Preferences for Fair Division

Jérôme Lang
LAMSADE, CNRS / Université Paris-Dauphine

COST Summer School on Fair Division

Grenoble, July 13–17, 2015

Plagiarizing:

- Sylvain Bouveret and Jérôme Lang, Tutorial on Graphical Preference Representation Languages, IJCAI-11.
- Jérôme Lang and Jörg Rothe, Fair Division of Indivisible Goods, Chapter 8 of Jörg Rothe (ed), Economics and Computation, Springer.

Outline

Some resource allocation problems

- Problem 1: *Allocating time slots to speakers*
  - Ulle has a slight preference for teaching on mornings, but above all prefers to have consecutive slots, that is, he prefers (14–15 and 15–16) to (9–10 and 11–12).
  - Ioannis has a preference for not teaching in the morning, and prefers to have his slots on two different days.
  - Christian has a preference for not teaching on Monday, and wants all his slots in the same day.
  - Jérôme's course should come before Ulle's and Christian's talks.

  Once the agents have reported their preferences, the allocation decision will be made centrally, by the COST Fair Division Summer School Central Organization (FDSSCO).

Some resource allocation problems

- Problem 2: *Divorcing*
  - *George and Helena*
    George and Helena are engaged in a divorce settlement process.
    They remain good friends and their divorce is not conflictual; therefore, they decide to do without a lawyer, and decide by themselves that Helena gets the books and George the bookshelves.
  - *John and Katia*
    John and Katia are unable to negotiate alone, and need to involve a lawyer, who helps them deciding that the children's custody will be shared equally between them, and that, in addition, Katia gets the house, while John gets the cat plus some monetary compensation from Katia.

Some resource allocation problems

- Problem 3: *Earth observation satellites*
    - France and Germany have jointly bought a very expensive Earth observation satellite. Every day, each country's responsible committee expresses its preferences over the photos it wants to be made.
    - There are some physical constraints on the satellite that restrict the set of photos that can be made on a single day, which needs a process to decide in a fair way which photos will be made.
    - This may be complicated by the fact that France paid for two thirds of the satellite while Germany paid only for one third, which leads to different entitlements on the number of photos.

Some resource allocation problems

- Problem 4: *Sport team formation*

  Two schoolchildren, Anna and David, have to form two sport teams. Resources are players. Anna chooses first one member of her team, then David one, then again Anna, then David, etc.

Some resource allocation problems

- Problem 5: *House allocation*
  - Version 1: n houses have to be allocated to n agents (exactly one each!); each agent expresses a preference ranking over all houses.
  - Version 2: n agents $a_1, \ldots, a_n$ initially live in house $h_1, \ldots, h_n$ respectively; each agent expresses a preference ranking over all houses; can we reallocate the houses so that some agent become happier but no agent becomes less happy?

Some resource allocation problems

- Problem 6: *Combinatorial auction*

  $\mathcal{O} = \{o_1, \ldots, o_p\}$ set of objects

  for each agent i, $V_i : 2^{\mathcal{O}} \to \mathbb{N}$

  $V_i(X) =$ maximum price that i is ready to pay for the set of objects X.

  if V_i additive for all i: then sell each object to its highest bidder

  but V_i is generally *non-additive* :

  - {left shoe}: 10 €; {right shoe}: 10 €; {left shoe, right shoe}: 50 €
  - {lemonade}: 2 €; {beer}: 3 €; {lemonade, beer}: 4 €

  optimal allocation π^*: maximizes the seller's revenue $\sum_{i=1}^{n} V_i(\pi(i))$
  where $\pi(i)$ is the set of objects allocated to agent i

Resource allocation problem (informal)

- a set of *resources* to be allocated
- a set of *agents*
- agents have *preferences* over resources
- the final *allocation* is subject to some *feasibility constraints*

... a final allocation is found somehow

Resource allocation problem (informal)

- a set of *resources* to be allocated
- a set of *agents*
- agents have *preferences* over resources
- the final *allocation* is subject to some *feasibility constraints*

... a final allocation is found somehow

Without additional parameters being fixed it is difficult to give a more precise definition.

1. Centralized versus decentralized

- Finding the allocation requires the agents to express, in one way or another, their preferences.
- The process that consists in querying the agents about their preferences is called **preference elicitation**.

Centralized mechanism There is a central authority that elicits the agents' preferences, and then determines the output allocation.

Decentralized / distributed mechanism There is no central authority, and the agents themselves compute the allocation, revealing their preferences by certain specific (inter)actions.

2. Divisible versus indivisible resources

Divisible resources

- homogeneous
- heterogeneous

Indivisible resources

- coming in single units
- coming in multiple units

3. Ordinal versus cardinal preferences

Cardinal preferences Agents associate **numerical values** with (sets of) resources

Ordinal preferences Agents are only allowed to **rank** with (sets of) resources

4. One-to-one versus many-to-one

One-to-one allocation
: Each agent gets exactly one resource: *matching problem*

Many-to-one allocation
: Each agent gets possibly several resources (*bundles*)

5. Money and initial endowments

Money or no money Is there any money involved in the mechanism? Do the agents pay and/or receive money?

Initial endowments Do the agents initially own resources?

6. Shareable versus nonshareable

Non-shareable resources Each resource is allocated to a single agent, who is the only one who can enjoy it.

Shareable resources Resources can be allocated to several (or even all) agents.

7. Fairness versus efficiency

Fairness What counts above all is to be fair and equitable to the agents: **fair division**

Efficiency What counts is the global efficiency of the outcome (for instance, monetary revenue)

Often: a mix of both.

Centralized fair division

Given

- a set of *resources* to be allocated
- a set of *agents*
- *preferences* of agents over resources
- the final *allocation* being subject to some *feasibility constraints*
- *fairness (and efficiency) criteria* for evaluating the quality of allocation

... determine a fair allocation of resources to agents

Decentralized fair division

Given

- a set of *resources* to be allocated
- a set of *agents*
- some prior knowledge about agents' *preferences* over resources
- the final *allocation* being subject to some *feasibility constraints*
- *fairness criteria* for evaluating the quality of allocation

... find an interaction protocol between agents guaranteeing that the outcome will have certain level of fairness.

Outline

Admissible bundles

From now on we focus on indivisible goods.

- $O = \{o_1, \ldots, o_m\}$ indivisible objects
- 2^O set of all *bundles of objects*
- $\mathcal{X} \subseteq 2^O$ set of admissible bundles that an agent may receive

Examples of admissible bundles:

- cardinality constraint: each agent receives exactly k objects:

$$\mathcal{X} = \{S \subseteq O, |S| \leq k\}$$

- categorized items (Mackin and Xia, 15): objects are clustered in *categories* and each agent receives exactly one item from each category:

$$\mathcal{X} = D_1 \times \ldots \times D_p$$

  where D_i is the set of all objects of category i.

