Distributed Artificial Intelligence
and
Intelligent Agents

Marco Peressutti

April 4, 2022

Contents

Chapter 1

Introduction

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 entitities 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 leave 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 Logicale 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.
  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).

Lack of programming paradigm

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 environement or act in the environment
- have a **model of behaviour**
- have intentions/motivations to be fulfilled by implementing corresponding goals.
  In fact, goals are not enough, the sytem needs to generate new goals from its intentions and beliefs.
- communication

model of behaviour

1.1.3 Agent definition and properties

Wooldridge, Jennings (**weak notion**):
"Agent is a hardware or (more usually) software-based computer system that enjoys the following properties:"

weak notion

- **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 indipendently

autonomy

- 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**. pro-activeness
  Agents do not simply act in response to their environment, they are able to exhibit goal-directed behavior by taking the initiative.
  In short, agent smust be able to create goals on their own intiative.
- **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 golas 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 ropbot 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:

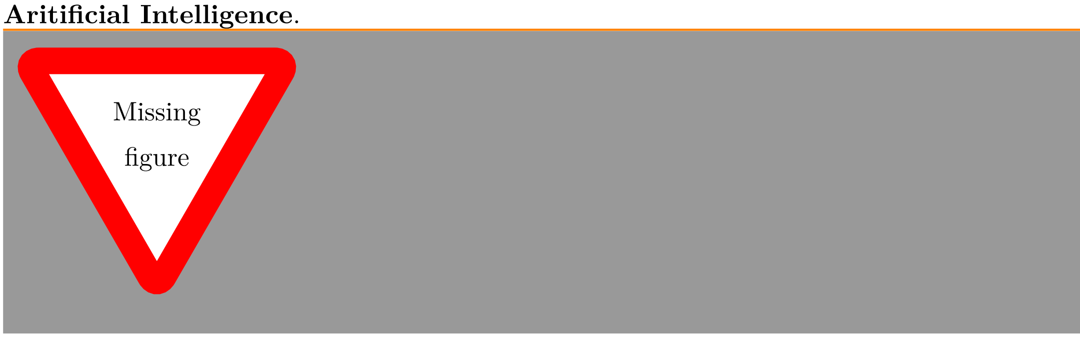
- **Individual agents perspective**. That deals with how to build agents that are capable of independent autonomous actions in order to successfully carry out the tasks that we delegate to them (**Micro aspects**).
- **Group agents perspective**. That deals with how to build agents that are capable of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

Individual agents perspective
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 **Aritificial Intelligence**.

Distributed Computing
Aritificial Intelligence



- Distributed AI includes two different subfield: **Distributed Problem Solving (DPS)** and **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

Distributed Problem Solving (DPS)
Multi-Agent Systems (MAS)

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 agetns 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

The are two subclasses of MAS:

- **Cooperative**. 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.

  Hence if we imagine that the goodness of the system is represented by a gloabal function. Agents in a cooperative MAS will try to maximize such function.
- **Self-interested** 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.

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

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. Swarm Intelligence
Multi-robot systems, which implement or adapt the concept of emergent behaviour, are commonly referred to as **swarm robotic systems**. swarm robotic systems

1.4.3 Self-Organisation

Self-Organization

Self-Organization is a dynami9cal and adaptive process where systems acquire and maintain structure themselves, without external control.// The term self-organization is also used as the process thart leads to the state of emergence.
Self-organization and emergent systems are distrinct concepts, they still have one thing in common, that is: there is no explicity external control whatsoever.
The main difference between self-organizatino 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-organizatino 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.

1.4.4 Self-Adaptation

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

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 lever 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 vaious purposes, like motion control, information sharing and decision making.

Chapter 2

Agent Negotiation

- 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
  procedural model
  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.
  Implicit negotiation

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

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

  – **Language primitives**
    Language primitives
    Low level communications (message sending) and conversational aspects (relative to speech acts and performatives)
  – **Object structure**
    Object structure
    What is the object of negotiation (price, schedule, tasks, ...) and what is the context of the message/information.
  – **Internal protocol**
    Internal protocol
    Specify the possibilities of initiating a negotiation cycle and responding to a message
  – **Semantics**
    Semantics
    Strictly related to language primitives (pre-conditions, post-conditions, modal logic).

