Tutorial Information

Lecturer: Ulle Endriss (ulle@illc.uva.nl)

Notes: This tutorial is based on the “MARA Survey”:

This paper has been written in the context of the AgentLink Technical Forum Group on Multiagent Resource Allocation (TFG-MARA).

Website: Survey and slides are also available at the tutorial website:
http://www.illc.uva.nl/~ulle/teaching/aamas-2006/

What is MARA?

A tentative definition would be the following:

Multiagent Resource Allocation (MARA) is the process of distributing a number of items amongst a number of agents.

Now: What kind of items (resources) are being distributed? How are they being distributed? And finally, why are they being distributed?

Outline

• Concerning the specification of MARA problems:
  – Overview of different types of resources
  – Representation of the preferences of individual agents
  – Notions of social welfare to specify the quality of an allocation

• Concerning methods for solving MARA problems:
  – Discussion of allocation procedures (centralised/distributed)
  – Some complexity results concerning allocation procedures

• Issues we will not have time to cover in this tutorial:
  – Strategic considerations: mechanism design
  – Algorithmic considerations: algorithm design
  – Experimentation using simulation platforms

Examples of Application Areas

The following applications are described in detail in the MARA Survey:

• Industrial Procurement
• Earth Observation Satellites
• Manufacturing Systems
• Grid Computing

Types of Resources
Types of Resources

- A central parameter in any resource allocation problem is the nature of the resources themselves.
- There is a whole range of different types of resources, and each of them may require different techniques . . .
- Distinguish properties of the resources themselves and characteristics of the chosen allocation mechanism. Examples:
  - Resource-inherent property: Is the resource perishable?
  - Characteristic of the allocation mechanism: Can the resource be shared amongst several agents?

Divisible or not

- Resources may be treated as being either divisible or indivisible.
- Continuous/discrete: physical property of resources
  - Divisible/indivisible: chosen feature of the allocation mechanism

Static or not

Resources that do not change their properties during a negotiation process are called static resources. There are at least two types of resources that are not static:
- consumable goods such as fuel
- perishable goods such as food

In general, resources cannot be assumed to be static. However, in many cases it is reasonable to assume that they are as far as the negotiation process at hand is concerned.

Resources vs. Tasks

- Tasks may be considered resources with negative utility.
- Hence, task allocation may be regarded a MARA problem.
- However, tasks are often coupled with constraints regarding their coherent combination (timing).

Continuous vs. Discrete Resources

- Resource may be continuous (e.g. energy) or discrete (e.g. fruit).
- Discrete resources are indivisible, continuous resources may be treated either as being (infinitely) divisible or as being indivisible (e.g. only sell orange juice in units of 50 litres ~ discretization).

  - Representation of a single bundle:
    - Several continuous resources: vector over non-negative reals
    - Several discrete resources: vector over non-negative integers
    - Several distinguishable discrete resources: vector over \{0, 1\}

- Classical literature in economics mostly concentrates on a single continuous resource; recent work in AI and Computer Science focusses on discrete resources.

Sharable or not

- A sharable resource can be allocated to a number of different agents at the same time. Examples:
  - a photo taken by an earth observation satellite
  - path in a network (network routing)
- More often though, resources are assumed to be non-sharable and can only have a single owner at a time. Examples:
  - energy to power a specific device
  - fruit to be eaten by the agent obtaining it

Single-unit vs. Multi-unit

- In single-unit settings there is exactly one copy of each type of good; all items are distinguishable (e.g. several houses).
- In multi-unit settings there may be several copies of the same type of good (e.g. 10 bottles of wine).
- Note that this distinction is only a matter or representation:
  - Every multi-unit problem can be translated into a single-unit problem by introducing new names (inefficient, but possible).
  - Every single-unit problem is in fact also a (degenerate) multi-unit problem.
- Multi-unit problems allow for a more compact representation of allocations and preferences, but also require a richer language (variables ranging over integers, not just binary values).

Preference Representation
Preference Representation

The second important parameter in the specification of a MARA problem are the preferences of individual agents.

Agents may have preferences over
- the bundle of resources they receive
- the bundles of resources received by others (externalities)
What are suitable languages for representing agent preferences?

For single-unit settings with indivisible resources, for instance, the number of alternatives is exponential in the number of goods, so an explicit representation may not be feasible …

Cardinal and Ordinal Preferences

A preference structure represents an agent’s preferences over a set of alternatives \( \mathcal{X} \). There are different types of preference structures:
- A cardinal preference structure is a function \( u: \mathcal{X} \rightarrow \text{Val} \), where \( \text{Val} \) is usually a set of numerical values such as \( \mathbb{N} \) or \( \mathbb{R} \).
  The function \( u \) is often called a utility (or valuation) function.
- An ordinal preference structure is a binary relation \( \preceq \) over the set of alternatives, that is reflexive and transitive (and connected).