  Example: one first dish + one main dish + one drink per agent

Preferences over bundles

- N sets of agents
- $O = \{o_1, \ldots, o_m\}$ indivisible objects

Notation: $[o_1 o_2 | o_3 | o_4 o_5]$ is the allocation where that agent 1 receives $\{o_1 o_2\}$, 2 receives $\{o_3\}$, 3 receives $\{o_4, o_5\}$.

"No externality" assumption:

an agent's preferences bear only on the bundle she receives

- 1 is indifferent between $[o_1 o_2 | o_3 | o_4 o_5]$ and $[o_1 o_2 | o_3 o_5 | o_4]$
- 2 is indifferent between $[o_1 o_2 | o_3 | o_4 o_5]$ and $[\emptyset | o_3 | o_1 o_2 o_4 o_5]$
- etc.

Therefore: it is sufficient to know each agent's preferences over *bundles* (as opposed to her preferences over all allocations).

Preference structures

Specifying preferences on $\mathcal{X}$: **comparing, ranking, evaluating** bundles.

Preference structures

Ordinal preferences

Preference relation on $\mathcal{X}$: **reflexive and transitive relation** $\succeq$

$x \succeq y$		x is at least as good as y
$x \succ y$	$\Leftrightarrow$	$x \succeq y$ and not $y \succeq x$ x is preferred to y (**strict preference**)
$x \sim y$	$\Leftrightarrow$	$x \succeq y$ and $y \succeq x$ x and y are equally preferred (**indifference**)
x Q y	$\Leftrightarrow$	neither $x \succeq y$ nor $y \succeq x$ x and y are (**incomparable**)

$\succeq$ is often assumed to be complete (no incomparabilities)

More sophisticated models: interval orders, semi-orders etc.

Preference structures

Cardinal preferences

- Utility function $u : \mathcal{X} \to \mathbb{R}$
- More generally $u : \mathcal{X} \to V$ ordered scale; example: $V = \{unacceptable, bad, medium, good, excellent\}$

From cardinal preferences to ordinal preferences:

$$x \succeq_u y \Leftrightarrow u(x) \geq u(y)$$

Preference structures

Dichotomous preferences

- $A \subseteq \mathcal{X}$ set of acceptable bundles
- dichotomous preferences are cardinal preferences:

$$V = \{0,1\}; \;\; u(S) = 1 \Leftrightarrow S \in A.$$

- dichotomous preferences are ordinal preferences:

$$S \succeq S' \Leftrightarrow (S \in A) \text{ or } (S' \notin A).$$

Preference structures

Fuzzy preferences

- $\mu_R : \mathcal{X}^2 \rightarrow [0,1]$
- $\mu_R(x,y)$ degree to which x is preferred to y.
- more general than both cardinal and ordinal preferences

Preference structures

$$
\begin{tikzcd}
 & \text{fuzzy preferences} \arrow[dl] \arrow[dr] & \\
\text{cardinal preferences} \arrow[dr] & & \text{ordinal preferences} \arrow[dl] \\
 & \text{dichotomous preferences} &
\end{tikzcd}
$$

Monotonicity

- $O = \{o_1, \ldots, o_m\}$ indivisible objects
- 2^O set of all *bundles of objects*
- $\mathcal{X} \subseteq 2^O$ set of admissible bundles that an agent may receive

Typically, preferences over bundles are *monotonic*: receiving one more good never makes an agent less happy.

- ordinal preferences: if $S \supseteq S'$ then $S \succeq S'$
- cardinal preferences: if $S \supseteq S'$ then $u(S) \geq u(S')$

Strict monotonicity:

- ordinal preferences: if $S \supset S'$ then $S \succ S'$
- cardinal preferences: if $S \supset S'$ then $u(S) > u(S')$

Preferential dependencies

Existence of *preferential dependencies* between variables:

- I'd like to have two consecutive time slots for my lectures (but not three)
- if I don't get the shared custody of the children then at least I'd like to keep the cat
- I want Ann or Charles or Daphne in my team, each of whom would be an excellent goal keeper
- if I receive the left shoe then I'm ready to pay more for the right shoe

An incursion into multi-attribute utility theory

- $N = \{1, 2, \ldots, n\}$ set of attributes
- D_i: set of values for the ith attribute
- $\mathcal{X} = D_1 \times \ldots \times D_n$ set of all conceivable alternatives. Here:
  - in general, $\mathcal{X} = 2^O$: attribute X_i is object o_i, binary domains $\{in, out\}$
  - (in categorized domains) attributes are categories.

- $J \subseteq N$ subset of attributes
- $D_J = \Pi_{j \in J} D_j$, $D_{-J} = \Pi_{j \notin J} D_j$,
- $(x_J, y_{-J}) \in \mathcal{X}$: contains x_j for each $i \in J$ and y_i for each $i \notin J$
- $(x_i, y_{-i}) \in \mathcal{X}$: identical to y except for the value of attribute i.

Example:

- $\mathcal{X} = 2^{\{o_1, o_2, o_3, o_4, o_5\}}$
- $x = (in, out, out, in, in) = \{o_1, o_4, o_5\}$;
- $y = (out, in, in, in, out) = \{o_2, o_3, o_4\}$;
- $(x_1, y_{-1}) = (in, in, in, in, out) = \{o_1, o_2, o_3, o_4\}$

An incursion into multi-attribute utility theory

The simplest model: *representing preferences via additively decomposable utilities*

(a) for all $x, y \in \mathcal{X}$, $x \succeq y \Leftrightarrow u(x) \geq u(y)$

(b) for all $x = (x_1, \dots, x_n) \in \mathcal{X}$, $u(x) = \sum_{i=1}^{n} u_i(x_i) \geq \sum_{i=1}^{n} u_i(y_i)$

- $x = (x_1, \dots, x_n), y = (y_1, \dots, y_n)$: alternatives
- x_i value of x on attribute i
- $u_i(x_i)$ marginal utility value of x on attribute i

When does an agent have an additively decomposable utility function?

Additive decompositions

Start with two attributes: $\mathcal{X} = D_1 \times D_2$

An agent's preference relation on $\mathcal{X}$ is representable by an additively decomposable utility function iff

$$\text{for all } x, y \in \mathcal{X},\ x \succeq y \Leftrightarrow u_1(x_1) + u_2(x_2) \geq u_1(y_1) + u_2(y_2)$$

where $u_1 : D_1 \to \mathbb{R}$; $u_2 : D_2 \to \mathbb{R}$

A first necessary condition (Debreu, 1954): $\succeq$ must be a weak order, *i.e.*, a relation satisfying

- completeness: for all $x, y \in \mathcal{X}$, either $x \succeq y$ or $y \succeq x$.
- transitivity: for all $x, y \in \mathcal{X}$, $x \succeq y$ and $y \succeq z$ implies $x \succeq z$.

From now on we assume that $\succeq$ is a weak order.