- **Negotiation decision category** (Decision making)
  Negotiation decision category
  The decision category deals with which strategy and consequently which language primitive to choose inside of a protocol.
  It deals with the concepts of:

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

- **Negotiation process category** (Protocols/procedural model)
  Negotiation process category
  It deals with the concepts of:

  – **Procedural negotiation model**
    Procedural negotiation model
    what is the protocol of negotiation and their properties
  – **System behaviour and analysis**
    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)

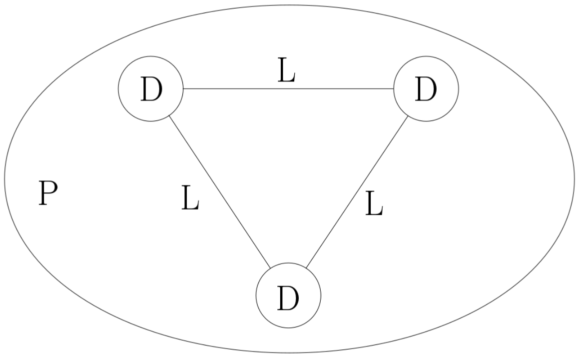


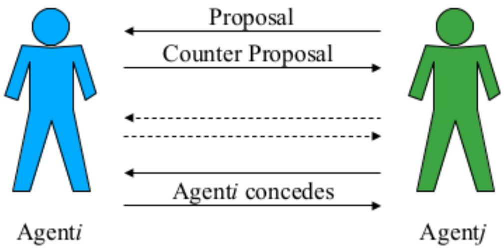
Figure 2.1: trivial form of how the categories are interralated in the negotiation context

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

conflict resolution

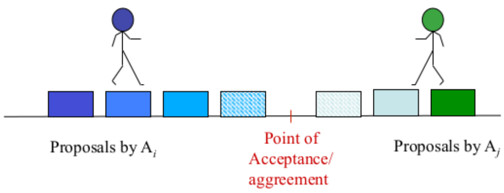
In general terms, negotiation usually proceeds in **series of rounds** with every agent making a proposal at every round. For this reason, the negotiation process can be seen as a mutual exchange of information.

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 conceed 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
- There exists a **set of outcomes** that the two agents have preferences over. We will refer to this set with the notation:

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

self-interest

set of outcomes

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

utility functions

$$u_i : \Omega \to \mathbb{R} \qquad ; \qquad u_j : \Omega \to \mathbb{R}$$

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 ω and ω' are both possible outcomes contained in Ω and $u_i(\omega) \geq u_i(\omega')$, then agent i preferes outcome ω at least as much as outcome ω'.

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:

$\omega \succeq_i \omega'$	as an abbreviation for	$u_i(\omega) \geq u_i(\omega')$
$\omega \succ_i \omega'$	as an abbreviation for	$u_i(\omega) > u_i(\omega')$

where the second expression represents the case in which outcome ω is **stricly preferred** by agent i over ω'.

stricly 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**
$$\forall \omega \in \Omega \to \omega \succeq_i \omega$$
2. **Transitivity**
$$\text{If } \omega \succeq_i \omega' \; \&\& \; \omega' \succeq_i \omega\text{``} \to \omega \succeq_i \omega\text{``}$$
3. **Comparability**
$$\forall \omega \in \Omega, \forall \omega' \in \Omega \to \omega \succeq_i \omega' \; || \; \omega' \succeq_i \omega$$

Reflexivity

Transitivity

Comparability

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 Ω will result
- the **actual outcome** that will result will depend on the particular **combination of actions** performed.
  In other terms: both agents can influence the actual outcome
- The two agents must perform an action
- The two agents cannot see the action performed by the other agent

actual outcome

combination of actions

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

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

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

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

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

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

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

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

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.

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

equilibrium

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 or 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 0	1 -1	-1 1
j plays paper	-1 1	0 0	1 -1
j plays scissors	1 -1	-1 1	0 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, $s_1, s_2, ..., s_k$, 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 $s_1, s_2, ..., s_k$ is a probability distribution over $s_1, s_2, ..., s_k$.

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. Pareto efficiency

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] Pareto optimality

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:

- 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 ω

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

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).

Other minor properties

- **Convergence/ guaranteed success**
  If it ensures that eventually agreement is certain to be reached.
  Convergence/ guaranteed success
- **Computational efficiency**
  As little computations is needed as possible. Trade off between the cost of the process and the solution quality
  Computational efficiency
- **Distribution**
  All else being equal, distributed protocols should be preferred to avoid a single point of failure and a performance bottleneck. This may conflict with minimizing the amount of communication that is required.
  Distribution
- **Stability**
  Among self interested agents, mechanism should be designed to be stable (non-manipulable), it should motivate each agent to behave in the desired manner.
  Stability
- **Individual rationality**
  Participation in a negotiation is individually rational to an agent if the agent's payoff in the negotiated solution is no less than payoff that the agent would get by not participating in the negotiation.
  A mechanism is individually rational if participation is individually rational for all agents.
  If the negotiated solution is not individually rational for some agent then self-interested agent would not participate in that negotiation.
  Individual rationality

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 ω', if ,and only if, ω' is preferred over ω by agent j, is said to be **strictly competitive**.

strictly competitive

Formally, a competitive scenario takes the form:

$$\omega \succ_i \omega' \qquad \Longleftrightarrow \qquad \omega' \succ_j \omega$$

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:

Zero-sum encounters

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

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:

Prisoner's Dilemma

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 coopering C.

