# Multiagent Resource Allocation

Ulle Endriss
ILLC
University of Amsterdam

Nicolas Maudet
LAMSADE
University Paris-Dauphine





## Table of Contents







## Introduction





## Tutorial Information

**Lecturers:** Ulle Endriss (`ulle@illc.uva.nl`)
Nicolas Maudet (`maudet@lamsade.dauphine.fr`)

**Notes:** This tutorial is based on the "MARA Survey":

Y. Chevaleyre, P.E. Dunne, U. Endriss, J. Lang, M. Lemaître, N. Maudet, J. Padget, S. Phelps, J.A. Rodríguez-Aguilar & P. Sousa. Issues in Multiagent Resource Allocation. *Informatica*, 30:3–31, 2006.

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/easss-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?





## Examples of Application Areas

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

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





## 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*





## 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?





## 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 $\leadsto$ *discretisation*).
- *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.





## 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*





## 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





## 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.





## 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).





## 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).





## 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 . . .





## Preference Representation Languages

Some central questions that arise when we have to choose a preference representation language:

- *Cognitive relevance:* How close is a given language to the way in which humans would express their preferences?
- *Elicitation:* How difficult is it to elicit the preferences of an agent so as to represent them in the chosen language?
- *Expressive power:* Can the chosen language encode all the preference structures we are interested in?
- *Succinctness:* Is the representation of (typical) preference structures succinct? Is one language more succinct than the other?
- *Complexity:* What is the computational complexity of related decision problems, such as comparing two alternatives?

We are going to concentrate on expressive power and succinctness.





## 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 $u : \mathcal{X} \to \mathit{Val}$, where $\mathit{Val}$ 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 $\mathcal{X} = 2^{\mathcal{R}}$.





## Some Observations

- *Intrapersonal comparison:* ordinal and cardinal preferences allow for comparing the satisfaction of an agent for different alternatives
- *Interpersonal comparison:* ordinal preferences don't allow for interpersonal comparison ("Ann likes $x$ more than Bob likes $y$")
- *Preference intensity:* ordinal preferences cannot express preference intensity; cardinal preferences can (subject to $\mathit{Val}$ being numerical)
- *Representability:* a connected ordinal preference relation $\preceq$ is representable by a utility function $u$: $x \preceq y$ iff $u(x) \leq u(y)$
- *Cognitive relevance:* hard to make general statements, but at least ordinal preferences don't require reasoning with numerical utilities
- *Explicit representation:* the explicit representations of cardinal and ordinal preferences have space complexity $O(|\mathcal{X}|)$ and $O(|\mathcal{X}|^2)$, respectively (why?)





## Example

Old story: hanging a frame ($f$) with a hammer ($h$) and a nail ($n$) . . .

| $B$ | $u_i(B)$ |
|---|---|
| $\{\}$ | 0 |
| $\{f\}$ | 10 |
| $\{h\}$ | 5 |
| $\{n\}$ | 0 |
| $\{f,n\}$ | 10 |
| $\{f,h\}$ | 15 |
| $\{h,n\}$ | 8 |
| $\{f,h,n\}$ | 20 |





## Example

Old story: hanging a frame ($f$) with a hammer ($h$) and a nail ($n$) . . .

| $\succeq$ | $\{\}$ | $\{f\}$ | $\{h\}$ | $\{n\}$ | $\{f,n\}$ | $\{f,h\}$ | $\{h,n\}$ | $\{f,h,n\}$ |
|---|---|---|---|---|---|---|---|---|
| $\{\}$ | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| $\{f\}$ | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| $\{h\}$ | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| $\{n\}$ | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| $\{f,n\}$ | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| $\{f,h\}$ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| $\{h,n\}$ | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| $\{f,h,n\}$ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |





## 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 $u : 2^{\mathcal{R}} \to \mathbb{R}$ 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?)