Additive decompositions: 2-dimensional spaces

Assume there exists u representing $\succeq$. Then for every $x_1, y_1 \in D_1$ and $x_2, y_2 \in D_2$,

$$\begin{aligned}
(x_1, x_2) \succeq (y_1, x_2) \quad & \Leftrightarrow u_1(x_1) + u_2(x_2) \geq u_1(y_1) + u_2(x_2) \\
& \Leftrightarrow u_1(x_1) \geq u_1(y_1) \\
& \Leftrightarrow u_1(x_1) + u_2(y_2) \geq u_1(y_1) + u_2(y_2) \\
& \Leftrightarrow (x_1, y_2) \succeq (y_1, y_2)
\end{aligned}$$

This property expresses some independence between the attributes: the decision maker takes into account the attributes separately.

Preferential independence for two attributes

Preferential independence (Keeney & Raiffa, 76):

Attribute 1 is preferentially independent from attribute 2 (w.r.t. $\succeq$) if for all $x_1, y_1 \in D_1$ and $x_2, y_2 \in D_2$,

$$(x_1, x_2) \succeq (y_1, x_2) \Leftrightarrow (x_1, y_2) \succeq (y_1, y_2)$$

The preferences over the possible values of D_1 are independent from the value of D_2

Example

Two binary attributes A, B with domains $\{a, \bar{a}\}$, $\{b, \bar{b}\}$

Preference relation: $ab \succ a\bar{b} \succ \bar{a}\bar{b} \succ \bar{a}b$

Preferential independence for two attributes

Preferential independence (Keeney & Raiffa, 76):

Attribute 1 is preferentially independent from attribute 2 (w.r.t. $\succeq$) if for all $x_1, y_1 \in D_1$ and $x_2, y_2 \in D_2$,

$$(x_1, x_2) \succeq (y_1, x_2) \Leftrightarrow (x_1, y_2) \succeq (y_1, y_2)$$

The preferences over the possible values of D_1 are independent from the value of D_2

Example

Two binary attributes A, B with domains $\{a, \bar{a}\}$, $\{b, \bar{b}\}$
Preference relation: $ab \succ a\bar{b} \succ \bar{a}\bar{b} \succ \bar{a}b$
A preferentially independent from B

Preferential independence for two attributes

Preferential independence (Keeney & Raiffa, 76):

Attribute 1 is preferentially independent from attribute 2 (w.r.t. $\succeq$) if for all $x_1, y_1 \in D_1$ and $x_2, y_2 \in D_2$,

$$(x_1, x_2) \succeq (y_1, x_2) \Leftrightarrow (x_1, y_2) \succeq (y_1, y_2)$$

The preferences over the possible values of D_1 are independent from the value of D_2

Example

Two binary attributes A, B with domains $\{a, \bar{a}\}$, $\{b, \bar{b}\}$
Preference relation: $ab \succ a\bar{b} \succ \bar{a}\bar{b} \succ \bar{a}b$
A preferentially independent from B
B preferentially dependent on A

Separability for two attributes

Separability A preference relation $\succeq$ on $\mathcal{X} = D_1 \times D_2$ is *separable* if 1 is independent from 2 and 2 is independent from 1 w.r.t. $\succeq$.

- $ab \succ a\bar{b} \succ \bar{a}\bar{b} \succ \bar{a}b$ not separable
- $ab \succ a\bar{b} \succ \bar{a}b \succ \bar{a}\bar{b}$ separable

Preferential independence for n attributes

N set of attributes; $\{U, V, W\}$ partition of N.

$D_U = \times_{i \in U} D_i$ etc.

Conditional preferential independence (Keeney & Raiffa, 76)

U is preferentially independent from V (given W) iff

for all $u, u' \in D_U$, $v, v' \in D_V$, $w, w' \in D_W$,

$(u, v, w) \succeq (u', v, w)$ iff $(u, v', w) \succeq (u', v', w)$

given any fixed value w of W, the preferences over the possible values of U are independent from the value of V

$$abc \succ ab\bar{c} \succ a\bar{b}\bar{c} \succ a\bar{b}c \succ \bar{a}\bar{b}\bar{c} \succ \bar{a}\bar{b}c \succ \bar{a}bc \succ \bar{a}b\bar{c}$$

Preferential independence for n attributes

N set of attributes; $\{U, V, W\}$ partition of N.

$D_U = \times_{i \in U} D_i$ etc.

Conditional preferential independence (Keeney & Raiffa, 76)

U is preferentially independent from V (given W) iff

$$\text{for all } u, u' \in D_U,\ v, v' \in D_V,\ w, w' \in D_W,$$
$$(u, v, w) \succeq (u', v, w) \text{ iff } (u, v', w) \succeq (u', v', w)$$

given any fixed value w of W, the preferences over the possible values of U are independent from the value of V

$$abc \succ ab\bar{c} \succ a\bar{b}\bar{c} \succ a\bar{b}c \succ \bar{a}\bar{b}\bar{c} \succ \bar{a}\bar{b}c \succ \bar{a}bc \succ \bar{a}b\bar{c}$$

- a independent from $\{b, c\}$?

Preferential independence for n attributes

N set of attributes; $\{U, V, W\}$ partition of N.

$D_U = \times_{i \in U} D_i$ etc.

Conditional preferential independence (Keeney & Raiffa, 76)

U is preferentially independent from V (given W) iff

$$\text{for all } u, u' \in D_U,\ v, v' \in D_V,\ w, w' \in D_W,$$
$$(u, v, w) \succeq (u', v, w) \text{ iff } (u, v', w) \succeq (u', v', w)$$

given any fixed value w of W, the preferences over the possible values of U are independent from the value of V

$$abc \succ ab\bar{c} \succ a\bar{b}\bar{c} \succ a\bar{b}c \succ \bar{a}\bar{b}\bar{c} \succ \bar{a}\bar{b}c \succ \bar{a}bc \succ \bar{a}b\bar{c}$$

- a independent from $\{b, c\}$? yes
- $\{b, c\}$ independent from a?

Preferential independence for n attributes

N set of attributes; $\{U, V, W\}$ partition of N.

$D_U = \times_{i \in U} D_i$ etc.

Conditional preferential independence (Keeney & Raiffa, 76)

U is preferentially independent from V (given W) iff

$$\text{for all } u, u' \in D_U,\ v, v' \in D_V,\ w, w' \in D_W,$$
$$(u, v, w) \succeq (u', v, w) \text{ iff } (u, v', w) \succeq (u', v', w)$$

given any fixed value w of W, the preferences over the possible values of U are independent from the value of V

$$abc \succ ab\bar{c} \succ a\bar{b}\bar{c} \succ a\bar{b}c \succ \bar{a}\bar{b}\bar{c} \succ \bar{a}\bar{b}c \succ \bar{a}bc \succ \bar{a}b\bar{c}$$

- a independent from $\{b, c\}$? yes
- $\{b, c\}$ independent from a? no
- b independent from $\{a, c\}$?

Preferential independence for n attributes

N set of attributes; $\{U, V, W\}$ partition of N.

$D_U = \times_{i \in U} D_i$ etc.