If the alternatives over which agents have to express preferences are bundles of indivisible resources from the set \( \mathcal{R} \), then we have \( \mathcal{X} = 2^\mathcal{R} \).

Cardinal Preferences: Explicit Representation

The explicit form of representing a utility function \( u \) consists of a table listing for every bundle \( X \subseteq \mathcal{R} \) the utility \( u(X) \). By convention, table entries with \( u(X) = 0 \) may be omitted.
- the explicit form is fully expressive: any utility function \( u: 2^\mathcal{R} \rightarrow \mathbb{R} \) may be so described
- the explicit form is not concise: it may require up to \( 2^n \) entries

Even very simple utility functions may require exponential space: e.g. the additive function mapping bundles to their cardinality (why?)

Expressive Power

The \( k \)-additive form is fully expressive, if we choose \( k \) large enough:

**Proposition 1** Any utility function is representable in \( k \)-additive form for some \( k \leq |\mathcal{R}| \).

**Proof:** For any utility function \( u \), we can define coefficients \( \alpha^X \):

\[ \alpha^X = u(\{\}) \]

\[ \alpha^X = u(X) - \sum_{T \subseteq X} \alpha^T \text{ for all } X \subseteq \mathcal{R} \text{ with } X \neq \{\} \]

Hence, \( u(X) = \sum_{T \subseteq X} \alpha^T \), which is \( k \)-additive for \( k = |\mathcal{R}| \)

The \( k \)-additive form allows for a parametrisation of synergistic effects:
- 1-additive = modular (no synergies)
- \( |\mathcal{R}| \)-additive = general (any kind of synergies)
- \( \ldots \) and everything in between

Comparative Succinctness

If two languages can express the same class of utility functions, which should we use? An important criterion is succinctness.

Let \( L \) and \( L' \) be two languages for defining utilities. We say that \( L' \) is at least as succinct as \( L \), denoted by \( L \preceq L' \), if there exist a mapping \( f: L \rightarrow L' \) and a polynomial function \( p \) such that:
- \( u \equiv f(u) \) for all \( u \in L \) (they represent the same functions); and
- \( \text{size}(f(u)) \leq p(\text{size}(u)) \) for all \( u \in L \) (polysize reduction).

Write \( L \preceq L' \) (strictly less succinct) if \( L \preceq L' \) but not \( L' \preceq L \).

Two languages can also be incomparable with respect to succinctness.
Explicit vs. $k$-additive Form

Proposition 2 The explicit and the $k$-additive form of representing utility functions are incomparable with respect to succinctness.

Proof sketch: The following two functions can be used to prove the mutual lack of a polysize reduction:

- $u_1(X) = |X|$: representing $u_1$ requires $|R|$ non-zero coefficients in the $k$-additive form (linear), but $2^{|R|} - 1$ non-zero values in the explicit form (exponential).
- $u_2(X) = 1$ for $|X| = 1$ and $u_2(X) = 0$ otherwise: requires $|R|$ non-zero values in the explicit form (linear); but $2^{|R|} - 1$ non-zero coefficients in the $k$-additive form (exponential), namely $a^2 = 1$ for $|T| = 1$, $a^2 = -2$ for $|T| = 2$, $a^2 = 3$ for $|T| = 3$, ...
Aggregating Priorities

There are several options for aggregating priorities over goals to a preference relation over alternatives:

- **Best-out ordering**: preference depends on the rank of the most important goal violated by each alternative.
- **Discrim ordering**: preference depends on the most important goal satisfied by one alternative but not by the other.
- **Leximin-ordering**: preference is defined as the lexicographic ordering over vectors that specify for each rank how many goals of that rank are satisfied by the associated alternative.

Refer to the MARA Survey for formal definitions.

Summary: Preference Representation

- Preferences of individual agents are a central parameter in the specification of a MARA problem.
- We have emphasised **expressive power** and **succinctness**:
  - expressive power should be appropriate; note that many game-theoretical results presuppose that agents can express any preference structure (e.g. whatever your true valuation, you should be able to communicate it to the auctioneer)
  - succinctness is crucial in **combinatorial domains**
- Languages considered (there are more):
  - **cardinal**: explicit form, k-additive form, weighted goals, bidding languages, and program-based representations
  - **ordinal**: explicit form, prioritised goals, and ceteris paribus statements

Social Welfare

A third parameter in the specification of a MARA problem concerns our goals: what kind of allocation do we want to achieve?

