Multiagent Systems: Spring 2006

Ulle Endriss
Institute for Logic, Language and Computation
University of Amsterdam

Multiagent Resource Allocation

Most previous lectures have been concerned with a specific aspect of the multiagent resource allocation problem: bilateral negotiation, basic auctions, combinatorial auctions, mechanism design, preference representation, bidding languages, and distributed negotiation.

The aim of this lecture is to present the field of Multiagent Resource Allocation in a systematic fashion and to also cover some of the issues left open in previous classes.

We are mostly going to follow the MARA Survey . . .

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

Plan for Today

- 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:
  - Different *allocation procedures* (centralised/distributed)
  - Some *complexity results* concerning allocation procedures
  - Strategic considerations: *mechanism design*
  - Algorithmic considerations: *algorithm design*
- Short presentation of some typical *application areas*

Types of Resources

- A central parameter in any resource allocation problem is the nature of the resources themselves.
- In this course, we have mostly been concerned with indivisible resources that can be owned by at most one agent each.
- But 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 $\rightsquigarrow$ *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 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

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

Agents may have preferences over

- the bundle of resources they receive
- the bundle of resources received by others (*externalities*)

What are suitable languages for representing agent preferences? Issues to consider include *cognitive relevance*, *elicitation*, *expressive power*, *succinctness*, and *computational complexity*.

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

Cardinal Preferences

We have discussed the following languages for expressing cardinal preferences (*i.e.* utility functions or valuations):

- The *explicit form:* list the utility of each bundle.
- The *k-additive form:* list the marginal utility of each bundle with cardinality $\leq k$ (also fully expressive, but often more succinct).
- *Weighted propositional formulas:* associate each good with a propositional letter and assign weights to propositional formulas (utility defined as sum of weights of satisfied formulas).
- *Bidding languages:* combinations of atomic bids using OR and XOR; use of dummy items to encode exclusiveness constraints.
- *Program-based representations:* straight-line programs

Ordinal Preferences

- *Explicit representation:* for each pair of alternatives, specify the preference of the agent.
- *Prioritised goals:* associate each good with a propositional letter and specify priority relation over formulas (ranking). Different forms of aggregation yield different preference languages:
  - *Best-out ordering:* what is the most important goal violated by an alternative (absolute)?
  - *Discrimin ordering:* what is the most important goal violated by one but not the other alternative (relative)?
  - *Leximin ordering:* lexicographic ordering over vectors specifying how many goals of each level of importance are being satisfied by a given alternative.
- *Ceteris paribus preferences:* “all other things being equal, I prefer these alternatives over those other ones”

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.

Pareto optimality is the most basic concept we have considered, but there are many others . . .

Collective Utility Functions

A CUF is a function $W : \mathbb{R}^n \rightarrow \mathbb{R}$ mapping utility vectors to the reals. Here we define them over allocations A (inducing utility vectors):

- The *utilitarian* social welfare is defined as the sum of utilities:

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

- The *egalitarian* social welfare is given by the utility of the agent that is currently worst off:

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

- The *Nash product* is defined as the product of individual utilities:

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

Collective Utility Functions (cont.)

- The *elitist* social welfare is given by the utility of the agent that is currently best off:

$$sw_{el}(P) \quad = \quad \max\{u_i(P) \mid i \in \mathcal{A}gents\}$$

- Let $\vec{u}_A$ be the *ordered utility vector* induced by allocation A. Then the *k-rank dictator* CUF sw_k is defined as follows:

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

  Recall that sw_k is the same as the egalitarian CUF for $k = 1$ and the same as the elitist CUF for $k = n$ (number of agents).

The Leximin-Ordering

The *leximin-ordering* $\preceq_\ell$ is a social welfare ordering that may be regarded as a refinement of the egalitarian CUF:

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

Generalisations

Consider the following *family of CUFs*, parametrised by $p \neq 0$:

$$sw_{(p)}(A) \quad = \sum_{i \in \mathcal{A}gents} g_{(p)}(u_i(A)) \text{ where } g_{(p)}(x) = \begin{cases} x^p & \text{if } p > 0 \\ -x^p & \text{if } p < 0 \\ \log x & \text{if } p = 0 \end{cases}$$

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 $-\infty$. To see this, consider for instance:

$$\begin{array}{rcl} \text{We want: } \langle 2,2,100,100\rangle & \prec_\ell & \langle 2,3,3,3\rangle \\ -(2^p + 2^p + 100^p + 100^p) & < & -(2^p + 3^p + 3^p + 3^p) \end{array}$$

Ordered Weighted Averaging

Another family of of CUFs are *ordered weighted averaging operators*.