Conditional preferential independence (Keeney & Raiffa, 76)

U is preferentially independent from V (given W) iff

$$\text{for all } u, u' \in D_U,\ v, v' \in D_V,\ w, w' \in D_W,$$
$$(u, v, w) \succeq (u', v, w) \text{ iff } (u, v', w) \succeq (u', v', w)$$

given any fixed value w of W, the preferences over the possible values of U are independent from the value of V

$$abc \succ ab\bar{c} \succ a\bar{b}\bar{c} \succ a\bar{b}c \succ \bar{a}\bar{b}\bar{c} \succ \bar{a}\bar{b}c \succ \bar{a}bc \succ \bar{a}b\bar{c}$$

- a independent from $\{b, c\}$? yes
- $\{b, c\}$ independent from a? no
- b independent from $\{a, c\}$? no
- b independent from a given c?

Preferential independence for n attributes

N set of attributes; $\{U, V, W\}$ partition of N.

$D_U = \times_{i \in U} D_i$ etc.

Conditional preferential independence (Keeney & Raiffa, 76)

U is preferentially independent from V (given W) iff

$$\text{for all } u, u' \in D_U,\ v, v' \in D_V,\ w, w' \in D_W,$$
$$(u, v, w) \succeq (u', v, w) \text{ iff } (u, v', w) \succeq (u', v', w)$$

given any fixed value w of W, the preferences over the possible values of U are independent from the value of V

$$abc \succ ab\bar{c} \succ a\bar{b}\bar{c} \succ a\bar{b}c \succ \bar{a}\bar{b}\bar{c} \succ \bar{a}\bar{b}c \succ \bar{a}bc \succ \bar{a}b\bar{c}$$

- a independent from $\{b, c\}$? yes
- $\{b, c\}$ independent from a? no
- b independent from $\{a, c\}$? no
- b independent from a given c? no
- b independent from c given a?

Preferential independence for n attributes

N set of attributes; $\{U, V, W\}$ partition of N.

$D_U = \times_{i \in U} D_i$ etc.

Conditional preferential independence (Keeney & Raiffa, 76)

U is preferentially independent from V (given W) iff

for all $u, u' \in D_U$, $v, v' \in D_V$, $w, w' \in D_W$,

$$(u, v, w) \succeq (u', v, w) \text{ iff } (u, v', w) \succeq (u', v', w)$$

given any fixed value w of W, the preferences over the possible values of U are independent from the value of V

$$abc \succ ab\bar{c} \succ a\bar{b}\bar{c} \succ a\bar{b}c \succ \bar{a}\bar{b}\bar{c} \succ \bar{a}\bar{b}c \succ \bar{a}bc \succ \bar{a}b\bar{c}$$

- a independent from $\{b, c\}$? yes
- $\{b, c\}$ independent from a? no
- b independent from $\{a, c\}$? no
- b independent from a given c? no
- b independent from c given a? yes

Separability and weak separability

- $U \subseteq N$ is *independent* for $\succeq$ if U is preferentially independent from $N \setminus U$
- $\succeq$ is *separable* if for every $U \subseteq N$, U is independent for $\succeq$
- $\succeq$ is *weakly separable* if for every $i \in N$, $\{i\}$ is independent for $\succeq$

(Remark: both notions coincide for $n = 2$)

- $abc \succ ab\bar{c} \succ a\bar{b}c \succ \bar{a}bc \succ \bar{a}\bar{b}c \succ a\bar{b}\bar{c} \succ \bar{a}\bar{b}\bar{c} \succ \bar{a}b\bar{c}$

Separability and weak separability

- $U \subseteq N$ is *independent* for $\succeq$ if U is preferentially independent from $N \setminus U$
- $\succeq$ is *separable* if for every $U \subseteq N$, U is independent for $\succeq$
- $\succeq$ is *weakly separable* if for every $i \in N$, $\{i\}$ is independent for $\succeq$

(Remark: both notions coincide for $n = 2$)

- $abc \succ ab\bar{c} \succ a\bar{b}c \succ \bar{a}bc \succ \bar{a}\bar{b}c \succ a\bar{b}\bar{c} \succ \bar{a}\bar{b}\bar{c} \succ \bar{a}b\bar{c}$
  - $\succeq$ is not weakly separable (b not independent from c given a)
- $abc \succ ab\bar{c} \succ a\bar{b}c \succ \bar{a}bc \succ \bar{a}\bar{b}c \succ a\bar{b}\bar{c} \succ \bar{a}b\bar{c} \succ \bar{a}\bar{b}\bar{c}$

Separability and weak separability

- $U \subseteq N$ is *independent* for $\succeq$ if U is preferentially independent from $N \setminus U$
- $\succeq$ is *separable* if for every $U \subseteq N$, U is independent for $\succeq$
- $\succeq$ is *weakly separable* if for every $i \in N$, $\{i\}$ is independent for $\succeq$

(Remark: both notions coincide for $n = 2$)

- $abc \succ ab\bar{c} \succ a\bar{b}c \succ \bar{a}bc \succ \bar{a}\bar{b}c \succ a\bar{b}\bar{c} \succ \bar{a}\bar{b}\bar{c} \succ \bar{a}b\bar{c}$
  - $\succeq$ is not weakly separable (b not independent from c given a)
- $abc \succ ab\bar{c} \succ a\bar{b}c \succ \bar{a}bc \succ \bar{a}\bar{b}c \succ a\bar{b}\bar{c} \succ \bar{a}b\bar{c} \succ \bar{a}\bar{b}\bar{c}$
  - $\succeq$ is weakly separable

Separability and weak separability

- $U \subseteq N$ is *independent* for $\succeq$ if U is preferentially independent from $N \setminus U$
- $\succeq$ is *separable* if for every $U \subseteq N$, U is independent for $\succeq$
- $\succeq$ is *weakly separable* if for every $i \in N$, $\{i\}$ is independent for $\succeq$

(Remark: both notions coincide for $n = 2$)

- $abc \succ ab\bar{c} \succ a\bar{b}c \succ \bar{a}bc \succ \bar{a}\bar{b}c \succ a\bar{b}\bar{c} \succ \bar{a}\bar{b}\bar{c} \succ \bar{a}b\bar{c}$
  - $\succeq$ is not weakly separable (b not independent from c given a)
- $abc \succ ab\bar{c} \succ a\bar{b}c \succ \bar{a}bc \succ \bar{a}\bar{b}c \succ a\bar{b}\bar{c} \succ \bar{a}b\bar{c} \succ \bar{a}\bar{b}\bar{c}$
  - $\succeq$ is weakly separable
  - $\succeq$ is not strongly separable

Additive decompositions

Question: is a strongly separable weak order $\succeq$ always representable by an additively decomposable utility function?

- $\mathcal{X} = D_1 \times D_2$ with $D_1 = \{a, b, c\}$ and $D_2 = \{d, e, f\}$
- $ad \succ bd \succ ae \succ af \succ be \succ cd \succ ce \succ bf \succ cf$
- $\succeq$ separable
- however $\succeq$ cannot be represented bu $u = u_1 + u_2$

$$
\begin{array}{ll}
(1) & af \succ be \Rightarrow u_1(a) + u_2(f) > u_1(b) + u_2(e) \\
(2) & be \succ cd \Rightarrow u_1(b) + u_2(e) > u_1(c) + u_2(d) \\
(3) & ce \succ bf \Rightarrow u_1(c) + u_2(e) > u_1(b) + u_2(f) \\
(4) & bd \succ ae \Rightarrow u_1(b) + u_2(d) > u_1(a) + u_2(e) \\
\hline
(1)+(2) & u_1(a) + u_2(f) > u_1(c) + u_2(d) \\
(3)+(4) & u_1(c) + u_2(d) > u_1(a) + u_2(f)
\end{array}
$$

Additive independence

We need a stronger notion of independence.