There exists 4 possible outcomes to the prisoner's dilemma: Where the number displayed are

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

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 **symmetric**: this will result in both agent choosing to defect.

symmetric

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

tragedy of the commons

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

So far we have lloked at the general setting of a multiagent encounter. In this section we will look at a specific class of protocols intended for making group decisions. This is the domain of social choice theory (aka voting theory).

2.2.1 Social Welfare Functions and Social Choice Functions

The general setting of a voting protocol is as follows:

- A set of agents or **voters** voters

  $$Ag = \{1, ..., n\}$$

  Ideally, this should be finite and the cardinality should be an odd number to avoid ties.
- A set of possible outcomes or **candidates** candidates

  $$\Omega = \{\omega_q, \omega_w, ...\}$$

  over which voters will make group decision. This set is assumed to be finite.
  The goal of the agents is to rank or order these candidates or simply to choose one from the set.

The problem that social choice theory tries to solve is,
"given a collection of preference orders, one for each agent, how do we combine these to derive a group decision?".

The answert isby using a **social welfare function** which maps the voter preferences to a **social preference order** social welfare function social preference order

$$f : \quad \Pi(\Omega) \times ... \times \Pi(\Omega) \to \Pi(\Omega)$$

The social preference ordering will be denoted with the notation $\succ^*$.

If the agents are required to choose just one candidate over the set Ω, we will use a **social choice function** or **voting procedure**: social choice function voting procedure

$$f : \quad \Pi(\Omega) \times ... \times \Pi(\Omega) \to \Omega$$

2.2.2 Voting Protocols

Simple majority voting

- Every voter submits their preference order
- The winner is the outcome that appears first in the preference orders the largest number of times
- This is generally applied to single choice, but it can be generalized to a social preference ordering.

Simple majority voting

Simple majority voting works relatively well with only two candidates, since it is straightforward to implement and understand by the voters.

However, when more than 2 voters are involved problems start to arise.

Let us consider that three voters submitted the following preferences:

$$42\%\omega_L \succ \omega_D \succ \omega_C$$
$$14\%\omega_D \succ \omega_L \succ \omega_C$$
$$44\%\omega_C \succ \omega_D \succ \omega_L$$

Then the winner is selected based on the top preferences (i.e. C), even though the majority of the voters considered C as the least preferrable outcome.

This means that in principle the majority of the voters will be unhappy with the result of the voting protocol.

Furthermore, this simple majority protocol is sensible to insincere strategical voting (e.g. the 14% that voted D instead could have voted for L just to make C lose the voting).

Moreover, the simple majority voting is sensible to **Condorcet's paradox**: Condorcet's paradox

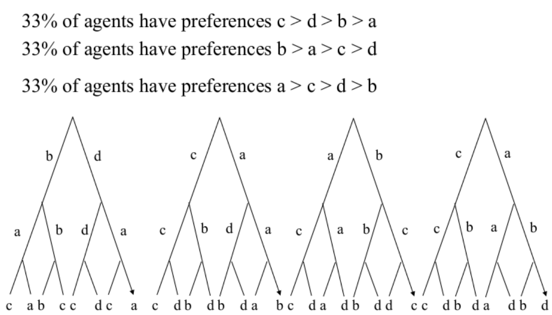
$$\omega_1 \succ_i \omega_2 \succ_i \omega_3 \omega_3 \succ_j \omega_1 \succ_j \omega_2 \omega_2 \succ_k \omega_3 \succ_k \omega_1$$

In this case the winner is dependent on the agenda: which order we start to compute the outcome, but nonetheless 2/3 of the voters will prefer another candidate.

Binary protocol

Among the alternative to simple majority voting there is the **Binary protocol** (aka sequential majority elections). Binary protocol

- Voters submit their preference ordering
- The element of the set of outcomes are compare pairwise
- the winner of the pairwise “election” is allowed to continue to the next pairwise election
- The order of pairwise elections (i.e. which candidates to compare first), is called **agenda** agenda



From the picture provided we can clearly notice that the winner of the overall election depends on the agenda, which imply that the protocol can be manipulated if the preference orderings in known a priori.