## The $k$-additive Form

- A utility function is called *$k$-additive* iff the utility assigned to a bundle $X$ can be represented as the sum of basic utilities assigned to subsets of $X$ with cardinality $\leq k$ (*limited synergies*).
- The *$k$-additive form* of representing utility functions:

$$u(X) \quad = \quad \sum_{T \subseteq X} \alpha^T \qquad \text{with } \alpha^T = 0 \text{ whenever } |T| > k$$

  Example: $u = 3.x_1 + 7.x_2 - 2.x_2.x_3$ is a 2-additive function
- That is, specifying a utility function in this language means specifying the *coefficients $\alpha^T$* for bundles $T \subseteq \mathcal{R}$.
- The value $\alpha^T$ can be seen as the additional benefit incurred from owning the items in $T$ *together*, *i.e.* beyond the benefit of owning all proper subsets.

## 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 $\alpha^X$:

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

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

The $k$-additive form allows for a *parametrisation* of synergetic effects:

- 1-additive = modular (no synergies)
- $|\mathcal{R}|$-additive = general (any kind of synergies)
- ... and everything in between





## Example

Old story: hanging a frame ($f$) with a hammer ($h$) and a nail ($n$) ...

| $B$ | $u_i(B)$ |
|---|---|
| $\{f\}$ | 10 |
| $\{h\}$ | 5 |
| $\{f,n\}$ | 10 |
| $\{f,h\}$ | 15 |
| $\{h,n\}$ | 8 |
| $\{f,h,n\}$ | 20 |

▶ $u = 10.f + 5.h + 3.h.n + 2.f.h.n$ (3-additive)





## Example

Old story: hanging a frame ($f$) with a hammer ($h$) and a nail ($n$) ...

| $B$ | $u_i(B)$ |
|---|---|
| $\{f\}$ | 10 |
| $\{h\}$ | 5 |
| $\{f,n\}$ | 10 |
| $\{f,h\}$ | 15 |
| $\{h,n\}$ | 8 |
| $\{f,h,n\}$ | 18 |

▶ $u = 10.f + 5.h + 3.h.n$ (2-additive)





## 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 $L \preceq L'$, iff there exist a mapping $f : L \to L'$ and a *polynomial* function $p$ such that:

- $u \equiv f(u)$ for all $u \in L$ (they represent the same functions); and
- $size(f(u)) \leq p(size(u))$ for all $u \in L$ (polysize reduction).

Write $L \prec L'$ (strictly less succinct) iff $L \preceq L'$ 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 $|\mathcal{R}|$ non-zero coefficients in the $k$-additive form (*linear*); but $2^{|\mathcal{R}|} - 1$ non-zero values in the explicit form (*exponential*).
- $u_2(X) = 1$ for $|X| = 1$ and $u_2(X) = 0$ otherwise: requires $|\mathcal{R}|$ non-zero values in the explicit form (*linear*); but $2^{|\mathcal{R}|} - 1$ non-zero coefficients in the $k$-additive form (*exponential*), namely $\alpha^T = 1$ for $|T| = 1$, $\alpha^T = -2$ for $|T| = 2$, $\alpha^T = 3$ for $|T| = 3$, ...

Y. Chevaleyre, U. Endriss, S. Estivie, and N. Maudet. *Multiagent Resource Allocation with k-additive Utility Functions.* DIMACS-LAMSADE Workshop 2004.





## Weighted Propositional Formulas

Notation: finite set of propositional letters $PS$ (representing goods); propositional language $\mathcal{L}_{PS}$ over $PS$ can describe requirements.

A *goal base* is a set $G = \{(\varphi_i, \alpha_i)\}_i$ of pairs, each consisting of a consistent propositional formula $\varphi_i \in \mathcal{L}_{PS}$ and a real number $\alpha_i$. The utility function $u_G$ generated by $G$ is defined by

$$u_G(M) = \sum \{\alpha_i \mid (\varphi_i, \alpha_i) \in G \text{ and } M \models \varphi_i\}$$

for all models $M \in 2^{PS}$. $G$ is called the *generator* of $u_G$.
Example: $G = \{(f, 10), (h, 5), (h \wedge n, 3), (f \wedge h \wedge n, 2)\}$
($\rightsquigarrow$ But isn't that exactly the $k$-additive representation?)
▶ $u = 10.f + 5.h + 3.h.n + 2.f.h.n$

J. Lang. *Logical Preference Representation and Combinatorial Vote.* Annals of Mathematics and Artificial Intelligence, 42(1–3):37–71, 2004.