$N = \{1, \dots, n\}$ attributes
Von Neumann - Morgenstern lottery over $\mathcal{X}$:

$$[(p, x); (1-p, x')]$$

where $x, x' \in \mathcal{X}$

Additive independence

$\succ$ satisfies *additive independence* if for every pair of lotteries L, L' over $\mathcal{X}$ such that for every attribute i, L and L' have the same marginal probabilities over D_i, we have $L \sim L'$.

Additive independence

$n = 2$; $\mathcal{X} = D_A \times D_B$.
Example : let $\succeq$ on the set of lotteries over $\mathcal{X}$ defined by $L \succeq L'$ if $\bar{u}(L) \geq \bar{u}(L')$ where u defined as follows:

$$\begin{array}{lll} u(a_0, b_0) = 10 & u(a_0, b_1) = 7 & u(a_0, b_2) = 5 \\ u(a_1, b_0) = 9 & u(a_1, b_1) = 6 & u(a_1, b_2) = 4 \\ u(a_2, b_0) = 5 & u(a_2, b_1) = 2 & u(a_2, b_2) = 0 \end{array}$$

- $[0.5, (a_1, b_1); 0.5, (a_0, b_0)] \sim [0.5, (a_1, b_0); 0.5, (a_0, b_1)]$
- $[0.5, (a_2, b_1); 0.5, (a_0, b_0)] \sim [0.5, (a_2, b_0); 0.5, (a_0, b_1)]$
- etc.

$\succeq$ satisfies additive independence.

Additive independence

$n = 2$; $\mathcal{X} = D_A \times D_B$. Let $\succeq$ satisfying additive independence.
For any $a, a' \in D_A, b, b' \in D_B$ we have

$$0.5u(a,b) + 0.5u(a',b') = 0.5u(a,b') + 0.5u(a',b)$$

therefore

- fix $a_0 \in D_A$, $b_0, b_1 \in D_B$;
- $u(a,b_0) - u(a,b_1) = u(a_0,b_0) - u(a_0,b_1) = C$
- $u(a,b_0) = u(a,b_1) + (u(a_0,b_0) - u(a_0,b_1)) = u(a,b_1) + C$

All marginal utility functions $u_A(.,b) : D_A \to \mathbb{R}$ are the same up to a translation.

- fix $u(a_0,b_0) = 0$.
- $u(a,b) = u(a,b_0) + u(a_0,b) = u_A(a) + u_B(b)$
- u is additively decomposable!

Additive independence

Characterization of additively decomposable utilities (Fishburn):

A weak order $\succeq$ satisfies additive independence if and only if there exists an additively decomposable utility function u such that for all lotteries L, L' over X, we have $L \succeq L'$ if and only if $\bar{u}(L) \geq \bar{u}(L')$

$\bar{u}(L)$ expected utility of L

Remark: this is a characterization theorem for preference relations *over lotteries*. Can we find a characterization theorem for preferences over alternatives?

Additive independence

A characterization when $\mathcal{X}$ is *finite*.
$\mathcal{X} = D_1 \times \ldots \times D_n$ where each D_i is a finite set.
Let m be an integer ≥ 2 and let $x^1, \ldots, x^m, y^1, \ldots, y^m \in \mathcal{X}$. We say that

$$(x^1, \ldots, x^m) E^m (y^1, \ldots, y^m)$$

if for all attributes $i \in N$, $(x_i^1, \ldots, x_i^m)$ is a permutation of $(y_i^1, \ldots, y_i^m)$.

Suppose that $(x^1, \ldots, x^m) E^m (y^1, \ldots, y^m)$; u is additively decomposable then

$$\sum_{j=1}^{m} \sum_{i=1}^{n} u_i(x_i^j) = \sum_{j=1}^{m} \sum_{i=1}^{n} u_i(y_i^j)$$

Therefore, if $x^j \succeq y^j$ for all $j = 1, \ldots, m-1$ then $x^m \preceq y^m$.

Condition C^m Let $m \geq 2$. C^m holds if for all $x^1, \ldots, x^m, y^1, \ldots, y^m \in \mathcal{X}$ such that $(x^1, \ldots, x^m) E^m (y^1, \ldots, y^m)$, we have

$$x^j \succeq y^j \text{ for all } j = 1, \ldots, m-1 \text{ implies } x^m \preceq y^m$$

Additive independence

Theorem (Fishburn)

Let $\succeq$ be a weak order on a finite set $\mathcal{X} = D_1 \times \ldots D_n$. There are real-valued functions u_i on D_i such that $u(x) = \sum_{i=1}^{n} u_i(x_i)$ for all $x \in \mathcal{X}$ if and only if $\succeq$ satisfies C^m for all m.

Remark: for a set $\mathcal{X}$ of given cardinality, only a finite number of values of m have to be checked.

Combinatorial spaces...

- $O = \{o_1, \ldots, o_m\}$ indivisible objects
- 2^O set of all *bundles of objects*
- $\mathcal{X} \subseteq 2^O$ set of admissible bundles that an agent may receive

Each agent has to express her preferences over $\mathcal{X}$:

- Sometimes, this is not a problem (for instance: one-to-one allocation)
- However, generally $\mathcal{X}$ has a heavy *combinatorial structure*

Combinatorial spaces...

The combinatorial trap...

Two objects...

$o_1 o_2 \succ o_2 \succ o_1 \succ \varnothing \rightarrow$ 4 subsets to compare

Combinatorial spaces...

The combinatorial trap...

Four objects...

$o_1o_2o_3o_4 \succ o_1o_2o_4 \succ o_1o_3o_4 \succ o_2o_3o_4 \succ o_1o_2o_3 \succ o_1o_3 \succ o_2o_4 \succ o_3o_4 \succ o_1o_4 \succ o_1 \succ o_2 \succ o_4 \succ o_3 \succ \varnothing \rightarrow$ 16 subsets

Combinatorial spaces...

The combinatorial trap...

