  
and  
Intelligent Agents

Marco Peressutti

April 3, 2022



# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Definition of an Agent . . . . .	3
1.1.1	Formal definitions . . . . .	3
1.1.2	Logicale behind autonomous systems . . . . .	4
1.1.3	Agent definition and properties . . . . .	4
1.2	Individual and Group Perspective . . . . .	6
1.3	Distributed AI and MAS . . . . .	6
1.3.1	Distributed Problem Solving (DPS) . . . . .	6
1.3.2	Multi-Agent Systems (MAS) . . . . .	7
1.4	Emergence, Swarm Intelligence and other terms . . . . .	9
1.4.1	Emergence . . . . .	9
1.4.2	Swarm Intelligence . . . . .	9
1.4.3	Self-Organisation . . . . .	9
1.4.4	Self-Adaptation . . . . .	9
1.4.5	Characteristics of Emergence, Swarm Intelligence, Self-Adaptation and Self-Organization . . . . .	9
1.4.6	Application . . . . .	9
<b>2</b>	<b>Agent Negotiation</b>	<b>10</b>
2.1	Multiagent Interactions . . . . .	12
2.1.1	Utilities and Preferences . . . . .	13
2.1.2	Setting the scene . . . . .	14
2.1.3	Solution Concepts and Solution Properties . . . . .	15
2.1.4	Competitive and Zero-Sum Interactions . . . . .	17
2.1.5	The Prisoner's Dilemma . . . . .	17
2.1.6	Other Symmetric 2×2 Interactions . . . . .	18
2.2	Voting . . . . .	20
2.2.1	Social Welfare Functions and Social Choice Functions . . . . .	20
2.2.2	Voting Protocols . . . . .	20
2.2.3	Desirable Properties for Voting Procedures . . . . .	20
2.2.4	Insincere Voters . . . . .	20
2.3	Auctions . . . . .	20
2.3.1	Auction parameters for classification . . . . .	20
2.3.2	English auctions . . . . .	20
2.3.3	Japanese auctions . . . . .	20
2.3.4	Dutch auctions . . . . .	20
2.3.5	First-price sealed-bid auctions . . . . .	20
2.3.6	Vickrey auctions . . . . .	20
2.3.7	Interralated auctions . . . . .	20
2.3.8	Lies and Collusion . . . . .	20
2.4	Contract Net Protocol (CNP) . . . . .	20
2.5	Negotiation parameters . . . . .	20
2.5.1	Monotonic Concession Protocol . . . . .	20
	<b>Bibliography</b>	<b>21</b>

# Chapter 1

## Introduction

---

<b>1.1 Definition of an Agent</b>	<b>3</b>
1.1.1 Formal definitions	3
1.1.2 Logicale behind autonomous systems	4
1.1.3 Agent definition and properties	4
<b>1.2 Individual and Group Perspective</b>	<b>6</b>
<b>1.3 Distributed AI and MAS</b>	<b>6</b>
1.3.1 Distributed Problem Solving (DPS)	6
1.3.2 Multi-Agent Systems (MAS)	7
<b>1.4 Emergence, Swarm Intelligence and other terms</b>	<b>9</b>
1.4.1 Emergence	9
1.4.2 Swarm Intelligence	9
1.4.3 Self-Organisation	9
1.4.4 Self-Adaptation	9
1.4.5 Characteristics of Emergence, Swarm Intelligence, Self-Adaptation and Self-Organization	9
1.4.6 Application	9

---

### 1.1 Definition of an Agent

- an agent has **independent behaviour**: when there is no possibility for direct supervision
- agents **collaborate** and **communicate** with each other
- angets have **proactive behaviour**: they have the reasoning possibility to proactively propose some solutions
- **privacy ensurance**: privacy should be ensured via **personalization**. A person's interest should not be disclouse to some external entity other than the agent itself.

independent  
behaviour

collaborate

communicate

proactive  
behaviour

privacy  
ensurance

personalization

#### 1.1.1 Formal definitions

##### 1. American Heritage Dictionary:

“One that acts or has the power or authority to **act** ... or **represent** another”

Agents are not independent entities that operate by their own view but they always represent some interest of those who create them.

##### 2. Negroponte

“Digital sister in law”

Agents should have specialized expertise and knowledge of preferences of those who create them

### 3. Russel and Norvig

“An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors.”

### 4. Pattie Maes

“Autonomous Agents are computational systems that **inhabit** some complex dynamic environment, **sense** and **act** autonomously in this environment, and by doing so **realize** a set of goals or tasks for which they are designed”

Agents live in an environment.

### 5. IBM

“Intelligent agents are software entities that carry out some set of **operations on behalf of** a user or another program with some degree of **independence** or **autonomy**, and in doing so, employ some **knowledge** or **representations** of the user’s **goals** or **desires**”

### 6. Coen

“Software agents are programs that engage in dialog [and] **negotiate** and **coordinate** transfer of information”

Agents must have the possibility to communicate and consequently coordinate and negotiate with one another.

## 1.1.2 Logics behind autonomous systems

- More and more everyday tasks are computer based
- The world is in a midst of an information revolution
- increasingly more users are untrained
- **Lack of programming paradigm** for decentralized program/system construction in dynamic environment.

Lack of programming paradigm

Existing paradigms were constructed for closed world (i.e. completely specified environment). However in AI the environment is not specified completely (hence there is the need for dynamic exploration of the environment).

