Reduction of Economic Inequality in Combinatorial Domains

Ulle Endriss
ILLC, University of Amsterdam

Abstract: Criteria for measuring economic inequality, such as the Lorenz curve and the Gini index, are widely used in the social sciences but have hardly been explored in Multiagent Systems, even though the significance of other concepts from fair division is widely accepted in the field. In a departure from the standard model used in Economics, we apply inequality criteria to allocation problems with indivisible goods, i.e., to the kind of problem typically analysed in Multiagent Systems. This gives rise to the combinatorial optimisation problem of computing an allocation that reduces inequality with respect to an initial allocation (and the closely related problem of minimising inequality), for a chosen inequality measure. We define this problem, we discuss the computational complexity of various aspects of it, and we formulate a generic approach to designing modular algorithms for solving it using integer programming.

The Model

We work in the standard model of multiagent resource allocation.

Finite sets of *agents* $\mathcal{N} = \{1, \ldots, n\}$ and of indivisible *goods* $\mathcal{G}$. Each good needs to be allocated to exactly one agent.

Any given *allocation* A induces a *utility* of $u_i(A)$ for agent $i \in \mathcal{N}$. So any allocation A induces a *utility vector* $(u_1(A), \ldots, u_n(A))$.

The Problem

We want *fair* allocations. How can we measure the *inequality* inherent in a given utility vector (induced by a given allocation)?

Once we have settled on a definition, how can we *compute* good allocations in view of that definition in practice?

And how *complex* is this task in theory?

Measuring Inequality

Which is more equal, (1,2,7,7,8) or (1,3,5,6,10)?

The Pigou-Dalton Principle

A move from allocation A to A' is called a *Pigou-Dalton transfer* if there are two agents $i, j \in \mathcal{N}$ such that:

- Only the bundles held by i and j change.
- Inequality reduces: $|u_i(A) - u_j(A)| > |u_i(A') - u_j(A')|$
- Total utility does not reduce: $u_i(A) + u_j(A) \leq u_i(A') + u_j(A')$

By the *Pigou-Dalton Principle*, any measure of fairness should consider Pigou-Dalton transfers (weak) improvements.

The Lorenz Curve

Ideally, all agents enjoy the same utility. The Lorenz curve is a way to visualise how far we are from this ideal.

Let $\boldsymbol{u}^*(A)$ be the *ordered utility vector* of A. Then $L_k(A) = \sum_{i=1}^{k} u_i^*(A)$ is the total utility of the k poorest agents.

The vector $(L_1(A), \ldots, L_n(A))$ is called the *Lorenz curve* of A.

Inequality Indices

An *inequality index* is a quantitative measure of inequality of an allocation (usually $\in$ [0,1]). Two popular indices:

- *Gini index* = *area* between line of perfect equality and Lorenz curve (divided by a suitable normalisation factor)
- *Robin Hood index* = maximal *distance* between line of perfect equality and Lorenz curve (also normalised)

Representation Languages

Before we can design an algorithm to reason about a multiagent resource allocation problem and before we can study its complexity, we need to fix a *language* for representing *utility functions*.

Three natural choices (familiar from combinatorial auctions):

- *OR-language:* compact, reasonably expressive, highly intractable
- *XOR-language:* not compact, fully expressive
- *weighted goals:* compact, fully expressive (for certain goals)

Interesting special case: *additive* utility functions (weighted atoms).

Complexity Results

Checking whether a *Pigou-Dalton* transfer is possible is

- *polynomial* for the *XOR-language*;
- *NP-hard* for the *OR-language* [NP-membership is open];
- *not polynomial* unless NP = coNP for *additive* utilities.

Checking whether a *Lorenz improvement* is possible is

- *NP-complete* for the *XOR-language*;
- *NP-complete* for the *OR-language*;
- *NP-complete* for many *weighted goal* languages.

Also in the paper: algorithms for computing Lorenz improvements and minimising inequality indices (using integer programming)