Twenty binary variables...

$$o_8 o_5 \succ o_5 o_3 o_9 \succ o_8 \succ \varnothing \succ o_5 \succ o_8 o_5 o_3 o_9 \succ o_8 o_3 \succ o_5 o_9 \succ o_3 o_9 \succ$$
$$o_8 o_9 \succ o_8 o_3 o_9 \succ o_5 o_3 \succ o_9 \succ o_3 \succ o_8 o_5 o_9 \succ o_8 o_5 o_3 o_1 o_2 o_5 o_8 o_9 \succ$$
$$o_1 o_5 o_6 \succ o_7 \succ o_2 o_3 o_4 o_5 o_6 o_7 o_8 \succ o_1 o_2 o_3 o_4 o_5 \succ o_1 o_3 \succ o_2 \succ$$
$$o_1 o_3 o_7 o_9 \succ o_1 o_5 \succ o_1 o_7 o_8 o_9 \succ o_2 \succ o_4 \succ o_6 \succ o_1 o_7 \succ o_1 o_2 o_3 \succ$$
$$o_1 o_2 \succ o_2 o_5 o_4 \succ o_1 \succ o_2 \succ o_1 o_2 o_5 o_4 \succ o_1 o_5 \succ o_2 o_4 \succ o_5 o_4 \succ$$
$$o_1 o_4 \succ o_1 o_5 o_4 \succ o_2 o_5 \succ o_4 \succ o_5 \succ o_1 o_2 o_4 \succ o_1 o_2 o_5 \succ o_1 o_5 \succ$$
$$o_5 o_3 o_9 \succ o_1 \succ \varnothing \succ o_5 \succ o_1 o_5 o_3 o_9 \succ o_1 o_3 \succ o_5 o_9 \succ o_3 o_9 \succ o_1 o_9 \succ$$
$$o_1 o_3 o_9 \succ o_5 o_3 \succ o_9 \succ o_3 \succ o_1 o_5 o_9 \succ o_1 o_5 o_3 o_9 o_6 o_5 o_1 o_9 \succ o_9 o_5 o_6 \succ$$
$$o_7 \succ o_6 o_3 o_4 o_5 o_6 o_7 o_1 \succ o_9 o_6 o_3 o_4 o_5 \succ o_9 o_3 \succ o_6 \succ o_9 o_3 o_7 o_9 \succ$$
$$o_9 o_5 \succ o_9 o_7 o_1 o_9 \succ o_6 \succ o_4 \succ o_6 \succ o_9 o_7 \succ o_9 o_6 o_3 \succ o_9 o_6 \succ$$
$$o_6 o_5 o_4 \succ o_9 \succ o_6 \succ o_9 o_6 o_5 o_4 \succ o_9 o_5 \succ o_6 o_4 \succ o_5 o_4 \succ o_9 o_4 \succ$$

$\rightarrow$ 1048575 subsets $\rightarrow$ the expression takes more than 12 days.

The dilemma

- The expression of **preferential dependencies** is often necessary.
- **but** . . . Representing and eliciting $\succeq$ or u *in extenso* is unfeasible in practice.

Outline

Combinatorial spaces: the dilemma

n attributes, each with d possible values $\Rightarrow$ d^n alternatives
[In fair division: alternatives are bundles of objects]

Way 1 Assume preferential independence

- elicitation and optimization are made easier (e.g. using decomposable utilities)
- *but* weak expressivity (impossibility to express preferential dependencies).

Way 2 Allow the user to express any possible preference over the alternatives

- full expressivity
- *but* representing and eliciting $\succeq$ or u *in extenso* is unfeasible in practice.

Combinatorial spaces: the dilemma

n attributes, each with d possible values $\Rightarrow$ d^n alternatives
[In fair division: alternatives are bundles of objects]

Way 1 Assume preferential independence

- elicitation and optimization are made easier (e.g. using decomposable utilities)
- *but* weak expressivity (impossibility to express preferential dependencies).

Way 2 Allow the user to express any possible preference over the alternatives

- full expressivity
- *but* representing and eliciting $\succeq$ or u *in extenso* is unfeasible in practice.

$\Downarrow$

Half-way: *languages for compact preference representation*

Representation languages for fair division

- $O = \{o_1, \ldots, o_m\}$ set of objects
- $\mathcal{X} = 2^O$

Representation language : $\langle L, I_L \rangle$, where

- L language
- $I_L : \Phi \in L \mapsto$ preference relation $\succeq_\Phi$ or utility function u_Φ induced by Φ

Representation languages for fair division

Example 1: a language for dichotomous preferences:

- L_{prop}: set of all propositional formulas built from the propositional symbols $\{o_1, \ldots, o_n\}$
- $\varphi \in L \mapsto u_\Phi$ defined by $u(S) = 1$ if $S \models \varphi$, $= 0$ otherwise.

Representation languages for fair division

Example 1: a language for dichotomous preferences:

- L_{prop}: set of all propositional formulas built from the propositional symbols $\{o_1, \ldots, o_n\}$
- $\varphi \in L \mapsto u_\Phi$ defined by $u(S) = 1$ if $S \models \varphi$, $= 0$ otherwise.

Example

- $\mathcal{O} = \{$ , , , , , ,  $\}$.
- Goal:  $\wedge \Big(\big($  $\wedge$  $\big) \vee$  $\Big)$

Representation languages for fair division

- $O = \{o_1, \dots, o_m\}$ set of objects
- $\mathcal{X} = 2^O$

Representation language : $\langle L, I_L \rangle$, where

- L language
- $I_L : \Phi \in L \mapsto$ preference relation $\succeq_\Phi$ or utility function u_Φ induced by Φ

Example 2: (obvious) language for additive utility functions:

- L_{add}: set of all collections of real numbers

$$W = \{u_i, 1 \leq i \leq m\}$$

for all $S \subseteq O$, $u_W(S) = \sum_{i, o_i \in S} u_i$

Representation languages for fair division

- $O = \{o_1, \ldots, o_m\}$ set of objects
- $\mathcal{X} = 2^O$

Representation language : $\langle L, I_L \rangle$, where

- L language
- $I_L : \Phi \in L \mapsto$ preference relation $\succeq_\Phi$ or utility function u_Φ induced by Φ

Example 3: "explicit" representations

- for utility functions: L_{exp} = set of all collections of pairs

$$\{\langle S, u(S) \rangle | S \in \mathcal{X}\}$$

- for preference relations: L'_{exp} = list

$$S_1 \succ S_2 \succ S_3 \succ \ldots$$

representing a ranking over $\mathcal{X}$.

Representation languages

On which criteria can we evaluate the different languages?

- **Expressive power**: what is the set of all preference structures expressible in the language?

Representation languages

On which criteria can we evaluate the different languages?

- **Expressive power**: what is the set of all preference structures expressible in the language?
- **Succinctness**: (informally) $\langle L_1, I_{L_1} \rangle$ is at least as succinct as language $\langle L_2, I_{L_2} \rangle$ is any preference structure expressible in $\langle L_2, I_{L_2} \rangle$ can be expressed in $\langle L_1, I_{L_1} \rangle$ *without any exponential growth of size*.

Representation languages

On which criteria can we evaluate the different languages?

- **Expressive power**: what is the set of all preference structures expressible in the language?
- **Succinctness**: (informally) $\langle L_1, I_{L_1} \rangle$ is at least as succinct as language $\langle L_2, I_{L_2} \rangle$ is any preference structure expressible in $\langle L_2, I_{L_2} \rangle$ can be expressed in $\langle L_1, I_{L_1} \rangle$ *without any exponential growth of size*.
- **Computational complexity**: how hard is it to compare two alternatives or to find an optimal alternative when the preferences are represented in $\langle L, I_L \rangle$?