## Weighted Propositional Formulas

Notation: finite set of propositional letters $PS$ (representing goods); propositional language $\mathcal{L}_{PS}$ over $PS$ can describe requirements.

A *goal base* is a set $G = \{(\varphi_i, \alpha_i)\}_i$ of pairs, each consisting of a consistent propositional formula $\varphi_i \in \mathcal{L}_{PS}$ and a real number $\alpha_i$. The utility function $u_G$ generated by $G$ is defined by

$$u_G(M) = \sum \{\alpha_i \mid (\varphi_i, \alpha_i) \in G \text{ and } M \models \varphi_i\}$$

for all models $M \in 2^{PS}$. $G$ is called the *generator* of $u_G$.
Example: $G = \{(f, 10), (h, 5), (h \wedge n, 3), (f \wedge (h \wedge n \vee g), 2)\}$
($\rightsquigarrow$ No! Recall we can use any well-formed logical formula ...)
▶ $u = 10.f + 5.h + 3.h.n + 2.f.h.n + 2.f.g - 2.f.h.n.g$

J. Lang. *Logical Preference Representation and Combinatorial Vote.* Annals of Mathematics and Artificial Intelligence, 42(1–3):37–71, 2004.





## Bidding Languages

Bidding languages are preference representation languages developed specifically for combinatorial auctions, to allow bidders to transmit their valuations to the auctioneer.

- Bids are combinations of *atomic bids* $\langle R, p \rangle$: bundle/price.
- In the *OR-language*, the valuation is taken to be the maximal value that can be obtained by accepting disjoint bids. Example:

  $$\langle \{a\}, 2 \rangle \text{ OR } \langle \{b\}, 2 \rangle \text{ OR } \langle \{c\}, 1 \rangle \text{ OR } \langle \{a, b\}, 5 \rangle$$

  That is, an OR-combination of two bids defining valuations $v_1$ and $v_2$ defines the following valuation:

  $$(v_1 \text{ OR } v_2)(X) = \max_{X_1 \subseteq X} (v_1(X_1) + v_2(X \setminus X_1))$$

N. Nisan. *Bidding Languages for Combinatorial Auctions.* In P. Cramton *et al.* (eds.), Combinatorial Auctions, MIT Press, 2006.

## Example

Old story: hanging a frame ($f$) with a hammer ($h$) and a nail ($n$) ...

| $B$ | $u_i(B)$ |
|---|---|
| $\{f\}$ | 10 |
| $\{h\}$ | 5 |
| $\{f,n\}$ | 10 |
| $\{f,h\}$ | 15 |
| $\{h,n\}$ | 8 |
| $\{f,h,n\}$ | 20 |

► $\langle\{f\},10\rangle$ OR $\langle\{h\},5\rangle$ OR $\langle\{h,n\},8\rangle$ OR $\langle\{h,f,n\},20\rangle$





## Bidding Languages (cont.)

- In the *XOR-language*, atomic bids by the same bidder are assumed to be mutually exclusive. Example:

  $$\langle\{a\},3\rangle \text{ XOR } \langle\{b\},3\rangle \text{ XOR } \langle\{a,b\},5\rangle$$

  So an XOR-combination of $v_1$ and $v_2$ has the following semantics:

  $$(v_1 \text{ XOR } v_2)(X) \quad = \quad \max\{v_1(X), v_2(X)\}$$

- XOR can represent all (normalised and monotonic) valuations, while OR can only represent supermodular valuations.
- *OR/XOR-language:* arbitrary combinations of OR and XOR
- The *OR*-language* is like the OR-language, but *dummy items* can be used to express exclusiveness constraints. Example:

  $$\langle\{a, \mathit{dummy}\},3\rangle \text{ OR } \langle\{b, \mathit{dummy}\},3\rangle \text{ OR } \langle\{a,b,\mathit{dummy}\},5\rangle$$





## Ordinal Preferences: Explicit Representation

Next we are going to look into different languages for representing *ordinal* preference structures.