The solution to this lack of paradigm is to **emulate human behaviour**, therefore agents must be able to:

emulate human behaviour

- perceive the environment
- affect the environment or act in the environment
- have a **model of behaviour**
- have intentions/motivations to be fulfilled by implementing corresponding goals.
- communication

model of behaviour

In fact, goals are not enough, the system needs to generate new goals from its intentions and beliefs.

## 1.1.3 Agent definition and properties

Wooldridge, Jennings (**weak notion**):

weak notion

“Agent is a hardware or (more usually) software-based computer system that enjoys the following properties:”

- **autonomy.**

autonomy

Agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state.

Hence in order for an agent to be autonomous it must:

- Act independently

- Have control over its internal state.

Contrary to objects, agents need to encapsulate behaviour other than their state in the environment.

They, in fact, differs from objects since they are required to have a:

1. **degree of autonomy**: agents embody a stronger notion of autonomy than objects, in particular, agents decide for themselves whether or not to perform an action. degree of autonomy
  2. **degree of smartness**: capable of flexible (reactive, pro-active, social) behaviour; standard object models do not have such behaviour degree of smartness
  3. **degree of activeness**: a multi-agent system is inherently multi-threaded in that each agent is assumed to have at least one thread of active control. degree of activeness
- **pro-activeness**.  
Agents do not simply act in response to their environment, they are able to exhibit goal-directed behavior by taking the initiative. pro-activeness  
In short, agent must be able to create goals on their own initiative.
  - **reactivity**. reactivity  
Agents perceive their environment and respond in a timely fashion to changes that occur in it.  
In short, communication with the environment.
  - **social ability**. social ability  
Agents interact with other agents (and possibly humans) via some kind of agent-communication language.  
In short, communication with other agents

Wooldridge, Jennings (**strong notion**): strong notion

- **Mentalistic notions**, such as beliefs and intentions are often referred to as properties of strong agents Mentalistic notions
- **Veracity**, agents will not knowingly communicate false information Veracity
- **Benevolence**: agents do not have conflicting goals and always try to do what is asked of it Benevolence
- **Rationality**: an agent will act in order to achieve its goals and will not act in such a way as to prevent its goals being achieved Rationality
- **Mobility**: the ability of an agent to move around a network Mobility

In addition to these basic properties several other alternatives have been proposed over the years:

1. Isaac Asimov's laws of robotics:

Law One

A robot may not injure a human being or, through inaction, allow a human being to come to harm

Law Two

A robot must obey orders given to it by human beings except where such orders would conflict with the First Law

Law Three

A robot must protect its own existence, as long as such protection does not conflict with the First or Second Law

Law Zeroth

A robot may not harm humanity, or, by inaction, allow humanity to come to harm

2. A robot must establish its identity as a robot in all cases
3. A robot must know it is a robot
4. A robot must reproduce. As long as such reproduction does not contradict with Laws 1,2 and 3
5. All robots endowed with comparable human reason and conscience should act towards one another in a spirit of brotherhood

Summary:

- Agents acts on behalf of other entities
- Agents must have weak agent characteristics
- Agents may have strong agent characteristics

## 1.2 Individual and Group Perspective

There are two fundamental agents dimensions:

- **Individual agents perspective.** That deals with how to build agents that are capable of independent autonomous actions in order to successfully carry out the tasks that we delegate to them (**Micro aspects**). Individual agents perspective
- **Group agents perspective.** That deals with how to build agents that are capable of interacting (cooperating, coordinating, negotiating) with other agents in order to successfully carry out the tasks we delegate to them (**Macro aspects**). Micro aspects  
Group agents perspective  
Macro aspects

## 1.3 Distributed AI and MAS

- Distributed AI is a topic/subject rather than the creation of a brain that is distributed.
- It became part of AI when it was possible to execute on more than one CPU.
- For this reason, it is often considered as the intersection of **Distributed Computing** and **Artificial Intelligence**. Distributed Computing  
Artificial Intelligence



- Distributed AI includes two different subfield: **Distributed Problem Solving (DPS)** and **Multi-Agent Systems (MAS)**. Distributed Problem Solving (DPS)  
Multi-Agent Systems (MAS)
- DAI deals with several aspects and dimensions such as:
  - Agent granularity
  - Heterogeneity of agents
  - Communication possibilities
  - Methods of distributing control among agents

### 1.3.1 Distributed Problem Solving (DPS)

DPS considers how the task of solving a particular problem can be divided among a number of modules that cooperate in dividing and sharing knowledge about the problem and its evolving solution(s):

- In pure DPS systems, all interaction strategies are incorporate as an integral part of the system
- DPS has focused on achieving goals under varying environmental conditions, having agents with established properties

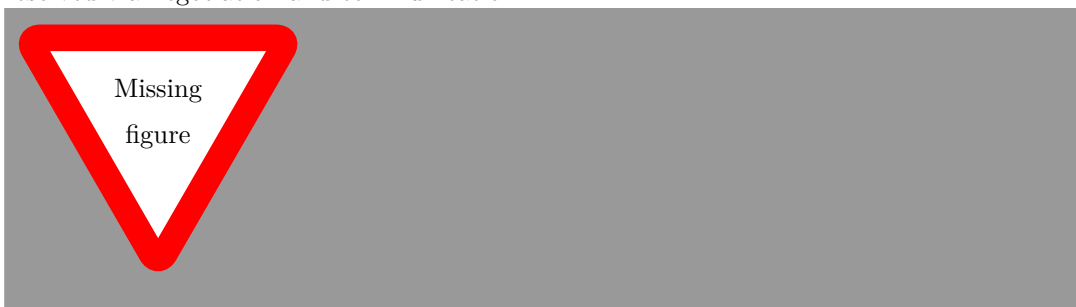
In other terms in DPS:

1. a problem is divided into modules
2. each module is allocated to a different agent

3. the agents will solve the problem by means of communicating the solution to their allocated module

### 1.3.2 Multi-Agent Systems (MAS)

- MAS are designed in a decentralized way with great part of independency and autonomy
- Agents with individual preferences will interact in particular environments such that each will consent to act in a way that leads to desired global goal
- MAS asks how, for a particular environment, can certain collective goal be realized if the properties of agents can vary uncontrollably:
  - loosely-coupled networks of problem solvers (agents) that work together to solve problems that are beyond their capabilities
  - no necessary guarantees about other agent
- MAS contain a number of agents which interact with one another through communication. The agents are able to act in an environment; where each agent will act upon or influence different parts of the environment. If two field of influence overlap, the conflict is (hopefully) resolved via negotiation and communication



- An important concept in MAS is that there is no central control (because the control is distributed to the various agents) and knowledge or information sources may also be distributed

In other terms, contrary to DPS, MAS do not deal with task to be solved but rather rules to be followed, and these rules are for example, their beliefs, desire and intentions or protocols for negotiation and communication.

The motives behind the use of MAS are:

- To solve problems that are too large for a centralized agent
- To allow interconnection and interoperation of multiple legacy systems
- To provide a solution to inherently distributed problems
- To provide solutions which draw from distributed information sources
- To provide solutions where expertise is distributed
- To offer conceptual clarity and simplicity of design

There are two subclasses of MAS:

- **Cooperative.**

Agents are designed by interdependent designers.

Agents act for increased good of the system.

Agents are concerned with increasing the performance of the system.

Cooperative

Hence if we imagine that the goodness of the system is represented by a global function.

Agents in a cooperative MAS will try to maximize such function.

- **Self-interested**

Agents designed by independent designers.

Agents have their own agenda and motivation.

Agents are concerned with the benefit and performance of the individual agent.

Self-interested



Hence, if we imagine that the goodness of each agent is represented by a local function (one for each agent). They will try to maximize their own function.

Among the benefits of using MAS over DPS we find that, MAS:

- Faster problem solving
- Decrease in communication
- Flexibility
- Increased reliability

However, its main drawback is the **lack of predictability**.

lack of  
predictability

Summary of definitions:

- Distributed Computing: focus on low level parallelization and synchronization
- Distributed AI: Intelligent control as well as data may be distributed. Focus on problem solving, communication and coordination
- DPS: Task decomposition (task sharing) and/or solution synthesis (result sharing): information management
- MAS: Behavior coordination and management

## 1.4 Emergence, Swarm Intelligence and other terms

### 1.4.1 Emergence

With the term **Emergence**, we refer to those global (macro level) behaviour, patterns and properties that arise from the interactions between local parts of the system (micro level).

Systems that exhibit emergence can be characterized as simple, robust and adaptive.

Emergence

### 1.4.2 Swarm Intelligence

The term **Swarm Intelligence** refers to any attempt to design algorithms or distributed problem-solving devices inspired by the collective behaviour of social insect colonies and other animal societies. Multi-robot systems, which implement or adapt the concept of emergent behaviour, are commonly referred to as **swarm robotic systems**.

Swarm Intelligence

swarm robotic systems

### 1.4.3 Self-Organisation

**Self-Organization** is a dynamical and adaptive process where systems acquire and maintain structure themselves, without external control.// The term self-organization is also used as the process that leads to the state of emergence.

Self-organization and emergent systems are distinct concepts, they still have one thing in common, that is: there is no explicit external control whatsoever.

The main difference between self-organization and emergence is that in the case of self-organization, individual entities can be aware of the system's intended global behaviour. In consequence, self-organization can be seen as a weak form of emergence.

The intuitive and regularly used approach to realize self-organization is applying the concept of feedback loops.

Self-Organization

### 1.4.4 Self-Adaptation

When the approach with feedback loop is applicable to single entity system it is usually referred as **self-adaptation**.

Self-adaptive software modifies its own behaviour in response to changes in its operating environment. If a decentralized system containing several entities exhibits adaptive behaviour to external changes this is as well considered self-adaptation.

self-adaptation

### 1.4.5 Characteristics of Emergence, Swarm Intelligence, Self-Adaptation and Self-Organization

having ensembles of robust (multi-)agent systems that maintain their structure and feature a high level of adaptation.

It would no longer be necessary to exactly specify the low level system behaviour in all possible situations that might occur, but rather leaving the system with a certain degree of freedom to allow for autonomous reaction and adaptation to new situations in an intelligent way.

### 1.4.6 Application

Self-organization algorithms have been applied in many multi-agent domains, like combinatorial optimization, communication networks and robotics.

The algorithms, respectively mechanism, are used for various purposes, like motion control, information sharing and decision making.