- Success may depend on a single factor (e.g. revenue of an auctioneer), but more often on an **aggregation of preferences** of the individual agents in the system.
- Concepts from Social Choice Theory and Welfare Economics can be useful here (”multiagent systems as *societies of agents*”).

We use the term **social welfare** in a very broad sense to describe metrics for assessing the quality of an allocation of resources.


Ceteris Paribus Preferences

In the language of ceteris paribus preferences, preferences are expressed as statements of the form \( C \cdot (p > q) \), meaning:

"If \( C \) is true, all other things being equal, I prefer alternatives satisfying \( p \land \neg q \) over those satisfying \( \neg p \land q \)."

The “other things” are the truth values of the propositional variables not occurring in \( p \) and \( q \). A preference relation can be constructed as the transitive closure of the union of individual preference statements.

Discussion: interesting from a cognitive point of view (close to human intuition), but of rather high complexity.

An important sublanguage of ceteris paribus preferences, imposing various restrictions on goals, are **CP-nets**.

Social Welfare

Efficiency and Fairness

When assessing the quality of an allocation (or any other agreement) we can distinguish (at least) two types of indicators of social welfare.

Aspects of **efficiency** (not in the computational sense) include:

- The chosen agreement should be such that there is no alternative agreement that would be better for some and not worse for any of the other agents (**Pareto optimality**).
- If preferences are quantitative, the sum of all payoffs should be as high as possible (**utilitarianism**).

Aspects of **fairness** include:

- The agent that is going to be worst off should be as well off as possible (**egalitarianism**).
- No agent should prefer to take the bundle allocated to one of its peers rather than keeping their own (**envy-freeness**).

Pareto Optimality

An allocation \( A \) is **Pareto-dominated** by another allocation \( A' \) iff the following hold:

- \( A \preceq_i A' \) for all agents \( i \in A \); and
- \( A \preceq_i A' \) for at least one agent \( i \in A \).

An allocation is **Pareto optimal** (or Pareto efficient) iff it is not Pareto-dominated by any other allocation.
Utilitarian Social Welfare

- Many a social welfare ordering (SWO) can be represented by means of a collective utility function (CUF).
- A CUF is a mapping from utility vectors to the reals. Here we define them directly over allocations (which induce utility vectors).
- The utilitarian collective utility function \(sw_u\) is defined as the sum of individual utilities:
  \[ sw_u(A) = \sum_{i \in \text{Agents}} u_i(A) \]
  This would be a useful metric for overall (or average) profit in an e-commerce application, for instance (\(\sim\) efficiency).
- The utilitarian CUF is zero-independent: adding a constant value to your utility function won’t affect social welfare judgements.

Egalitarian Social Welfare

- The egalitarian collective utility function \(sw_e\) is defined in terms of the agent currently worst off:
  \[ sw_e(A) = \min\{u_i(A) | i \in \text{Agents}\} \]
  Maximising this function amounts to improving the situation of the weakest members of society (\(\sim\) fairness).
- Allocation \(A'\) is strictly preferred over allocation \(A\) (by society) iff \(sw_e(A) < sw_e(A')\) holds (so-called maximin-ordering).

Ordered Utility Vectors

Every allocation \(A\) gives rise to an ordered utility vector \(\bar{u}(A)\): compute \(u_i(A)\) for all \(i \in \text{Agents}\) and present results in increasing order.

Example: \(\bar{u}(A) = (0.5, 20)\) means that the weakest agent enjoys utility 0, the strongest utility 20, and the middle one utility 5.

(Notation in the MARA Survey: \(v^i_A\))

Utilitarianism vs. Egalitarianism

- In the MAS literature the utilitarian viewpoint (that is, social welfare = sum of individual utilities) is often taken for granted.
- In philosophy/sociology/economics not.
- John Rawls’ “veil of ignorance” (A Theory of Justice, 1971): Without knowing what your position in society (class, race, sex, …) will be, what kind of society would you choose to live in?
- Reformulating the veil of ignorance for multiagent systems:
  If you were to send a software agent into an artificial society to negotiate on your behalf, what would you consider acceptable principles for that society to operate by?
- Conclusion: worthwhile to investigate egalitarian (and other) social principles also in the context of multiagent systems.

The Leximin-Ordering

We now introduce an SWO that may be regarded as a refinement of the maximin-ordering induced by the egalitarian CUF.