Borda protocol

One issue with simple majority voting and the binary protocol is that they completely ignore the information that is enconded in each submitted preference ordering (they only loop at the top voted candidates).
The **Borda protocol** addresses such issue:

Borda protocol

- Each voter submit its preference ordering over the candidates
- a number equal to the cardinality of the set of candidates (i.e. $|\Omega|$) is assigned to the highest candidate in some agent's preference list
- a value of $|\Omega| - 1$ is assigned to the second and so on
- a the end we sum up the numeric value associated to each preference orderic and obtain the social choice result

This protocol is fast when compared to the binary protocol in the case of a large amount of candidates.
However it is not **indipendent of irrelevant alternatives**, i.e. removing one candidate will change the end result.

indipendent of irrelevant alternatives

Majority Graph: Condorcet's Winner and Slater ranking

An useful tool to analyse voting procedures or protocols is the **Majority Graph**. It can be considered as a succint representation of the voter preferences:

Majority Graph

- each node in the majority graph is a candidate
- each edge in the majority graph represents a preference (the edge will start from the node most preferred and will terminate in the node less preferred)
- a **possible winner** is a candidate that is the overall winner for at least one agenda. (occurs when there are loops in the majority graph)
- a **Condorcet's winner** is a candidate that is the overall winner for all possible agenda (occurs when there are no loops in the majority graph). In simple terms, is that candidate that will beat all others in a pairwise election.

possible winner

Condorcet's winner

While the overall winner determination is straightforward in the case of a majority graph with no loops (cfr. Condorcet's Winner), it is otherwise in the case of a majority graph with loops.
In that specific scenario, the **slater rule** or **slater ranking** is considered: it, in fact, tries to find a consistent ranking that does not contain any cycle that is as close a possible to the majority graph.
In other terms, it tries to "minimize the number of disagreements between the majority graph and the social choice".
In practice, the Slater ranking considers the Condorcet's winners obtained by inverting the least amount of edges in the majority graph.

slater rule

slater ranking

2.2.3 Desirable Properties for Voting Procedures

1. **Pareto condition**
   There is no other outcome that makes one agent better off without making another agent worse off.

   Borda and simple majority voting satisfy this condition.
   Binary protocol does not (the result depends on the agenda).

   Pareto condition

2. **Condorcet winner condition**
   The Condorcet winner should be selected by the social choice function.

   Only Binary protol satisfies this condition.

   Condorcet winner condition

3. **Indipendence of irrelevant alternatives**
   Given a set of candidates of which ω_i and ω_j are part of.
   Let us assume that $\omega_i \succ^* \omega_j$.
   A change in preferences in any other candidate should not change the relative ranking of ω_i and ω_j.

   Indipendence of irrelevant alternatives

None of the protocol we have seen satisfies this property.

4. **Dictatorship**
   A social welfare function is said to be a dictatorship if for some voter i we have

$$f(\bar{\omega_1}, \bar{\omega_2}, ...) = \bar{\omega_i}$$

   All protocols we have seen so far are non-dictatorship.

Dictatorship

Theorem 1 (Arrow's theorem)**.**
If $|\Omega| > 2$ then the only voting procedure that satisfies Pareto condition, Condercet Winner condition and IIA is dictatorship.

Arrow's theorem

2.2.4 Insincere Voters

- In reality it is seldom that all agent's preferences are known: usually agents have to reveal their preferences.
- Assuming knowledege of the preferences is equivalent to assuming that the agents reveal their preferences truthfully
- If an agent can benefit from insincerely declaring his preferences, it will do so.
- The goal is to generate protocols such that when agents use them according to some stability solution concept (dominant strategy equilibrium) the desirable social outcomes follow
- The strategies are not externally imposed on the agents, but instead each agent uses the strategy that is best for itself

However:

Let each agent i from A have some ordering ϑ_i from Θ which totally characterized his preferences. A social choice function $f: \quad \Theta \to \Omega$ chooses a social outcome given the agent's orderings.

Theorem 2 (Gibbard-Satterthwaite impossibility theorem)**.**
Let each agent's ordering ϑ_i consists of a preference order $\succ_i$ on Ω.
Let there be no restictions on $\succ_i$ (i.e. each agent may rank the outcomes Ω in any order).
Let $|\Omega| > 2$.
Now, if the social function $f(\cdot)$ is truthfully implementable in a dominant strategy equilibrium (or is not manipulable), then $f(\cdot)$ is dictatorial, i.e. there is some agent who gets one of its most preferred outcome chosen no matter what types the other reveal.

Gibbard-Satterthwaite impossibility theorem

Broadly speaking, the only procedure or mechanism that is immune to strategic manipulation (in the sense of untruthful report of an agent preference) is dictatorship in the case of more than 2 candidates.

There are ways to circumvent the impossibility theorem, by relaxing the assumptions made by the theorem itself.
The individual preferences, in fact, may happen to belong to some restrincted domain, thus invalidating the conditions of the impossibility theorem, and it is known that there are island in the space of agent's preferences for which non-manipulable non-dictatorial protocols can be constructed.