The *explicit representation* of an ordinal preference relation $\preceq$ over $2^n$ alternatives has a space complexity of $O(2^n \cdot 2^n)$: for each pair of bundles, say which one is preferred.





## Prioritised Goals

Again, associate goods with propositional letters in $PS$ and bundles with models $M \in 2^{PS}$. *Goals* can be expressed as formulas in the propositional language $\mathcal{L}_{PS}$.

Instead of weights, we now have a *priority relation* over goals. Assuming this priority relation is a total order, it can be represented by a function $\mathit{rank} : \mathbb{N} \to \mathbb{N}$ mapping each (index of a) goal to its rank. By convention, a *lower rank* means *higher priority*.

A *goal base* is now a finite set of goals with an associated rank function: $G = \langle\{\varphi_1, \ldots, \varphi_m\}, \mathit{rank}\rangle$.

► Ideally, all goals will get satisfied. But if not, how can we extend the priority relation over goals to a preference relation over alternatives?





## 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.
- *Discrimin 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.





## Example

Old story: hanging a frame ($f$) with a hammer ($h$) and a nail ($n$) ...

| $\mathit{rank}$ | $B$ |
|---|---|
| 0 | $f \wedge h \wedge n$ |
| 1 | $f$ |
| 2 | $h \wedge n$ |
| 3 | $h$ |

Comparing e.g. $a_1 \equiv \{f\}$ and $a_2 \equiv \{h,n\}$ ...

- *best-out :* both violate $f \wedge h \wedge n$ (rank 0) $\rightsquigarrow$ $a_1 \sim a_2$
- *discrimin:* $a_1$ satisfies $f$ (rank 1) while $a_2$ does not $\rightsquigarrow$ $a_1 \succ a_2$
- *leximin:* $\langle 0,1,0,0\rangle$ lex. dominates $\langle 0,0,1,1\rangle$ $\rightsquigarrow$ $a_1 \succ a_2$





## Ceteris Paribus Preferences

In the language of *ceteris paribus* preferences, preferences are expressed as statements of the form $C : \varphi > \varphi'$, meaning:

> *"If $C$ is true, all other things being equal, I prefer alternatives satisfying $\varphi \wedge \neg\varphi'$ over those satisfying $\neg\varphi \wedge \varphi'$."*

The "other things" are the truth values of the propositional variables not occurring in $\varphi$ and $\varphi'$. 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*.

C. Boutilier *et al.* *CP-nets: A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements.* JAIR, 21:135–191, 2004.





## 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, and bidding languages
  - *ordinal:* explicit form, prioritised goals, and ceteris paribus statements

## Social Welfare





## 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.

H. Moulin. *Axioms of Cooperative Decision Making*. CUP, 1988.





## 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*).





## Notation

- Set of agents $\mathcal{A} = \{1, \ldots, n\}$
- Agents have preferences over allocations:
  - ordinal: $A \preceq_i A'$ means agent $i$ likes $A$ no less than $A'$
  - cardinal: $u_i(A) = x \in \mathbb{R}$ means agent $i$ assigns utility $x$ to $A$
- Remark: Preferences over allocation *could* be induced by preferences over bundles (no externalities), but this does in fact not affect our definitions (with one exception: 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 $i \in \mathcal{A}$; and
- $A \prec_i A'$ for at least one agent $i \in \mathcal{A}$.

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:

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

  This would be a useful metric for overall (or average) profit in an $e$-commerce application, for instance ($\leadsto$ 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:

$$sw_e(A) \quad = \quad min\{u_i(A) \mid i \in \mathcal{A}gents\}$$

  Maximising this function amounts to improving the situation of the weakest members of society ($\leadsto$ fairness).
- Allocation $A'$ is strictly preferred over allocation $A$ (by society) iff $sw_e(A) < sw_e(A')$ holds (so-called *maximin*-ordering).