The leximin-ordering \(\preceq_L\) is defined as follows:
  \[ A \preceq_L A' \iff \exists h (\bar{u}(A)_h = \bar{u}(A')_h) \text{ and } \bar{u}(A)_k < \bar{u}(A')_k \]

That means:
- \(\bar{u}(A) = \bar{u}(A')\) or
- there exists a \(k \leq n\) such that
  - \(\bar{u}(A)_i = \bar{u}(A')_i\), for all \(i < k\) and
  - \(\bar{u}(A)_k < \bar{u}(A')_k\)

Example: \(A \sim_L A'\) for \(\bar{u}(A) = (0.6, 20, 29)\) and \(\bar{u}(A') = (0.6, 24, 25)\)

Nash Product

- The Nash collective utility function \(sw_N\) is defined as the product of individual utilities:
  \[ sw_N(A) = \prod_{i \in \text{Agents}} u_i(A) \]
  This is a useful measure of social welfare as long as all utility functions can be assumed to be positive.
- Like the utilitarian CUF, the Nash CUF favours increases in overall utility, but also inequality-reducing redistributions (2 · 6 < 4 · 4).
- The Nash CUF is scale independent: whether a particular agent measures their own utility in euros or yen does not affect social welfare judgements.

Rank Dictators

The \(k\)-rank dictator CUF for \(k \in A\) is mapping allocations to the utility enjoyed by the \(k\)-poorest agent:

\[ sw_k(A) = \bar{u}(A)_k \]

For \(k = 1\) this is the egalitarian CUF. For \(k = n\) we obtain an elitist CUF measuring social welfare in terms of the agent that is best off.

Generalisations

Consider the following family of CUFs, parametrised by \(p \neq 0\):

\[ sw_{(p)}(A) = \sum_{i \in \text{Agents}} g_p(u_i(A)) \quad \text{where} \quad g_p(x) = \begin{cases} x^p & \text{if } p > 0 \\ -x^p & \text{if } p < 0 \\ \log x & \text{if } p = 0 \end{cases} \]

This generalises several of our social welfare orderings:
- \(sw_{(1)}\) measures utilitarian social welfare.
- \(sw_{(0)}\) induces the same SWO as the Nash product.
- The leximin-ordering is the limit of the SWO induced by \(sw_{(p)}\) as \(p\) goes to \(-\infty\).
Ordered Weighted Averaging

Another family of CUFs are ordered weighted averaging operators. Let \( w = (w_1, w_2, \ldots, w_n) \) be a vector of real numbers. Define:

\[
sw_w(A) = \sum_{i \in A \text{valid}} w_i \cdot \overline{u}(A)_i
\]

Again, this generalises several other SWOs:

- If \( w_i = 0 \) for all \( i \neq k \) and \( w_k = 1 \), then we have exactly the \( k \)-rank dictator CUF.
- If \( w_i = 1 \) for all \( i \) then we obtain the utilitarian CUF.
- If \( w_i = \alpha^{-1} \), with \( \alpha > 0 \), then the lexicin-ordering is the limit of the SWO induced by \( sw_w \) as \( \alpha \) goes to 0.

Envy-Freeness

- An allocation is called envy-free if no agent would rather have one of the bundles allocated to any of the other agents:
  \( A(i) \succeq A(j) \)

Here, \( A(i) \) is the bundle allocated to agent \( i \) in allocation \( A \).

- Note that envy-free allocations do not always exist (at least not if we require either complete or Pareto optimal allocations).
- As we cannot always ensure envy-free allocations, one option would be to reduce envy as much as possible.
- What would be a reasonable definition of minimal envy?
  - minimise the number of envious agents
  - minimise the average degree of envy (distance to the most envied competitor) of all envious agents

Welfare Engineering

- Choice (and possibly design) of social welfare orderings that are appropriate for specific agent-based applications.
  - Example: The elitist collective utility function seems unethical for human society, but may be appropriate for a distributed application where each agent gets the same task.
  - Slogan: “welfare economics for artificial agent societies”
- Design of suitable rationality criteria and interaction mechanisms for negotiating agents in view of different notions of social welfare.
  - Example: To achieve allocations with maximal utilitarian social welfare in modular domains with money, ask agents to negotiate mutually beneficial deals over one resource at a time.
  - Slogan: “inverse welfare economics” (\( \rightarrow \) mechanism design)

Allocation Procedures

Normalised Utility

It can be useful to normalise utility functions before aggregation:

- If \( A_0 \) is the initial allocation, then we may restrict attention to allocations \( A \) that Pareto-dominate \( A_0 \) and use the utility gains \( u_i(A) - u_i(A_0) \) rather than \( u_i(A) \) as problem input.
- We could evaluate an agent’s utility gains relative to the gains it could expect in the best possible case. Define an agent’s maximum utility with respect to a set \( \text{Adm} \) of admissible allocations:
  \[
  \hat{u}_i = \max\{u_i(A) \mid A \in \text{Adm}\}
  \]

Then define the normalised individual utility of agent \( i \) as follows:

\[
\tilde{u}_i(A) = \frac{u_i(A)}{\hat{u}_i}
\]

The optimum of the lexicin-ordering with respect to normalised utilities is known as the Kalai-Smorodinsky solution.