Representation languages

On which criteria can we evaluate the different languages?

- **Expressive power**: what is the set of all preference structures expressible in the language?
- **Succinctness**: (informally) $\langle L_1, I_{L_1} \rangle$ is at least as succinct as language $\langle L_2, I_{L_2} \rangle$ is any preference structure expressible in $\langle L_2, I_{L_2} \rangle$ can be expressed in $\langle L_1, I_{L_1} \rangle$ *without any exponential growth of size*.
- **Computational complexity**: how hard is it to compare two alternatives or to find an optimal alternative when the preferences are represented in $\langle L, I_L \rangle$?
- **Easiness of elicitation**
  Preference elicitation = interaction with a user, so as to acquire her preferences, encoded in a language $\langle L, I_L \rangle$.

  Is it easy to construct protocols for eliciting the agent's preferences in $\langle L, I_L \rangle$?

Expressive power

Representation language: $\langle L, I_L \rangle$

Expressive power of a language = set of all preference structures that can be expressed in the language $= I_L(L)$.

$$\langle L, I_L \rangle \text{ at least as expressive as } \langle L', I_{L'} \rangle \text{ iff } I_L(L) \supseteq I_{L'}(L').$$

Examples :

- expressive power of L_{add}: all additive utility functions over $\mathcal{X}$;
- expressive power of L_{exp}: all utility functions over $\mathcal{X}$.

$\langle L_{exp}, I_{L_{exp}} \rangle$ is more expressive than $\langle L_{add}, I_{L_{add}} \rangle$.

Succinctness

Relative notion:

$\langle L_1, I_{L_1} \rangle$ is at least as succinct as $\langle L_2, I_{L_2} \rangle$ if there exists $F : L_2 \to L_1$ and a polynomial function p such that for all $\Phi \in L_2$:

- $I_{L2}(\Phi) = I_{L1}(F(\Phi))$: Φ and $F(\Phi)$ induce the same preferences
- $|F(\Phi)| \leq p(|\Phi|)$: the translation is succinct

Example:

- $\langle L_{exp,add}, I_{exp,add} \rangle$ = explicit representation restricted to additive utility functions = set of all collections of pairs

$$U = \{\langle x, u(x) \rangle | x \in \mathcal{X}\}$$

  such that u is additively decomposable

- $\langle L_{add}, I_{L_{add}} \rangle$ is strictly more succinct than $L_{exp,add}$;
- but $\langle L_{exp}, I_{L_{exp}} \rangle$ and $\langle L_{add}, I_{L_{add}} \rangle$ are incomparable because $\langle L_{exp}, I_{L_{exp}} \rangle$ is more expressive than $\langle L_{add}, I_{L_{add}} \rangle$.

Computational complexity

What is the computational complexity of the following problems *when the preferences on $\mathcal{X}$ are represented in the language $\langle L, I_L \rangle$*:

Given an input Φ in the language $\langle L, I_L \rangle$, ...

- DOMINANCE: and $x, y \in X$, do we have $x \succeq_\Phi y$?
- OPTIMISATION: find the preferred alternative (or one of the preferred alternatives)

  (trivial for monotonic preferences)

- CONSTRAINED OPTIMISATION: and a subset C, possibly defined succinctly, find the preferred option (or one of the preferred options) $x \in C$.

Measuring hardness uses *computational complexity* notions.

Elicitation

Preference elicitation = interaction with a user, so as to acquire her preferences, encoded in a language L (or more generally, so as to acquire *enough information* about her preferences)

Construction of elicitation protocols for some families of languages:

- exploiting preferential independencies so as to reduce the amount of information to elicit and the cognitive effort spent in communication;
- trade-off expressivity *vs.* elicitation complexity.

Outline

Ranking single objects

- $O = \{o_1, \ldots, o_m\}$ set of objects
- $\mathcal{X} = 2^O$

- L_{sing}: set of all rankings over O
- for each ranking $\rhd$ over O, $I(\rhd) = \succ$ is the *monotonic and separable extension* of $\rhd$ to 2^O, that is, the smallest preference relation $\succ$ over 2^O such that
  - $\succ$ extends $\rhd$: for all $o_i, o_j \in O$, $o_i \rhd o_j$ implies $\{o_i\} \succ' \{o_j\}$
  - $\succ$ is separable
  - $\succ$ is monotonic
- $\succ$ sometimes called the *Bossong-Schweigert* extension, or the *responsive extension* of $\rhd$.

Ranking single objects

$m = 2,\ o_1 \rhd o_2$

$$\begin{tikzcd}
o_1 o_2 \arrow[d] \\
o_1 \arrow[d] \\
o_2 \arrow[d] \\
\varnothing
\end{tikzcd}$$

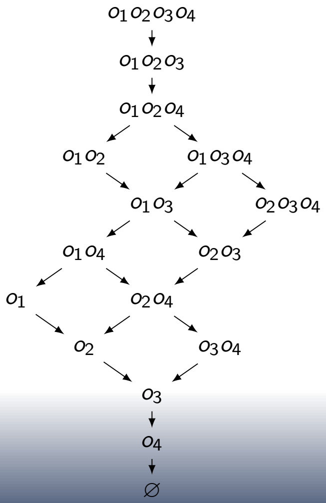
Ranking single objects

$m = 3,\ o_1 \rhd o_2 \rhd o_3$

$$\begin{tikzcd}
& o_1 o_2 o_3 \arrow[d] & \\
& o_1 o_2 \arrow[d] & \\
& o_1 o_3 \arrow[dl] \arrow[dr] & \\
o_1 \arrow[dr] & & o_2 o_3 \arrow[dl] \\
& o_2 \arrow[d] & \\
& o_3 \arrow[d] & \\
& \varnothing &
\end{tikzcd}$$

Ranking single objects

$m = 3,\ o_1 \rhd o_2 \rhd o_3 \rhd o_4$



Ranking single objects

Pros:

- communication complexity: $O(m.\log m)$.

Cons:

- assumes separability: what will an agent report if she prefers o_2 over o_3 when she has o_1 and o_3 over o_2 if not?

$$o_1 o_2 o_3 \succ o_1 o_2 \succ o_2 o_3 \succ o_1 \succ o_3 \succ o_2 \succ \varnothing$$

$$o_1 \rhd o_3 \rhd o_2 \text{ or } o_1 \rhd o_2 \rhd o_3 \text{ ?}$$

- produces a (very) partial order

Conditional importance networks

(Bouveret, Endriss, Lang, 09)

- allow to express conditional importance statements such as

$$a\bar{b} : cde \triangleright fg$$

if I have a and I do not have b
then I prefer to have $\{c, d, e\}$ rather than $\{f, g\}$
all other things being equal

Conditional importance networks

Conditional importance statement

$\mathcal{S}^+, \mathcal{S}^- : \mathcal{S}_1 \triangleright \mathcal{S}_2$ (with $\mathcal{S}^+$, $\mathcal{S}^-$, $\mathcal{S}_1$ and $\mathcal{S}_2$ pairwise-disjoint).