## 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.

## Nash Product

- The *Nash collective utility function* $sw_N$ is defined as the product of individual utilities:

$$sw_N(A) \quad = \quad \prod_{i \in \mathcal{A}gents} u_i(A)$$

  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 ($2 \cdot 6 < 4 \cdot 4$).
- The Nash CUF is *scale independent:* whether a particular agent measures their own utility in euros or yen does not affect social welfare judgements.





## Ordered Utility Vectors

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

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

(Notation in the MARA Survey: $v_A^{\uparrow}$)





## Rank Dictators

The *$k$-rank dictator* CUF for $k \in \mathcal{A}$ is mapping allocations to the utility enjoyed by the $k$-poorest agent:

$$sw_k(A) \quad = \quad \vec{u}(A)_k$$

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.





## 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* $\preceq_\ell$ is defined as follows:

$A \preceq_\ell A' \;\Leftrightarrow\; \vec{u}(A)$ lexically precedes $\vec{u}(A')$ (not necessarily strictly)

That means:

- $\vec{u}(A) = \vec{u}(A')$ or
- there exists a $k \leq n$ such that
  - $\vec{u}(A)_i = \vec{u}(A')_i$ for all $i < k$ and
  - $\vec{u}(A)_k < \vec{u}(A')_k$

Example: $A \prec_\ell A'$ for $\vec{u}(A) = \langle 0, 6, 20, 29 \rangle$ and $\vec{u}(A') = \langle 0, 6, 24, 25 \rangle$





## Ordered Weighted Averaging

We can build families of parametrised CUFs that induce several SWOs. An example are the *ordered weighted averaging operators*.

Let $w = (w_1, w_2, \ldots, w_n)$ be a vector of real numbers. Define:

$$sw_w(A) \quad = \quad \sum_{i \in \mathcal{A}gents} w_i \cdot \vec{u}(A)_i$$

This generalises several other SWOs:

- If $w$ such that $w_i = 0$ 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 $w_i = \alpha^{i-1}$, with $\alpha > 0$, then the *leximin*-ordering is the limit of the SWO induced by $sw_w$ as $\alpha$ goes to $0$.





## 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 $Adm$ of admissible allocations:

$$\widehat{u}_i \quad = \quad \max\{u_i(A) \,|\, A \in Adm\}$$

  Then define the *normalised* individual utility of agent $i$ as follows:

$$u'_i(A) \quad = \quad \frac{u_i(A)}{\widehat{u}_i}$$

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





## Envy-Freeness

- An allocation is called *envy-free* iff no agent would rather have one of the bundles allocated to any of the other agents:

$$A(i) \quad \succeq_i \quad A(j)$$

  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





## Example

Consider the following example with two agents and three resources: $\mathcal{A} = \{1, 2\}$ and $\mathcal{R} = \{a, b, c\}$. Suppose utility functions are additive:

| | | |
|---|---|---|
| $u_1(\{a\}) = 18$ | $u_1(\{b\}) = 12$ | $u_1(\{c\}) = 8$ |
| $u_2(\{a\}) = 15$ | $u_2(\{b\}) = 8$ | $u_2(\{c\}) = 12$ |

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 *leximin-optimal*, but not *envy-free*.
- There is no allocation that would be both Pareto optimal *and* envy-free. But if we change $u_1(\{a\}) = 20$ (from 18), then $A$ becomes Pareto optimal and envy free.

## 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" ($\rightsquigarrow$ mechanism design)





## 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





## 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?





## 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.





## 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*.





## 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.

R.P. McAfee and J. McMillan. *Auctions and Bidding*. Journal of Economic Literature, 25:699–738, 1987.

P. Cramton, Y. Shoham, and R.Steinberg (eds.). *Combinatorial Auctions*. MIT Press, 2006.

## 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.

R.G. Smith. *The Contract Net Protocol: High-level Communication and Control in a Distributed Problem Solver.* IEEE Trans. on Computers, 29:1104–1113, 1980.