Example

Consider the following example with two agents and three resources:

\( A = \{1, 2\} \) and \( R = \{a, b, c\} \). Suppose utility functions are additive:

\[
\begin{align*}
  u_1(a) &= 18 \\
  u_1(b) &= 12 \\
  u_1(c) &= 8 \\
  u_2(a) &= 15 \\
  u_2(b) &= 8 \\
  u_2(c) &= 12
\end{align*}
\]

Let \( A \) be the allocation giving \( a \) to agent 1 and \( b \) and \( c \) to agent 2.

- \( A \) has maximal egalitarian social welfare (18); utilitarian social welfare is not maximal (38 rather than 42); and neither is elitist social welfare (20 rather than 38).
- \( A \) is Pareto optimal as well as lexicin-optimal, but not envy-free.
- There is no allocation that would be both Pareto optimal and envy-free. If we change \( u_1(a) \) to 20 (from 18), then \( A \) becomes Pareto optimal and envy free.

Summary: Social Welfare

- Social welfare (or more generally, some aggregation of agent preferences) can be used to define goals in a MARA setting.
- There is a large range of collective utility functions and other concepts from Social Choice Theory and Welfare Economics that can be used to assess social welfare.
- To date, most work in MAS has only used the concepts of Pareto optimality and utilitarian social welfare, . . .
- . . . but other social welfare measures, in particular those related to fairness issues, are important as well.

Allocation Procedures

To solve a MARA problem, we firstly need to decide on an allocation procedure. This is a very complex issue, involving at least:

- Protocols: What types of deals are possible? What messages do agents have to exchange to agree on one such deal?
- Strategies: What strategies may an agent use for a given protocol? How can we give incentives to agents to behave in a certain way?
- Algorithms: How do we solve the computational problems faced by agents when engaged in negotiation?
Centralised vs. Distributed Negotiation

An allocation procedure to determine a suitable allocation of resources may be either centralised or distributed:

- In the centralised case, a single entity decides on the final allocation, possibly after having elicited the preferences of the other agents. Example: combinatorial auctions
- In the distributed case, allocations emerge as the result of a sequence of local negotiation steps. Such local steps may or may not be subject to structural restrictions (say, bilateral deals).

Which approach is appropriate under what circumstances?

Disadvantages of the Centralised Approach

But there are also some disadvantages of the centralised approach:

- Can we trust the centre (the auctioneer)?
- Does the centre have the computational resources required?
- Less natural to take an initial allocation into account (in an auction, typically the auctioneer owns everything to begin with).
- Less natural to model step-wise improvements over the status quo.
- Arguably, only the distributed approach is a serious implementation of the MAS paradigm.

The Contract Net Protocol

Originally developed for task decomposition and allocation, but also applicable to distributed negotiation over resources.

Each agent may assume to roles of manager and bidder. The Contract Net protocol is a one-to-many protocol matching an offer by a manager to one of potentially many bidders. There are four phases:

- Announcement phase: The manager advertises a deal to a number of partner agents (the bidders).
- Bidding phase: The bidders send their proposals to the manager.
- Assignment phase: The manager elects the best bid and assigns the resource(s) accordingly.
- Confirmation phase: The elected bidder sends a confirmation.

Auctions are centralised mechanisms for the allocation of goods amongst several agents. Agents report their preferences (bidding) and the auctioneer decides on the final allocation (and on prices).

- Distinguish direct and reverse auctions (auctioneer buying).
- Bidding may be open-cry (English) or by sealed bids.
- Open-cry: ascending (English) or descending bids (Dutch).
- Pricing rule: first-price or second-price (Vickrey).
- Combinatorial auctions: several goods, sold/bought in bundles.

Advantages of the Centralised Approach

Much recent work in the MAS community on negotiation and resource allocation has concentrated on centralised approaches, in particular on combinatorial auctions.

There are several reasons for this:

- The communication protocols required are relatively simple.
- Many results from economics and game theory, in particular on mechanism design, can be exploited.
- There has been a recent push in the design of powerful algorithms for winner determination in combinatorial auctions.

Auction Protocols

Auctions are centralised mechanisms for the allocation of goods amongst several agents. Agents report their preferences (bidding) and the auctioneer decides on the final allocation (and on prices).