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

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

Again, 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 α goes to 0.

Normalised Utility

It can often be necessary 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 the maximum individual utility for each agent with respect to the 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:

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

  Observe that this entails that maximum utility if 1 for each agent.

  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:

$$u_i(A(i)) \quad \geq \quad u_i(A(j))$$

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

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

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

  Advantages: simple protocols; known results on mechanism design; experience with algorithms

- 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).

  Advantages: no need to trust a centre; division of labour; more natural for many applications; serious test for the MAS paradigm

Auction Protocols

Distinguish *direct* auctions (where the auctioneer acts as a seller) and *reverse* auctions (where the auctioneer acts as a buyer).

Concerning *bidding*, we can distinguish the following parameters:

- either *open-cry* or *sealed-bid*
- either *ascending* or *descending* or *one-shot*

Then we have to choose an *allocation rule*. This will usually be formulated as an optimisation problem over the bids received.

Finally we have to specify a *pricing rule*. For simple auctions, we have (at least) the following two options:

- either *first-price* or *second-price*

For combinatorial auctions, the latter has been generalised to the VCG mechanism with the Clarke tax.

Negotiation Protocols

Probably the best-known type of protocol for negotiation is the *Contract Net* and its extensions:

- Contract Net phases:
  announcement, *bidding*, *assignment*, *confirmation*
- Existing extensions to the basic model include:
  - selling *bundles* of resources
  - *barter* instead of monetary payments
  - *concurrent* contracting with the option to decommit
  - *levelled-commitment contracts*
- Developing a working negotiation protocol for fully distributed and multilateral negotiation is still an open issue . . .

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? ($\leadsto$ *mechanism design*)
- *Complexity results* . . .

Complexity Issues

Next we review some of these *complexity results* . . .

Let $\mathcal{A}$ be a finite set of agents; $\mathcal{R}$ a finite set of resources; and $\mathcal{U}$ a representation of the agents' utility functions.

As for all complexity results, the *representation* of the input problem is crucial. For instance, if the input problem is represented inefficiently (e.g. using exponential space when this is not required), then complexity results (which are expressed with respect to the size of the input problem) may seem much more favourable than they really are.

As our focus here is on demonstrating *what kind of questions* people have been asking rather than on the exact complexity results, we are not going to give much detail about this here. Most results apply to a variety of representation forms (such as k-additive utilities or straight-line programs).

Quantitative Criteria

The first decision problem concerns utilitarian social welfare:

Welfare Optimisation (WI)
Instance: $\langle \mathcal{A}, \mathcal{R}, \mathcal{U} \rangle$; $K \in \mathbb{Z}$
Question: Is there an allocation A such that $sw_u(A) > K$?

This is basically the same as the decision problem underlying the WDP in combinatorial auctions, which we have seen to be NP-complete.

The following closely related 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 $sw_u(A) < sw_u(A')$?

Qualitative Criteria

A decision problem is said to be in coNP iff its complementary problem 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?

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: Σ_2^p-complete (NP with NP oracle).

Path and Convergence Properties

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

Φ-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 Φ-deals leading from A to A'?

One of several known results is that Φ-Path is PSPACE-complete in case Φ is the predicate selecting all individually rational 1-deals. Recall that for modular utilities, the same problem is a trivial one (the answer is always "yes").

A related problem, *Φ-Convergence*, asks whether *any* given sequence of Φ-deals would result in a socially optimal allocation.

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)

Mechanism Design

Mechanism design is concerned with the *design of mechanisms* for collective decision making (including MARA) that favour particular outcomes despite agents pursuing their own interests.

- Main result: the *Vickrey-Clarke-Groves* (VCG) mechanism makes truth-telling a dominant strategy.
- But *manipulation* is possible: collusion and false-name bidding
- Standard theory applies to *centralised* systems.
- High *complexity:* computing prices involves solving many NP-hard optimisation problems

Algorithm Design

Algorithm design comes in at a variety of points. We have discussed *algorithms for winner determination* in combinatorial auctions in an earlier class:

- The WDP can be tackled using both off-the-shelf *mathematical programming* software and specialised *AI search techniques*.
- While it is 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) . . .

Examples of Application Areas

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

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

Summary

We have given an overview of the Multiagent Resource Allocation research area, the main topic of this course.

- 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
- There are many *applications* of MARA, such as industrial procurement, earth observation satellites, manufacturing systems, and grid computing.

References

This lecture has been based on the following survey paper:

- Y. Chevaleyre *et al.* Issues in Multiagent Resource Allocation. *Informatica*, 30:3–31, 2006.

The 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.

Mechanism design and *algorithm design* are not covered by the survey; see slides of earlier lectures for references.