# Chapter 2

## Agent Negotiation

---

<b>2.1</b>	<b>Multiagent Interactions . . . . .</b>	<b>12</b>
2.1.1	Utilities and Preferences . . . . .	13
2.1.2	Setting the scene . . . . .	14
2.1.3	Solution Concepts and Solution Properties . . . . .	15
	Dominant strategies - Dominance . . . . .	15
	Nash Equilibria . . . . .	15
	Pareto efficiency . . . . .	16
	Maximizing social welfare . . . . .	17
2.1.4	Competitive and Zero-Sum Interactions . . . . .	17
2.1.5	The Prisoner's Dilemma . . . . .	17
2.1.6	Other Symmetric 2×2 Interactions . . . . .	18
<b>2.2</b>	<b>Voting . . . . .</b>	<b>20</b>
2.2.1	Social Welfare Functions and Social Choice Functions . . . . .	20
2.2.2	Voting Protocols . . . . .	20
	Simple majority voting . . . . .	20
	Binary protocol . . . . .	20
	Borda protocol . . . . .	20
	Majority Graph: Condorcet's Winner and Slater ranking . . . . .	20
2.2.3	Desirable Properties for Voting Procedures . . . . .	20
2.2.4	Insincere Voters . . . . .	20
<b>2.3</b>	<b>Auctions . . . . .</b>	<b>20</b>
2.3.1	Auction parameters for classification . . . . .	20
2.3.2	English auctions . . . . .	20
2.3.3	Japanese auctions . . . . .	20
2.3.4	Dutch auctions . . . . .	20
2.3.5	First-price sealed-bid auctions . . . . .	20
2.3.6	Vickrey auctions . . . . .	20
2.3.7	Interralated auctions . . . . .	20
2.3.8	Lies and Collusion . . . . .	20
<b>2.4</b>	<b>Contract Net Protocol (CNP) . . . . .</b>	<b>20</b>
<b>2.5</b>	<b>Negotiation parameters . . . . .</b>	<b>20</b>
2.5.1	Monotonic Concession Protocol . . . . .	20

---

- Davis and Smith  
“Negotiation is a process of improving agreement (reducing inconsistency and uncertainty) on common viewpoints or plans through the exchange of relevant information”.

Negotiation is a two way exchange of information (more than one agent must be involved for negotiation to happen).

Each party evaluates the information from its own perspective.

final agreement is achieved by **mutual selection**.

mutual selection

- Pruitt  
“Negotiation is a process by which a joint decision is made by two or more parties. The parties first verbalize contradictory demands and then move towards agreement by a process of concession or search for new alternatives”

Mutual conflict

**Mutual conflict** is the necessary condition to start negotiation.

Negotiation involves three main elements: **communication**, **decision making** (decision about next concession to make in order to continue negotiation) and a **procedural model** (protocol inside which we communicate and make decision).

communication

decision making

**Implicit negotiation** where one or more parties does not know that there is conflict can happen but it will not be treated in the course.

procedural model

From the definitions provided we can conclude that there are three basic negotiation categories:

Implicit negotiation

- **Negotiation language category** (Communication)

We will consider the language category more in depth in chapter ??: Agent Communication. The language category deals with the concepts of:

Negotiation language category

- **Language primitives**

Low level communications (message sending) and conversational aspects (relative to speech acts and performatives)

Language primitives

- **Object structure**

What is the object of negotiation (price, schedule, tasks, ...) and what is the context of the message/information.

Object structure

- **Internal protocol**

Specify the possibilities of initiating a negotiation cycle and responding to a message

Internal protocol

- **Semantics**

Strictly related to language primitives (pre-conditions, post-conditions, modal logic).

Semantics

- **Negotiation decision category** (Decision making)

The decision category deals with which strategy and consequently which language primitive to choose inside of a protocol.

Negotiation decision category

It deals with the concepts of:

- preferences  
what are preferable outcomes
- utility functions  
numerical/functional representation of preferences
- comparing and matching functions
- negotiation strategies

- **Negotiation process category** (Protocols/procedural model)

It deals with the concepts of:

Negotiation process category

- **Procedural negotiation model**

what is the protocol of negotiation and their properties

Procedural negotiation model

- **System behaviour and analysis**

analysis of the interaction between different parts of the system in order to recognize what are protocols and preferences (it will not be considered in the course)

System behaviour and analysis

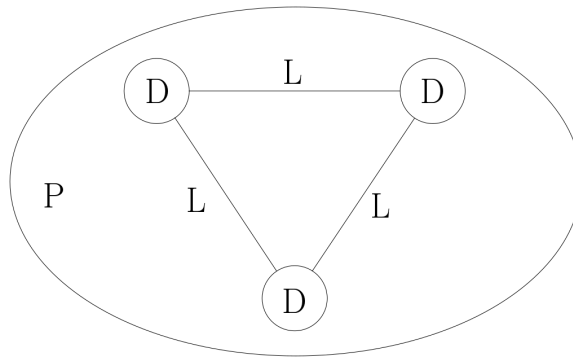


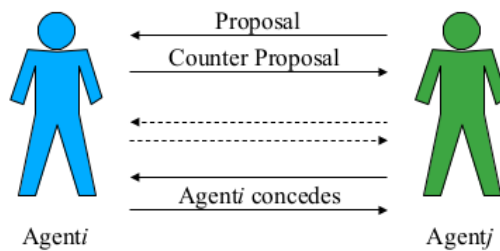
Figure 2.1: trivial form of how the categories are interrelated in the negotiation context

Furthermore, since negotiation happens when mutual conflict occurs, we can say that negotiation is a **conflict resolution** approach/method/area.

In general terms, negotiation usually proceeds in **series of rounds** with every agent making a proposal at every round. For this reason, the negotiation process can be seen as a mutual exchange of information.

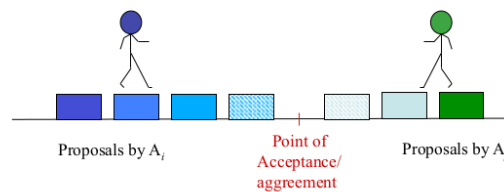
conflict resolution

series of rounds



Another way of looking at the negotiation process is as an iterative concession process: each agent starts at its most preferred outcome and makes a proposal, it will then proceed to concede to a less preferred outcome until a **Point of acceptance or agreement** is reached. (Be aware of the fact that not all negotiation end with agreement).