- Distinguish direct and reverse auctions (auctioneer buying).
- Bidding may be open-cry (English) or by sealed bids.
- Open-cry: ascending (English) or descending bids (Dutch).
- Pricing rule: first-price or second-price (Vickrey).
- Combinatorial auctions: several goods, sold/bought in bundles.

Properties of Allocation Procedures

We may study different properties of allocation procedures:

- Termination: Is the procedure guaranteed to terminate eventually?
- Convergence: Will the final allocation be optimal according to our chosen social welfare measure?
- Incentive-compatibility: Do agents have an incentive to report their valuations truthfully? (→ mechanism design)
- Complexity results: What is the computational complexity of finding a socially optimal allocation of resources?

Next, we are going to see an example for a convergence property . . .

Negotiating Socially Optimal Allocations

We are now going to analyse a specific model of distributed negotiation (defined on the next slide).

We are not going to talk about designing a concrete negotiation protocol, but rather study the framework from an abstract point of view. The main question concerns the relationship between

- the local view: what deals will agents make in response to their individual preferences?, and
- the global view: how will the overall allocation of resources evolve in terms of social welfare?

Extensions

The immediate adaptation of the original Contract Net protocol only allows managers to advertise a single resource at a time, and a bidder can only offer money in return for that resource (not other items).

Possible extensions:

- Allow for negotiation over the exchanges of bundles of resources.
- Allow for deals without explicit utility transfer (monetary payments). The announcement phase remains the same, but bids are now about offering resources in exchange, rather than money.
- Allow agents to negotiate several deals concurrently and to decommit from deals within a certain period.
- In leveled-commitment contracts, agents are also allowed to decommit, but have to pay a pre-defined penalty.

Refer to the MARA Survey for references to these works.
An Abstract Negotiation Framework

- Finite set of agents $A$ and finite set of indivisible resources $R$.
- An allocation $A$ is a partitioning of $R$ amongst the agents in $A$.
  Example: $A(i) = \{r_x, r_y\}$ — agent $i$ owns resources $r_x$ and $r_y$.
- Every agent $i \in A$ has a utility function $u_i : 2^R \rightarrow \mathbb{R}$.
  Example: $u_i(A) = u_i(A(i)) = 577.8$ — agent $i$ is pretty happy.
- Agents may engage in negotiation to exchange resources in order to benefit either themselves or society as a whole.
- A deal $\delta = (A, A')$ is a pair of allocations (before/after).
- A deal may come with a number of side payments to compensate some of the agents for a loss in utility. A payment function is a function $p : A \rightarrow \mathbb{R}$ with $\sum_{i \in A} p(i) = 0$.
  Example: $p(i) = 5$ and $p(j) = -5$ means that agent $i$ pays $\$5$, while agent $j$ receives $\$5$.

Proof.

A rational agent (who does not plan ahead) will only accept deals that improve its individual welfare.

- A deal $\delta = (A, A')$ is called individually rational iff there exists a payment function $p$ such that $u_i(A') - u_i(A) > p(i)$ for all $i \in A$, except possibly $p(i) = 0$ for agents $i$ with $A(i) = A'(i)$.

That is, an agent will only accept a deal iff it results in a gain in utility (or money) that strictly outweighs a possible loss in money (or utility).

The Local/Individual Perspective

Suppose that as system designers we are interested in maximising utilitarian social welfare:

$$sw_u(A) = \sum_{i \in A} u_i(A)$$

Linking the Local and the Global Perspectives

It turns out that individually rational deals are exactly those deals that increase social welfare:

Lemma 4 (Rationality and social welfare) A deal $\delta = (A, A')$ with side payments is individually rational iff $sw_u(A) < sw_u(A')$.

Proof. "$\Rightarrow$": Rationality means that overall utility gains outweigh overall payments (which are $0$).

"$\Leftarrow$": The social surplus can be divided amongst all deal participants by using the following payment function:

$$p(i) = u_i(A') - u_i(A) + \frac{sw_u(A') - sw_u(A)}{|A|} \geq 0$$

Convergence

It is now easy to prove the following convergence result (originally stated by Sandholm in the context of distributed task allocation):

Theorem 5 (Sandholm, 1998) Any sequence of individually rational deals will eventually result in an allocation with maximal social welfare.

Proof. Termination follows from our lemma and the fact that the number of allocations is finite so let $A$ be the terminal allocation.

Assume $A$ is not optimal, i.e. there exists an allocation $A'$ with $sw_u(A') < sw_u(A)$. Then, by our lemma, $\delta = (A, A')$ is individually rational $\Rightarrow$ contradiction.