The solution in practive is to make agents precisely internalize the externality by imposing a tax on those agents whose vote changes the outcome. The size of tax is exactly how much its vote lowers the other's utility.
Such solution is known as the **Clark tax algorithm**:

Clark tax algorithm

- Every agent i from A reveals his valuation $v_i^*(g)$ for every possible g (which may be non-truthful)
- The social choice is

$$g^* = \underset{g}{\text{argmax}} \sum v_i^*(g)$$

- Every agent is levied a tax:

$$tax_i = \sum_{j \neq i} v_j^*(g) \left(\underset{g}{\text{argmax}} \sum_{k \neq i} v_k^*(g) \right) - \sum_{j \neq i} v_j^*(g)$$

It follows that

Theorem 3. *If each agent has quasilinear preferences, then, under the Clarke tax algorithm, each agent's dominant strategy is to reveal his true preferences, i.e.*

$$v_i^*(g) = v_i(g) \qquad \forall g$$

In conclusion, the Clarke tax algorithm:

- the mechanism leads to the socially most preferred g to be chosen
- because of truth telling, the agents need not waste effort in counter speculating each other preference declarations
- participation in the mechanism may only increase an agent's utility
- the mechanism does not mantain budget balance: too much tax is collected
- it is not coalision proof

Other way to circumvent the Impossibilty theorem:

- some fairness can be achieved by choosing the dictator randomly in the protocol
- to use a protcol for which computing an untruthful revelation (that is the better than the truthful one) is prohibitively costly computationally

2.3 Auctions

Auctinos are mechanisms sed to reach agreeements on one very simple issue: that of how to allocate scarce resources to agents.

It is important to understand that the resource in question is scarce and it is tipically desired by more than one agent (if one of the preconditions is not met than the allocation is trivial and straighforward).

Auctions provide a reasonable and principled way to allocate resources to agents, in particular they are effective at allocating resources efficiently, in the sense of allocating resources to those that value them the most.

2.3.1 Auction parameters for classification

In general terms:

- An auction takes place between an agent known as the **auctioneer** and a collection of agents known as the **bidders**
- The goal of the auctino is for the auctioneer to allocate a good to one of the bidders
- The auctioneer desires to maximize the price at which the good is allocated.
  Hence the agent auctioneer will attempt to achieve its desire through the design of an appropriate auction
- The bidders desire to minimize the selling price.
  Hence the bidder agents will attempt to achieve their desires by using an effective strategy

`auctioneer`

`bidders`

There are several factors that can affect both the auctino protocol and the strategy that agents use:

- The good has a **private value**, **public/common value** or a **correlated value** (i.e. an agent valuation of the good depends partly on private factors and partly on other agents' valuations of it).
- **winner determination**: who gets the good that the bidders are bidding for and how much do they pay.
  In this setting, there are **first-price auctions** (the price goes to the highest bidder) and **second-price auctions** (the price goes to the highest bidder but it will pay the amount bid by the second highest bid)
- Whether or not the bids made by the agents are known to each other.
  In this setting, we will differentiate between **open-cry** and **sealed-bid**

`private value`

`public/common value`

`correlated value`

`winner determination`

`first-price auctions`

`second-price auctions`

`open-cry`

`sealed-bid`

- The mechanism by which bidding proceeds.
  In this setting, we will differentiate between **one shot**, **ascending auctions** or **descending auctions** (in which the price starts at a **reservation price**: in the case of ascending the reservation price is proposed by the first bidder, whereas in descending the reservation price is proposed by the auctioneer).

one shot

ascending auctions

descending auctions

reservation price

2.3.2 English auctions

English auctions are: first-price, open cry, ascending auctions.

- The auctioneer sets a reservation price for the good (low price) and the good is allocated to the auctioneer for this amount
- Bids are then invited from agents, who must bid more than the current highest bid. (all agents can see the bid)
- When no agent is willing to raise the bid, then the good is allocated to the agent that has made the current highest bid at the amount of its bid.

The dominant strategy is to bid a small amount more than the current highest bid until the bid price reaches their private valuation and then to withdraw.

English auction's however are sensible to the **winner's curse**: should the winner feel "happy" that they have obtained the good for less than or equal to their private valuation or should they feel worried because no other agent valued the good so highly?

winner's curse

2.3.3 Japanese auctions

Japanese auctions are: open cry, ascending auctions

- The auctioneer sets a reservation price
- Each agent must choose whether or not to be in
- The auctioneer successively increases the price
- After each increase an agent must say if he stays or not. When agent drops out it is irrevocable
- The auction ends when exactly one agent is left
- The winner must buy the good for the current price

2.3.4 Dutch auctions

Dutch auctions are: open-cry, descending auctions:

- The auctioneer sets a reservation price
- The auctioneer then continually lowers the offer price of the good by some small value, until some agent makes a bid for the good
- The good is then allocated to the agent that made the offer

Dutch auctions are also susceptible to the winner's curse.