Point of acceptance or agreement



Hence the Negotiation process can be seen as moving towards a point of agreement.

## 2.1 Multiagent Interactions

In the typical structure of a multiagent systems, one can notice the presence of some common elements:

- the system contains a number of agents
- each agent has the ability to communicate with another agent
- each agent has the ability to act in an environment
- different agents have different **spheres of influence**, i.e. they are able to influence or have control over the environment, or a part of it.

spheres of influence

- these spheres of interest may coincide or overlap giving rise to dependencies between agents.

Thus:

“When faced with what appears to be a multiagent domain, it is critically important to understand the type of interaction that takes place between agents in order to be able to make the best decision possible about what action to perform.”

### 2.1.1 Utilities and Preferences

In order to simplify the analysis of multiagent interactions, throughout this chapter we will consider the following assumption, unless stated differently:

- Two agents act and interact in an environment. We will refer to these two agents as agent  $i$  and agent  $j$
- The two agents are **self-interest**, i.e. each agent has its own preferences and desires about how the world should be self-interest
- There exists a **set of outcomes** that the two agents have preferences over. We will refer to this set with the notation: set of outcomes

$$\Omega = \{\omega_1, \omega_2, \dots\}$$

The preferences of the two agents are formally captured by means of **utility functions**, which maps each outcome in the set  $\Omega$  to a real number describing how “good” an outcome is. utility functions  
Since the agents are self-interest each agent will have its own utility function, hence:

$$u_i : \Omega \rightarrow \mathbb{R} \quad ; \quad u_j : \Omega \rightarrow \mathbb{R}$$

The introduction of an utility function eventually leads to a **preference ordering** over the outcomes of the set for each agent. This means that, if  $\omega$  and  $\omega'$  are both possible outcomes contained in  $\Omega$  and  $u_i(\omega) \geq u_i(\omega')$ , then agent  $i$  prefers outcome  $\omega$  at least as much as outcome  $\omega'$ . preference ordering

Since we will make extensive use of the concept of preference ordering throughout the next sections, it is worth introduce a much more compact notation. We will write:

$$\begin{array}{lll} \omega \succeq_i \omega' & \text{as an abbreviation for} & u_i(\omega) \geq u_i(\omega') \\ \omega \succ_i \omega' & \text{as an abbreviation for} & u_i(\omega) > u_i(\omega') \end{array}$$

where the second expression represents the case in which outcome  $\omega$  is **strictly preferred** by agent  $i$  over  $\omega'$ . strictly preferred

In other words, the notation can be summarized as follows:

$$\boxed{\omega \succ_i \omega' \iff u_i(\omega) \geq u_i(\omega') \text{ and not } u_i(\omega) = u_i(\omega')}$$

Moreover we can notice that the ordering relation expressed by the operator  $\succeq$ , has the following properties:

1. **Reflexivity** Reflexivity  
$$\forall \omega \in \Omega \rightarrow \omega \succeq_i \omega$$
2. **Transitivity** Transitivity  
$$\text{If } \omega \succeq_i \omega' \text{ \&\& } \omega' \succeq_i \omega'' \rightarrow \omega \succeq_i \omega''$$
3. **Comparability** Comparability  
$$\forall \omega \in \Omega, \forall \omega' \in \Omega \rightarrow \omega \succeq_i \omega' \parallel \omega' \succeq_i \omega$$

The strict preference relationship still has the second and third property, however it is not reflexive

## 2.1.2 Setting the scene

So far, we have talked introduced a model to represent the agents preferences, but we still need to formalize a model of the environment in which agents act and interact. In particular, we will assume that:

- Two agents will simultaneously choose an action to perform in the environment
- as a result of the actions they select, an outcome in  $\Omega$  will result
- the **actual outcome** that will result will depend on the particular **combination of actions** performed. actual outcome
- In other terms: both agents can influence the actual outcome combination of actions
- The two agents must perform an action
- The two agents cannot see the action performed by the other agent

We will restrict our analysis to two possible actions that the agent can choose from: cooperate  $C$  and defect  $D$ . Hence, we will refer to the **set of actions** with the notation set of actions

$$Ac = \{C, D\}$$

Given all the above, the environment formally described by a **state transformer function**: state transformer function

$$\tau : \underbrace{Ac}_{\text{agent } i\text{'s action}} \times \underbrace{Ac}_{\text{agent } j\text{'s action}} \rightarrow \Omega$$

Hence, several scenarios can happen:

1. The environment maps each combination of actions to a different outcome, and thus is sensitive to the actions of both agents

$$\tau(D, D) = \omega_1 \quad \tau(D, C) = \omega_2 \quad \tau(C, D) = \omega_3 \quad \tau(C, C) = \omega_4$$

2. The environment maps each combination of actions to the same outcome and thus neither of the agents have influence in the environment

$$\tau(D, D) = \omega_1 \quad \tau(D, C) = \omega_1 \quad \tau(C, D) = \omega_1 \quad \tau(C, C) = \omega_1$$

3. The environment maps each combination of actions to an outcome that is correlated to the choice of only one agent and thus the outcome depends solely on the action performed by one agent

$$\tau(D, D) = \omega_1 \quad \tau(D, C) = \omega_2 \quad \tau(C, D) = \omega_1 \quad \tau(C, C) = \omega_2$$