## 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 explict utility transfers* (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 *levelled-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.





## 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? ($\rightsquigarrow$ *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?

U. Endriss, N. Maudet, F. Sadri and F. Toni. *Negotiating Socially Optimal Allocations of Resources.* JAIR, 25:315–348, 2006.





## An Abstract Negotiation Framework

- Finite set of *agents* $\mathcal{A}$ and finite set of indivisible *resources* $\mathcal{R}$.
- An *allocation* $A$ is a partitioning of $\mathcal{R}$ amongst the agents in $\mathcal{A}$.
  Example: $A(i) = \{r_5, r_7\}$ — agent $i$ owns resources $r_5$ and $r_7$
- Every agent $i \in \mathcal{A}$ has got a *utility function* $u_i : 2^{\mathcal{R}} \to \mathbb{R}$.
  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 $p : \mathcal{A} \to \mathbb{R}$ with $\sum_{i \in \mathcal{A}} p(i) = 0$.
  Example: $p(i) = 5$ and $p(j) = -5$ means that agent $i$ *pays* €5, while agent $j$ *receives* €5.





## The Local/Individual Perspective

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 $i \in \mathcal{A}$, 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 Global/Social Perspective

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

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





## Example

Let $\mathcal{A} = \{ann, bob\}$ and $\mathcal{R} = \{chair, table\}$ and suppose our agents use the following utility functions:

$$\begin{array}{rclrcl} u_{ann}(\{\,\}) & = & 0 & u_{bob}(\{\,\}) & = & 0 \\ u_{ann}(\{chair\}) & = & 2 & u_{bob}(\{chair\}) & = & 3 \\ u_{ann}(\{table\}) & = & 3 & u_{bob}(\{table\}) & = & 3 \\ u_{ann}(\{chair, table\}) & = & 7 & u_{bob}(\{chair, table\}) & = & 8 \end{array}$$

Furthermore, suppose the initial allocation of resources is $A_0$ with $A_0(ann) = \{chair, table\}$ and $A_0(bob) = \{\,\}$.

► Social welfare for allocation $A_0$ is 7, but it could be 8. By moving only a *single* resource from agent $ann$ to agent $bob$, the former would lose more than the latter would gain (not individually rational).
The only possible deal would be to move the whole *set* $\{chair, table\}$.





## Linking the Local and the Global Perspectives

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

**Lemma 3 (Rationality and social welfare)** *A deal $\delta = (A, A')$ with side payments is individually rational iff $sw_u(A) < sw_u(A')$.*

*Proof.* "⇒": Rationality means that overall utility gains outweigh overall payments (which are $= 0$).

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

$$p(i) \;=\; u_i(A') \;-\; u_i(A) \;-\; \underbrace{\frac{sw_u(A') - sw_u(A)}{|\mathcal{A}|}}_{>0}$$

□

## Convergence

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

**Theorem 4 (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).

T. Sandholm. *Contract Types for Satisficing Task Allocation: I Theoretical Results.* AAAI Spring Symposium 1998.





## Summary: Allocation Procedures

- Distinguish *centralised* and *distributed* approaches to MARA.
- We have briefly introduced *auction protocols*. There is a rich literature on this topic and nowadays a lot of research on auctions is taking place in the MAS community.
- The *Contract Net protocol* can be used to take care of the communication requirements in distributed negotiation and provides means to help agents to identify possible deals, at least for structurally simple deals (e.g. bilateral deals).
- We have analysed distributed negotiation from an abstract point of view: there are some nice correspondences between the *local* level (rational deals) and the *global* level (social improvements). *Convergence* to a social optimum can be guaranteed in theory, but requires very expressive negotiation protocols in practice.