The dominant strategy is to shade bid a bit below the true willingness to pay.

2.3.5 First-price sealed-bid auctions

First-price sealed-bid auctions are: first-price, sealed-bid, one-shot auctions.

- A single round of proposals is considered
- Bidders submit to the auctioneer a bid for the good
- The good is awarded to the agent that made the highest bid
- The winner pays the price of the highest bid

The dominant strategy for an agent is to bid less than it true valuation. How much less will of course depend on what the other agents bid (hence there is no general solution).

2.3.6 Vickrey auctions

Vickrey auctions are: second-price, sealed-bid auctions.

- There is a single bidding round
- Each bidder submits a single bid.
- The good is awarded to the agent that made the highest bid
- The winning bidder pays the price of the second-highest bid

The dominant strategy is truth telling: a bidder's dominant strategy in a private value Vickrey auction is to bid their true valuation.

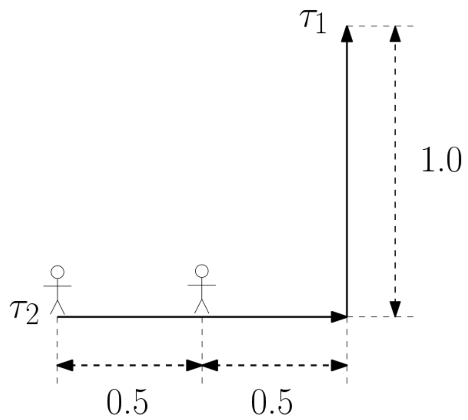
As such Vickrey auction are not prone to strategic manipulation, however it makes possible for **antisocial behaviour** to occur: one agent may bid close to the highest bid just to make the winner pay close to the amount that they have bid. antisocial behaviour

2.3.7 Interralated auctions

Some auctions are interralated to each other:

- Two good are sold separately, but the bidders would like to acquire them together
- In this case the strategy to consider for more than one auction but for several interrelated auctions and then accomodate some valuations and bids according to this strategy but not for some particolar auction.

Let us consider the example of the proposed figure:



Two agents (agent 1 on the left and agent 2 on the right) are concurring for the acquisition of two tasks: τ_1 and τ_2.
It makes no sense for them to acquire just one of the two tasks. Please notice that in order to complete the full task an agent must go to the start of the arrow/task and fully traverse its edge.

Condering all of the above the valuation of the tasks for the two agents can be represented as a cost-to-go:

$$\begin{array}{llll}
\text{Agent 1:} & c_1(\tau_1) = 2.0 & c_1(\tau_2) = 1.0 & c_1(\tau_1, \tau_2) = 2.0 \\
\text{Agent 2:} & c_2(\tau_1) = 1.5 & c_2(\tau_2) = 1.5 & c_2(\tau_1, \tau_2) = 2.5
\end{array}$$

However if both agents decided to follow this strategy, task τ_1 would go to agent 2 (lower cost to go), whereas task τ_2 would go to agent 1 (lower cost to go).

In order for agent 1 to win both tasks, since it is of no use for an agent to win just one task in an interrelated auction, it can incorporate the full look ahead in its bid.

If in fact, agent 1 wins τ_1, it will bid $c_1(\tau_2) = c_1(\tau_1, \tau_2) - c_1(\tau_1) = 0$ for the second task, otherwise it will bid $c_1(\tau_2) = 1$. So it makes sense to accomodate the first bid with this knowledge. If agent 1 wins τ_1, it will win τ_2 at the price of 0 and gets a payoff of $c_1(\tau_2) - 0 = 1$ that can redistribute to the first bid.

In conclusion the dominant strategy is to accomodate the payoff into the first bid:

$$c_1(\tau_1) - payoff = 2 - 1 = 1$$

In other terms this implies that if agent 1 realizes that by winning τ_2 it will automatically win τ_1, it will bid "higher" for τ_1 or say that the effort to complete task τ_1 will be lower by winning also τ_2.

2.3.8 Lies and Collusion

It is in the auctioneer interest to have a protocol that is immune to collusion by bidders, i.e. that made it against the bidder' best interests to engage in collusion with other bidders.

Similarly, as a potential bidder in an auction, we would like a protocol that made honesty on the part of the auctioneer the dominant strategy.

None of the auction discussed above is immune to collusion: for any of them the grand coalition of all agents involved in bidding for the good can agree beforehand to collude to put forward artificially low bids for the good on offer.
When the good is obtaines, the bidders can then obtain its true value and split the profits among themselves. A solution to collusion of the bidder is to modify the protocol so that the bidders cannot identify each other which however it's not possible in open cry auctions.