Since the latter two cases are of no interest for our analysis we will focus on the scenario in which both agents exert some kind of influence on the actual outcome. We, therefore, will consider the agent preferences (in the form of utility function) as follows

$$\left. \begin{array}{llll} u_i(\omega_1) = 1 & u_i(\omega_2) = 1 & u_i(\omega_3) = 4 & u_i(\omega_4) = 4 \\ u_j(\omega_1) = 1 & u_j(\omega_2) = 4 & u_j(\omega_3) = 1 & u_j(\omega_4) = 4 \end{array} \right\}$$

Given the state transformer function, we can abuse notation and write

$$\left. \begin{array}{llll} u_i(D, D) = 1 & u_i(D, C) = 1 & u_i(C, D) = 4 & u_i(C, C) = 4 \\ u_j(D, D) = 1 & u_j(D, C) = 4 & u_j(C, D) = 1 & u_j(C, C) = 4 \end{array} \right\}$$

Hence, with respect to agent  $i$ 's preferences (first row) over the possible outcomes, we can characterize the preference ordering as follows:

$$(C, C) \succeq_i (C, D) \succeq_i (D, C) \succeq_i (D, D)$$

Similarly for agent  $j$ , the resulting preference ordering can be characterize as follows:

$$(C, C) \succeq_i (D, C) \succeq_i (C, D) \succeq_i (D, D)$$

	i defects	i cooperates
j defects	1 1	4 1
j cooperates	1 4	4 4

It is straightforward to see that if both agents **act rationally**, i.e. “they both choose to perform the action that will lead to their preferred outcomes”[1], they will both choose to cooperate. This is because, both agents prefer all the outcomes in which they cooperate, regardless of what the other agent action is. act rationally

A common way to represent the previously described interaction scenario is via **pay-off matrix** (standard game-theoretic notation): The pay-off matrix uniquely defines a **game in strategic form**. The way to interpret it is as follows: pay-off matrix  
game in strategic form

- each cell in the matrix correspond to one of the possible outcomes
- The top-right value in each cell corresponds to the payoff received by the column player
- The bottom-left value in each cell corresponds to the payoff received by the row player

### 2.1.3 Solution Concepts and Solution Properties

A question still remains unanswered: What should an agent do? What action should he chooses?

We briefly touched on the topic on the previous section, but we have provided neither the concepts that make up the solution nor the desirable properties that a solution should have.

In this section, we will tackle the problem to its core.

#### Dominant strategies - Dominance

One of the main concept that we will introduce is that of **dominance**.

“A strategy  $s_i$  is [said to be] **dominant** for player  $i$  if, no matter what strategy  $s_j$  agent  $j$  chooses,  $i$  will do at least as well playing  $s_i$  as it would doing anything else”[1]. The notion of **dominant strategy** is strictly related to the concept of **best response**, in the sense that “a strategy  $s_i$  for agent  $i$  is dominant if it is the best response to **all** of agent strategies”[1]. dominance  
dominant strategy  
best response

Thus, a dominant strategy, if it exists, simplify the decision about what action to perform: “the agent guarantees its best outcome by performing the dominant strategy”[1].

#### Nash Equilibria

The notion of **equilibrium**, or more precisely **Nash equilibrium**, is hard to formalize in the context of strategic decision making. equilibrium

We can, however, provide a basic definition by saying that two strategies  $s_1$  and  $s_2$  are in Nash equilibrium if: Nash equilibrium

1. under the assumption that agent  $i$  plays  $s_1$ , agent  $j$  can do no better than play  $s_2$ , and
2. under the assumption that agent  $j$  plays  $s_2$ , agent  $i$  can do no better than play  $s_1$

From this conditions, one can clearly notice that **two strategies are in Nash equilibrium if they are the best response to each other**.

The mutual nature of the concept makes it so that *neither agent has any incentive to deviate from a Nash equilibrium*: even if an agent chooses a different strategy, the other agent can do no better than to choose the strategy in Nash equilibrium.

Technically, this type of Nash equilibrium is known as **pure strategy Nash equilibrium**, and as much as it is appealing from its theoretical stand point, it is extremely expensive from a computational stand point. pure strategy Nash equilibrium

In fact, finding one or more Nash equilibria requires to consider each combination of strategy and check if they are in Nash equilibrium.



Thus, if there are  $n$  agents, each with  $m$  possible strategies to choose from, the number of possible combinations of actions (and hence possible outcomes) will be  $m^n$ .

Nonetheless, the presence of Nash equilibrium provides a definite answer to what action an agent should choose. However, two common issues might emerge:

1. Not every interaction scenario has a pure strategy Nash equilibrium
2. Some interaction scenarios have more than one pure strategy Nash equilibrium

With regard to the first issue, we need to modify our notion of what a strategy is. So far, we have implicitly considered a strategy as a deterministic choice of an action. This is inline with the fact that the subroutines that an agent can execute should not be subject to uncertainties or randomness (in general, we would like a subroutine to yield the same correct result on different execution).

However, “it can be useful to introduce randomness or uncertainty into our actions”[1]. The reason why that is can be best explained considering the game of rock-paper-scissors, of which the pay-off matrix is provided below: From the provided payoff matrix, one can clearly notice

	i plays rock	i plays paper	i plays scissors
j plays rock	0	1	-1
j plays paper	-1	0	1
j plays scissors	1	-1	0

that the game has no pure strategy Nash equilibrium as well as no dominant strategy. Yet, this is not completely true: a **mixed strategy** allows the agent to choose between possible choices by introducing randomness into the selection. mixed strategy

If the agent chooses one of the actions at random, with each choice having equal probability of being selected, the strategy turns out to be in **Nash equilibrium with itself**. This is because if an agent decides to choose an action at random, the other agent can do no better than adopting the same strategy and vice versa.