Agents can act locally and need not be aware of the global picture (convergence towards a global optimum is guaranteed by the theorem).

Complexity Results

- When designing a new resource allocation system, awareness of known complexity results is important to understand general limitations as well as opportunities.
- We are going to explain one complexity result in detail: welfare optimisation is NP-complete.
- Brief review of other results: focus on seeing what type of questions people have been asking, not on technical details.
Recap: Complexity Theory

- Given a class of problems parametrised by their "size", how hard it is to solve a problem of size $n$?
- Distinguish: time/space worst-case/average-case complexity
- Problems that can be solved in polynomial time ($P$) are considered tractable, problems requiring exponential time (EXPTIME) not.
- Think of a problem that requires searching through a tree. If you are lucky and go down the right branch at every branching point, you may need only polynomial time, otherwise exponential time.
- A nondeterministic algorithm is a (hypothetical) algorithm with an "oracle" that tells us which branch to explore next.
- NP is the class of decision problems that can be solved by such nondeterministic algorithms in polynomial time.

Resource Allocation Settings

We are going to analyse the case of MARA for indivisible non-sharable resources. A resource allocation setting $(A, R, U)$ is given by:

- $A = \{1, 2, \ldots, n\}$ is a set of $n$ agents;
- $R = \{r_1, r_2, \ldots, r_m\}$ is a collection of $m$ resources; and
- $U = \{u_1, u_2, \ldots, u_n\}$ describes the utility function $u_i: 2^R \rightarrow \mathbb{Q}$ for the agent $i \in A$.

The set of allocations $A$ is the set of partitionings of $R$ amongst $A$ (or equivalently, the set of total functions from $R$ to $A$).

Proof of NP-hardness

We are going to reduce our problem to Set Packing, one of the standard problems known to be NP-complete:

**Set Packing**

Instance: Collection $C$ of finite sets and $K \in \mathbb{Q}$

Question: Is there a collection of disjoint sets $C' \subseteq C$ s.t. $|C'| > K$?

Given an instance $C$ of Set Packing, consider this MARA setting:

- Resources: each item in one of the sets in $C$ is a resource
- Agents: one for each set in $C$ and one other agent (called 0)
- Utilities: $w(R) = 1$ if $R = C$ and $w(R) = 0$ otherwise; $u_i(R) = 0$ for all bundles $R$.

That is, every agent values "its" bundle at 1 and every other bundle at 0. Agent 0 values all bundles at 0.

- Any algorithm for WO can solve also Set Packing problems; so WO must be at least NP-hard.

Representation Issues

- As for all complexity results, the representation of the input problem is crucial: if the input problem is represented inefficiently (e.g. using exponential space when this is not required), then complexity results (expressed with respect to the size of the input) may seem much more favourable than they really are.
- NP-completeness of Welfare Optimisation has been shown with respect to several representations of utilities ($k$-additive form, straight-line programs).
- In the sequel, the focus is on demonstrating what questions people have been asking rather than on exact complexity results. Therefore, we do not give details regarding representation (but most results apply to a variety of representation forms).

Welfare Optimisation

How hard is it to find an allocation with maximal social welfare? Rephrase this optimisation problem as a decision problem:

**Welfare Optimisation (Welfare Improvement)**

Instance: $(A, R, U), K \in \mathbb{Q}$

Question: Is there an allocation $A$ such that $sw_u(A) > K$?

Unfortunately, the problem is intractable:

**Theorem 6 Welfare Optimisation is NP-complete.**

The proof (next slide) uses a reduction from a standard reference problem (Set Packing) known to be NP-complete. In the context of MARA, this kind of result seems to have first been stated by Rothkopf et al. (1998).


Proof of Membership in NP

This part is in fact very easy …

Recall that a problem belongs to NP if it is possible to verify the correctness of a candidate solution in polynomial time. This is clearly the case here: Given an allocation $A$, we can compute $sw_u(A)$ in polynomial time. $A$ is a good solution iff $sw_u(A) > K$.

Welfare Improvement

The following problem is also NP-complete:

**Welfare Improvement (WI)**

Instance: $(A, R, U)$; allocation $A$

Question: Is there an allocation $A'$ such that $sw_u(A) < sw_u(A')$?

Given the close connection to Welfare Optimisation, this is not very surprising.
**Pareto Optimality**

A decision problem is said to be in coNP iff its complementary problem ("is it not the case that ") is in NP. Checking whether a given allocation is Pareto optimal is an example for a coNP-complete decision problem:

<table>
<thead>
<tr>
<th>Pareto Optimality (PO)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance: ( (A, R, \mathcal{U}) ); allocation ( A )</td>
</tr>
<tr>
<td>Question: Is ( A ) Pareto optimal?</td>
</tr>
</tbody>
</table>

---

**Path and Convergence Properties**

Related to the distributed negotiation framework introduced earlier, we can ask whether an allocation with certain characteristics is reachable using only deals meeting certain conditions (\( \Phi \)-deals).