With regard to the honesty of the auctioneer, the main opportunity for lying occurs in the Vickrey auctions: the auctioneer can lie to the winner about the price of the second highest bid. A solution to this is to sign bids in some way so that the winner can independently verify the value of the second highest bid ro use a trusted third party to handle bids.

In open cry auctino settings there is no possibility for lying by the auctioneer, nor in the first-price sealed-bid auctions.

Lastly, another possible opportunity for lying by the auctioneer is to place bogus bidders (aka **shills**), in an attempt to artificially inflate the current bidding price. shills

Moreover, there are main limitation of auctions is that they are only concerned with the allocation of goods and as such are not adequate for settling agreements that concerns maters of mutual interest.

2.4 Contract Net Protocol (CNP)

2.5 Negotiation parameters

The basic components that make up a negotiation setting are:

- A **Negotiation set**, which represents the space of possible proposals that agents can make Negotiation set
- A **protocol**, which defines the legal proposals that agents can make, as a funciton of prior negotiation history. protocol
- A collection of **strategies**, one for each agent, which determine what proposals the agents will make. strategies
- A **rule** that determines when a deal has been struck, and what this agreement deal is. rule

Negotiation usually proceeds in a series of rounds, with some proposal made at every round. The proposals that agents make are defined by their strategy, must be drawn from the negotiation set, and must be legal, as defined by the protocol.
If agreement is reaches, as defined by the agreement rule, then negotiation terminates with the agreement deal.

Several attributes may complicate negotiation:

- Multiple issues are involved: in case of a single attribute/price we have a symmetric scenario which is easy to analyze because it is always obvious what represents a concession.
  On the other hand, in multiple-issue negotiation scenarios, agents negotiate over not just the value of a single attribute, but the values of multiple attributes, which may be interrelated. In such scenarios, it is usually much less obvious what represents a true concession.

Moreover multiple attributes also lead to an exponential growth in the space of possible deals. This means that, in attempting to decide what proposal to make next, it will be entirely unfeasible for an agent to explicitly consider every possible deal in domains of moderate size.

- Another source of complexity in negotiation is the number of agents involved in the process, and the way in which these agents interact. There are three obvious possibilities:
  1. **One-to-one negotiation** One-to-one negotiation
  2. **Many-to-one negotiation**, as in the case of auctions or reverse auctions Many-to-one negotiation
  3. **Many-to-many negotiation** Many-to-many negotiation

2.5.1 Task Oriented Domain

The idea behind Negotiation for task allocation is that agents who have tasks to carry out may be able to benefit by reorganizing the distribution of tasks among themselves; but this raises the issue of how to reach agreement on who will do which tasks.

Formally this scenario is known under the name of **Task Oriented Domain (TOD)**. A task oriented domain is a triple: Task Oriented Domain (TOD)

$$< T, Ag, c >$$

where

- T is the finite set of all possible tasks
- $Ag = \{1, ..., n\}$ is the finite set of negotiation participant agents
- $c : 2^T \rightarrow \mathbb{R}_+$ is a function which defines the cost of executing each subset of tasks: the cost of executing any set of tasks is a positive real number

The cost functino must satisfy two constraints:

1. It must be **monotonic** monotonic

$$\text{If } T_1, T_2 \subseteq T \text{ are sets of tasks such that } T_1 \subseteq T_2, then c(T_1) \leq c(T_2)$$

2. The cost of doing nothing is zero

$$c(0) = 0$$

We will restrict our attention as follows:

- One-to-one negotiation scenario, with two agents $\{1, 2\}$
- Given an encounter $< T_1, T_2 >$, a **deal** will be an allocation of the tasks $T_1 \cup T_2$ to the agents 1 and 2. deal
- Three types of deal may happen under this conditions:
  1. **Pure deals**: agents are deterministically allocated exhaustive disjoint task set. Formally, a pure deal is a pair $< D_1, D_2 >$ where Pure deals

$$D_1 \cup D_2 = T_1 \cup T_2$$

  2. **Mixed deals**: specify a probability distribution over partitions Mixed deals
  3. **All-or-Nothing deals**: mixed deals where the alternatives only include partitions where one agent handles the tasks of all agents. All-or-Nothing deals
- The **cost** to an agent i of a deal $\delta =< D_1, D_2 >$ is defined to be $c(D_i)$ and it will be denoted as $cost_i(\delta)$ cost
- The **utility** of a deal δ to an agent i is the difference between the cost of agent i doing the tasks T_i that it was originally assigned in the encounter and the cost $cost_i(\delta)$ of the tasks it is assigned in δ: utility

$$utility_i(\delta) = c(T_i) - cost_i(\delta)$$

Thus the utility of a deal represents how much the agent has to gain from the deal (a negative utility means that the agent is worse off than it was originally)