In general, if a player has  $k$  possible choices,  $s_1, s_2, \dots, s_k$ , then a mixed strategy over these choices takes the form:

- play  $s_1$  with probability  $p_1$
- play  $s_2$  with probability  $p_2$
- ...
- play  $s_k$  with probability  $p_k$

In other terms, a mixed strategy over  $s_1, s_2, \dots, s_k$  is a probability distribution over  $s_1, s_2, \dots, s_k$ .

The result formalized by John Forbes Nash, Jr. best summarize the discussion that we have made so far:

“Every game in which every player has a finite set of possible strategies has a Nash equilibrium in mixed strategies”

## Pareto efficiency

The notion of **Pareto efficiency** or **Pareto optimality**, contrary to the notion of Nash equilibrium and dominant strategy, is more of a (desirable) property of solutions rather than a solution concept.

Formally:

“an outcome is Pareto efficient if there is no other outcome that improves one player’s utility without making somebody else worse off”.[1]

On the other hand: “An outcome is said to be Pareto inefficient if there is another outcome that makes at least one player better off without making anybody else worse off”[1].

To put it in simpler terms we can consider an example: a brother and a sister have to divide a cake. Among the solutions of this problem, those who are Pareto efficient are:

Pareto efficiency

Pareto optimality

- The brother eats the whole cake
- The sister eats the whole cake
- Any other distribution of the cake, which leaves no cake left

Hence, a solution is Pareto efficient if it consumes/uses the total utility.

### Maximizing social welfare

Similarly to Pareto efficiency, social welfare is an important property of outcomes, but is not generally a way of directly selecting them. social welfare

The core principle of Maximizing social welfare involves the measurement of how much utility is created by an outcome in total. Maximizing social welfare

Formally, let  $sw(\omega)$  denote the sum of utilities of each agent for outcome  $\omega$

$$sw(\omega) = \sum_{i \in Ag} u_i(\omega)$$

the outcome that maximized social welfare is the one that maximizes this value.

From an individual agent's point of view, the problem with maximizing social welfare is that it does not look at the pay-offs of individual agents, only to the total welfare created. (maximizing social welfare does not care about how the utility of an outcome is divided among the players).

### 2.1.4 Competitive and Zero-Sum Interactions

A scenario in which an outcome  $\omega \in \Omega$  is preferred by agent  $i$  over an outcome  $\omega'$ , if, and only if,  $\omega'$  is preferred over  $\omega$  by agent  $j$ , is said to be strictly competitive.

Formally, a competitive scenario takes the form:

$$\omega \succ_i \omega' \iff \omega' \succ_j \omega$$

Under this condition, it is straightforward to see that the preferences of the players are diametrically opposed to one another. diametrically opposed

Zero-sum encounters, similarly, are those in which, for any particular outcome, the utilities of the two agents sum to zero. Formally, these scenario is described by the condition:

$$u_i(\omega) + u_j(\omega) = 0 \quad \forall \omega \in \Omega$$

Once again, any zero-sum scenario is strictly competitive, allowing for no possibility of cooperative behavior: "the best outcome for an agent is the worst outcome for its opponent. If an agent allows the opponent to get a positive utility, then it will get negative utility." [1]

Popular Zero-sum encounters are the games of chess and checkers as well as rock-paper-scissors.

Interesting enough, zero-sum is debatable that zero-sum games actually exists in real-world scenarios (apart from artificially forms of interaction like the games mentioned before). However, it appears that people interacting in many scenarios have a tendency to treat them as if they were zero sum.

### 2.1.5 The Prisoner's Dilemma

The Prisoner's Dilemma is as follows: "Two man are collectively charged with a crime and held in separate cells. They have no way of communicating with each other or making any kind of agreement. The two man are told that:

1. If one of them confesses to the crime and the other does not, the confessor will be freed, and the other will be jailed for three years
2. If both confess to the crime, then each will be jailed for two years

Both prisoners know that if neither confesses, then they will each be jailed for one year." [1] Under this conditions let us associate the act of confessing with defection  $D$  and not confessing with cooperating  $C$ .

	i defects	i cooperates
j defects	2 2	0 5
j cooperates	5 0	3 3

There exists 4 possible outcomes to the prisoner's dilemma: Where the number displayed are NOT the year spent in prison.

From the provided payoff matrix we can come up with the preference ordering of each agent/prisoner:

$$\begin{cases} (D, C) \succ_i (C, C) \succ_i (D, D) \succ_i (C, D) & \text{for agent } i \\ (C, D) \succ_j (C, C) \succ_j (D, D) \succ_j (D, C) & \text{for agent } j \end{cases}$$

Thus, we can analyze the best response of an agent to the choice of action of the other:

- Assuming the other player cooperates. The best response is to defect
- Assuming the other player defect. The best response is to defect

From this we can conclude that defection for  $i$  is the best response to all possible strategies of the player  $j$ : by definition, defection is thus a dominant strategy for  $i$ .

Since the same conclusion can be drawn for agent  $j$ , the scenario under consideration is **sym-** symmetric **metric**: this will result in both agent choosing to defect.