## Complexity Results





## 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.





## Recap: Complexity Theory (cont.)

- Equivalent definition: NP is the class of problems for which a candidate solution can be *verified* in (determ.) polynomial time.
- A decision problem is *NP-hard* iff it is at least as hard as any of the problems in NP.
- A decision problem is *NP-complete* iff it is NP-hard and in NP.
- We do not know whether P = NP, but strongly suspect P ≠ NP.
- NP-complete problems are generally considered intractable. Unless P = NP, there can be no general algorithm solving NP-complete problems efficiently.
- As a rule of thumb, NP-completeness means that a naïve approach won't work, but a sophisticated algorithm may well give good results in practice.





## Resource Allocation Settings

We are going to analyse the case of MARA for indivisible non-sharable resources. A resource allocation setting $\langle \mathcal{A}, \mathcal{R}, \mathcal{U} \rangle$ is given by:

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

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





## Welfare Optimisation

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

> WELFARE OPTIMISATION (WO)
> **Instance:** $\langle \mathcal{A}, \mathcal{R}, \mathcal{U} \rangle$; $K \in \mathbb{Q}$
> **Question:** Is there an allocation $A$ such that $sw_u(A) > K$?

Unfortunately, the problem is intractable:

**Theorem 5** 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).

M.H. Rothkopf, A. Pekeč, and R.M. Harstad. *Computationally Manageable Combinational Auctions.* Management Science, 44(8):1131–1147, 1998.

## 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 $\mathcal{C}$ of finite sets and $K \in \mathbb{Q}$
> **Question:** Is there a collection of disjoint sets $\mathcal{C}' \subseteq \mathcal{C}$ s.t. $|\mathcal{C}'| > K$?

Given an instance $\mathcal{C}$ of SET PACKING, consider this MARA setting:

- Resources: each item in one of the sets in $\mathcal{C}$ is a resource
- Agents: one for each set in $\mathcal{C}$ + one other agent (called $0$)
- Utilities: $u_C(R) = 1$ if $R = C$ and $u_C(R) = 0$ otherwise; $u_0(R) = 0$ 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 $0$ (to model "free disposal" in SET PACKING).

▶ Any algorithm for WO can also solve SET PACKING problems; so WO must be at least NP-hard. ✓





## 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 $sw_u(A) > K$. ✓





## 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* (such as the $k$-additive form).
- 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 the representation (but most results apply to a variety of representation forms).





## Welfare Improvement

The following problem is also NP-complete:

> WELFARE IMPROVEMENT (WI)
> **Instance:** $\langle \mathcal{A}, \mathcal{R}, \mathcal{U} \rangle$; 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:

> PARETO OPTIMALITY (PO)
> **Instance:** $\langle \mathcal{A}, \mathcal{R}, \mathcal{U} \rangle$; allocation $A$
> **Question:** Is $A$ Pareto optimal?





## Envy-Freeness

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

> ENVY-FREENESS (EF)
> **Instance:** $\langle \mathcal{A}, \mathcal{R}, \mathcal{U} \rangle$
> **Question:** Is there a (complete) allocation $A$ that is envy-free?

Checking whether there is an allocation that is both Pareto optimal and envy-free is even harder: $\Sigma_2^p$-complete (NP with NP oracle).

S. Bouveret and J. Lang. *Efficiency and Envy-freeness in Fair Division of Indivisible Goods: Logical Representation and Complexity.* IJCAI-2005.





## Path and Convergence Properties

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

> $\Phi$-PATH
> **Instance:** $\langle \mathcal{A}, \mathcal{R}, \mathcal{U} \rangle$; allocations $A$ and $A'$ with $sw_u(A) < sw_u(A')$
> **Question:** Is there a sequence of $\Phi$-deals leading from $A$ to $A'$?

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.

P.E. Dunne and Y. Chevaleyre. *Negotiation Can be as Hard as Planning.* Technical Report ULCS-05-009, Dept. of Computer Science, University of Liverpool, 2005.





## 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-had 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).





## Conclusions





## 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





## 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.

H. Varian. *Mechanism Design for Computerized Agents.* Usenix Workshop on Electronic Commerce, 1995.





## 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) . . .

T. Sandholm. *Optimal Winner Determination Algorithms.* In P. Cramton *et al.* (eds.), Combinatorial Auctions, MIT Press, 2006.





## 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 . . .