- If the agents fail to reach an agreement they must perform the tasks that they were originally allocated (**conflict deal**, $\Theta =< T_1, T_2 >$) conflict deal

The properties that govern this agent interactions are:

- Dominance
  A deal δ_1 is said to be dominant if and only if:
  1. Deal δ_1 is at least as good for every agent as δ_2

  $$\forall i \in \{1, 2\}, utility_i(\delta_1) \geq utility_i(\delta_2)$$

  2. Deal δ_1 is better for some agent than δ_2

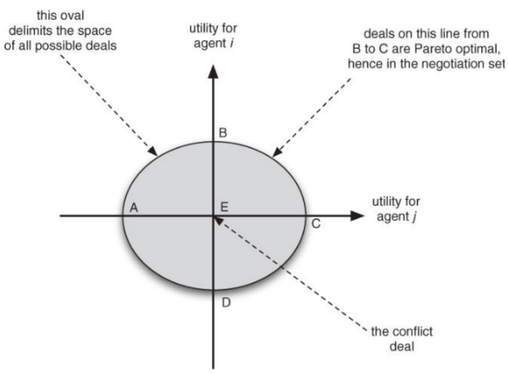
  $$\exists i \in \{utility_i(\delta_1) > utility_i(\delta_2)\}$$

  A deal is said to be **weakly dominant** if at least the first condition holds. weakly dominant
- Pareto optimal
  A deal δ is Pareto optimal if there is no deal δ' such that $\delta' \succ \delta$
- Individual rationality
  A deal δ is said to be individual rational if it weakly dominates the conflict deal

  $$\delta \succeq \Theta$$

  If a deal is not individual rational, then at least one agent can do better by simply performing the tasks it was originally allocated (it prefers the conflict deal)

Given all of the above, we are in the position to define the space of possible proposals that agents can make: the negotiation set consists of the set of deals that are individual rational and Pareto optimal.
In the following figure is proposed the set of possible deals in a 2 agent encounter



Here, the first, third and fourth quadrant includes the subset of deals that are not individual rational. (there is no point for an agent to accept a deal that produces negative utility).
Similarly, all deals strictly inside the ellipsis are not Pareto Optimal. Which brings as to the boundary of the negotiation set in the second quadrant.
Agent i (ordinate axis) will start at its best possible deal (B), whereas agent j (abscissa axis) will start at its best possible deal (C).
Both agent will then concede in one or more rounds until a point of mutual agreement on the boundary of the negotiation set is reached.

2.5.2 Worth Oriented Domain

2.5.3 Monotonic Concession Protocol (MCP)

The rules of this negotiation protocol are:

- Negotiation proceeds in a series of rounds
- On the first round both agents simultaneously propose a deal from the negotiation set
- An agreement is reached if the two agents propose deals δ_1 and δ_2, respectively, such that either:

$$utility_1(\delta_2) \geq utility_1(\delta_1) \qquad || \qquad utility_2(\delta_1) \geq utility_2(\delta_2)$$

- If both agents offers match or exceed those of the other agent, then one of the proposals is selected at random
- If only one proposal exceeds or matches the other's proposal, then this is the agreement deal
- If no agreement is reached, then negotiation proceeds to another round of simultaneous proposals, where no agent is allowed to make a proposal that is less preferred by the other agent than the deal it proposed at the previous round
- If neither agent makes a concession in some round, then negotiation terminates with the conflict deal

Using such protocol negotiation is guaranteed to end (with or without agreement) after a finite number of rounds, however the protocol does not guarantee that the agreement (or lack of it) will be reached quickly.

2.5.4 The Zeuthen strategy

1. What should an agent's first proposal be?
2. On any given round, who should concede?
3. If an agent concedes, then how much should it concede?

With regard to the first question: an agent's first proposal should be its most preferred deal.

With respect to the second question: the idea of the **Zeuthen strategy** is to measure an agent's willingness to risk conflict (i.e. an agent is more willing to risk conflict if the difference in utility between its current proposal and the conflict deal is low). Zeuthen strategy

Agent i's willingness to risk conflict at round t is:

$$risk_i^t = \frac{\text{utilty } i \text{ loses by conceding and accepting} j\text{`s offer}}{\text{utility } i \text{loses by not conceding and causing conflict}}$$

Until an agreement is reached, the value of risk is between 0 and 1 (the higher the value the higher the risk, which means that an agent has less to lose from conflict and as such it is more willing to risk conflict).
Formally:

$$risk_i^t = \begin{cases} 1 & \text{If } utility_i(\delta_i^t) = 0 \\ \dfrac{utility_i(\delta_i^t) - utility_i(\delta_j^t)}{utility_i(\delta_i^t)} & \text{otherwise} \end{cases}$$

2.5.5 Deception

Bibliography

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