From an analysis of the problem under the concepts that we have seen previously, we can state that

- the pair  $(D, D)$  is the only Nash equilibrium of the scenario, however intuition says that this is not the best the players can do.
- the pair of actions  $(D, D)$  is also the only one that is not Pareto efficient.
- The outcome that maximizes social welfare is  $(C, C)$

“The fact that utility seems to be wasted here, and that the agents could both do better by cooperating, even though the rational thing to do is to defect, is why this is referred to as dilemma.” [1]

Moreover, the prisoner's dilemma also seems to be the game that characterized the **tragedy of the commons**: which is concerned with the use of a shared, depletable resource by a society of self-interested individuals (e.g. overfishing in the seas, exploitation of bandwidth capacity on the Internet).

Many people find the conclusion of the analysis deeply upsetting: “the result seems to imply that cooperatoin can only arise as a result of irrational behavior, and that cooperative behavior can be exploited by those who behave rationally.”[1]

Binmore argues that the discomfort we have with the analsys of the prisoner's dilemma is misplaced: “A whole generation of scholars swallowed the line that the prisoner's dilemma embodies the essence of the problem of human cooperation. They therefore set themselves the hopeless task of giving reasons why [this analysis] is mistaken... Rational players don't cooperate in the prisoner's dilemma because the conditions necessary for rational cooperation are absent”.

### 2.1.6 Other Symmetric 2×2 Interactions

The prisoner's dilemma and its variation are not the only type of multiagent interaction that exists.

In fact, if we restrict our attention to interaction in which there are:

- Two agents
- Each agent has two possible actions (C or D)
- The scenario is symmetric

Then  $4! = 24$  possible orderings of preferences, and as a consequence, 24 different games, can be constructed.

Scenario	Preferences over outcomes	Comment
1	$(C, C) \succ_i (C, D) \succ_i (D, C) \succ_i (D, D)$	cooperation dominates
2	$(C, C) \succ_i (C, D) \succ_i (D, D) \succ_i (D, C)$	cooperation dominates
3	$(C, C) \succ_i (D, C) \succ_i (C, D) \succ_i (D, D)$	
4	$(C, C) \succ_i (D, C) \succ_i (D, D) \succ_i (C, D)$	stag hunt
5	$(C, C) \succ_i (D, D) \succ_i (C, D) \succ_i (D, C)$	
6	$(C, C) \succ_i (D, D) \succ_i (D, C) \succ_i (C, D)$	
7	$(C, D) \succ_i (C, C) \succ_i (D, C) \succ_i (D, D)$	
8	$(C, D) \succ_i (C, C) \succ_i (D, D) \succ_i (D, C)$	
9	$(C, D) \succ_i (D, C) \succ_i (C, C) \succ_i (D, D)$	
10	$(C, D) \succ_i (D, C) \succ_i (D, D) \succ_i (C, C)$	
11	$(C, D) \succ_i (D, D) \succ_i (C, C) \succ_i (D, C)$	
12	$(C, D) \succ_i (D, D) \succ_i (D, C) \succ_i (C, C)$	
13	$(D, C) \succ_i (C, C) \succ_i (C, D) \succ_i (D, D)$	game of chicken
14	$(D, C) \succ_i (C, C) \succ_i (D, D) \succ_i (C, D)$	prisoner's dilemma
15	$(D, C) \succ_i (C, D) \succ_i (C, C) \succ_i (D, D)$	
16	$(D, C) \succ_i (C, D) \succ_i (D, D) \succ_i (C, C)$	
17	$(D, C) \succ_i (D, D) \succ_i (C, C) \succ_i (C, D)$	
18	$(D, C) \succ_i (D, D) \succ_i (C, D) \succ_i (C, C)$	
19	$(D, D) \succ_i (C, C) \succ_i (C, D) \succ_i (D, C)$	
20	$(D, D) \succ_i (C, C) \succ_i (D, C) \succ_i (C, D)$	
21	$(D, D) \succ_i (C, D) \succ_i (C, C) \succ_i (D, C)$	
22	$(D, D) \succ_i (C, D) \succ_i (D, C) \succ_i (C, C)$	
23	$(D, D) \succ_i (D, C) \succ_i (C, C) \succ_i (C, D)$	defection dominates
24	$(D, D) \succ_i (D, C) \succ_i (C, D) \succ_i (C, C)$	defection dominates

For the sake of the analysis we will briefly look into two more of these games: stag hunt and the game of chicken.

- The stag hunt

	i defects	i cooperates
j defects	1 1	0 2
j cooperates	2 0	3 3

- The Game of chicken

	i defects	i cooperates
j defects	0 0	1 3
j cooperates	3 1	2 2

## **2.2 Voting**

### **2.2.1 Social Welfare Functions and Social Choice Functions**

### **2.2.2 Voting Protocols**

Simple majority voting

Binary protocol

Borda protocol

Majority Graph: Condorcet's Winner and Slater ranking

### **2.2.3 Desirable Properties for Voting Procedures**

### **2.2.4 Insincere Voters**

## **2.3 Auctions**

### **2.3.1 Auction parameters for classification**

### **2.3.2 English auctions**

### **2.3.3 Japanese auctions**

### **2.3.4 Dutch auctions**

### **2.3.5 First-price sealed-bid auctions**

### **2.3.6 Vickrey auctions**

### **2.3.7 Intertrelated auctions**

### **2.3.8 Lies and Collusion**

## **2.4 Contract Net Protocol (CNP)**

## **2.5 Negotiation parameters**

### **2.5.1 Monotonic Concession Protocol**

# Bibliography

- [1] Michael Wooldridge. *An introduction to multiagent systems*. John wiley & sons, 2009.