<table>
<thead>
<tr>
<th>( \Phi )-Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance: ( (A, R, \mathcal{U}) ); allocations ( A ) and ( A' ) with ( su(A) &lt; su(A') )</td>
</tr>
<tr>
<td>Question: Is there a sequence of ( \Phi )-deals leading from ( A ) to ( A' )?</td>
</tr>
</tbody>
</table>

One of several known results is that \( \Phi \)-Path is PSPACE-complete in case \( \Phi \) is the predicate selecting all individually rational 1-deals (involving just a single resource each).

A related problem, \( \Phi \)-Convergence, asks whether any given sequence of \( \Phi \)-deals would result in a socially optimal allocation.

---

**Simulation Platforms**

The MARA Survey also discusses simulation platforms, which can be useful tools to test hypotheses experimentally, when it is difficult or impossible to obtain the desired theoretical results.

- Issues: simulation vs. implementation, simulating time, agent modelling, extensibility and integration
- Examples of systems: Swarm, RePast, and others

---

**Envy-Freeness**

Checking whether a given setting admits an envy-free allocation (if all goods need to be allocated) is again NP-complete:

<table>
<thead>
<tr>
<th>Envy-Freeness (EF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance: ( (A, R, \mathcal{U}) )</td>
</tr>
<tr>
<td>Question: Is there a (complete) allocation ( A ) that is envy-free?</td>
</tr>
</tbody>
</table>

Checking whether there is an allocation that is both Pareto optimal and envy-free is even harder: \( \Sigma^P_2 \)-complete (NP with NP oracle).

---

**Aspects of Complexity**

For concrete allocation procedures (rather than abstract optimisation problems), communication complexity becomes an issue . . .

(1) How many deals are required to reach an optimal allocation?
   - communication complexity as number of individual deals
(2) How many dialogue moves are required to agree on one such deal?
   - affects communication complexity as number of dialogue moves
(3) How expressive a communication language do we require?
   - Minimum requirements: performatives propose, accept, reject
   - content language to specify multilateral deals
   - affects communication complexity as number of bits exchanged
(4) How complex is the reasoning task faced by an agent when deciding on its next dialogue move?
   - computational complexity (local rather than global view)

---

**Summary: Complexity Results**

- There are many problems in MARA with interesting complexity questions: computational and communication complexity.
- Decision questions naturally arising in MARA are often intractable: NP-hard or worse.
- Successful algorithm design is still possible, but simple brute force approaches won’t work.
- Sometimes negative complexity results can be avoided by imposing restrictions (on utilities for instance).

---

**Mechanism Design**

An important topic that we have not covered is the game-theoretical analysis of MARA problems, in particular mechanism design.

- While game theory analyses the strategic behaviour of rational agents in a given game, mechanism design uses these insights to design games inducing certain strategies (and hence outcomes).
- A central result is the incentive-compatibility of reporting your true valuation in the Vickrey-Clarke-Groves mechanism (which is a generalisation of second-price auctions).

Varian (1995) gives an easy-reading introduction to mechanism design accessible to computer scientists.


---

**Conclusions**
Algorithm Design

Another important topic that we have not covered concerns the algorithmic aspects of designing allocation procedures. Most work to date has concentrated on centralised approaches (auctions):

- The Winner Determination Problem (WDP) in combinatorial auctions is close to Welfare Optimisation.
- The WDP can be tackled using both off-the-shelf mathematical programming software and specialised AI search techniques.
- While the WDP is also an NP-hard problem, these approaches often work well in practice, even for larger problem instances.

In principle, similar ideas could be used also for distributed negotiation (to support the individual agents with their decision making) . . .


Summary

We have given an overview of the MARA research area:

- Specifying a MARA problem requires fixing at least the following parameters: type of resource, agent preferences, social welfare or similar concept used to define global aims.
- To design a solution method for a given class of MARA problems: choose either a centralised or a distributed allocation procedure; take care of the algorithmic aspects of the problem, considering known complexity results; use mechanism design techniques to achieve incentive-compatibility; and use social simulation to better understand and refine your procedure.
- Applications include industrial procurement, earth observation satellites, manufacturing systems, and grid computing.

MARA is an exciting and timely area of research, with many open problems still to be addressed, in all of the subareas discussed . . .