$\succ$ is compatible with $\mathcal{S}^+, \mathcal{S}^- : \mathcal{S}_1 \triangleright \mathcal{S}_2$ if for every $A, B \subseteq O$ such that

- $A \supseteq \mathcal{S}^+$ and $B \supseteq \mathcal{S}^+$
- $A \cap \mathcal{S}^- = \varnothing$ and $B \cap \mathcal{S}^- = \varnothing$
- $A \supseteq \mathcal{S}_1$ and $B \not\supseteq \mathcal{S}_1$
- $B \supseteq \mathcal{S}_2$ and $A \not\supseteq \mathcal{S}_2$
- for each $o \in O \setminus (\mathcal{S}^+ \cup \mathcal{S}^- \cup \mathcal{S}_1 \cup \mathcal{S}_2)$, we have $o \in A$ iff $o \in B$

then $A \succ B$

Example: $a\overline{d} : b \triangleright ce$ implies for example $ab \succ ace$, $abfg \succ acefg$, ...

CI-net

A CI-net is a set $\mathcal{N}$ of conditional importance statements.

Conditional importance networks

Conditional importance statement

$\mathcal{S}^+, \mathcal{S}^- : \mathcal{S}_1 \triangleright \mathcal{S}_2$ (with $\mathcal{S}^+$, $\mathcal{S}^-$, $\mathcal{S}_1$ and $\mathcal{S}_2$ pairwise-disjoint).

CI-net

A CI-net is a set $\mathcal{N}$ of conditional importance statements on $\mathcal{V}$.

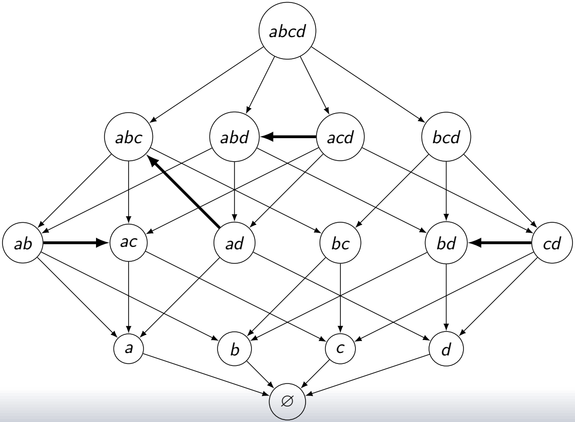
Preference relation induced from a CI-net

$\succ_{\mathcal{N}}$ is the smallest preference relation over $2^{\mathcal{O}}$ such that

- $\succ_{\mathcal{N}}$ is compatible with every conditional importance statement in $\mathcal{N}$
- $\succ_{\mathcal{N}}$ is monotonic

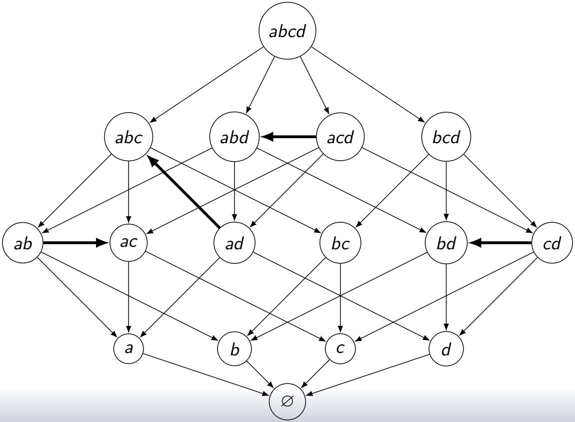
Conditional importance networks

A CI-net of 4 objects $\{a, b, c, d\}$: $\{a : d \triangleright bc, a\overline{d} : b \triangleright c, d : c \triangleright b\}$



Conditional importance networks

A CI-net of 4 objects $\{a, b, c, d\}$: $\{a : d \rhd bc, a\overline{d} : b \rhd c, d : c \rhd b\}$



Induced preference relation $\succ_{\mathcal{N}}$: the smallest preference **monotonic** relation compatible with **all CI-statements**.

Conditional importance networks

- we recover the singleton ranking form when the CI-net is of the form

$$\begin{array}{l} \varnothing, \varnothing : o_1 \rhd o_2 \\ \varnothing, \varnothing : o_2 \rhd o_3; \\ \dots \\ \varnothing, \varnothing : o_{m-1} \rhd o_m \end{array}$$

- CI-nets can express all strict monotonic preference relations on 2^O.
- dominance and satisfiability: PSPACE-complete (existence of exponentially long irreducible dominance sequences)
- in P for precondition-free, singleton-comparing CI-statements (such as $\{a \rhd c, b \rhd c, e \rhd d\}$).

Prioritized goals

- $\Phi = \{\varphi_1, \ldots, \varphi_q\}$ + a weak order $\trianglerighteq$ on $\{\varphi_1, \ldots, \varphi_q\}$
- equivalently, $\Phi = \langle \Phi_1, \ldots, \Phi_q \rangle$ where Φ_1 is the set of highest priority formulas, etc.

leximin semantics $A \succ B$ if there is a $k \leq q$ such that

- $|\{\varphi \in \Phi_i, A \models \Phi_k\}| = |\{\varphi \in \Phi_i, B \models \Phi_k\}|$;
- for each $i < k$: $|\{\varphi \in \Phi_i, A \models \Phi_i\}| = |\{\varphi \in \Phi_i, B \models \Phi_i\}|$.

discriimin semantics $A \succ B$ if there is a $k \leq q$ such that

- $\{\varphi \in \Phi_i, A \models \Phi_k\} \supset \{\varphi \in \Phi_i, B \models \Phi_k\}$;
- for each $i < k$: $\{\varphi \in \Phi_i, A \models \Phi_i\} = \{\varphi \in \Phi_i, B \models \Phi_i\}$.

Particular case: *conditionally lexicographic preferences*

Outline

k-additive utilities

- A utility function over $\mathcal{X} = 2^O$ is **k-additive** if it can be expressed as the sum of sub-utilities over subsets of objects of cardinality $\leq k$.
- Φ: $u : \{S \subseteq O, |S| \leq k\} \to \mathbb{R}$

$$u(x) = \sum_{S \subseteq O, |S| \leq k} u(S)$$

Example: $O = \{a, b, c, d\}, k = 2$

$$u(a, b, d) = u(ab) + u(ad) + u(bd) + u(a) + u(b) + u(d)$$

k-additive utilities

- u is 1-additive $\Leftrightarrow$ u is additive
- every utility function is m-additive ($m = |O|$)
- a k-additive function can be also expressed as the sum of sub-utilities over subsets of attributes of cardinality *exactly* k.

$$u(x) = \sum_{S \subseteq O, |S| = k} v(S)$$

- can be specified by values $v(S)$ for all $|S| = k$: $\binom{m}{k}$ values
- polynomially large if k is a constant, otherwise exponentially large

k-additive utilities

An example

- O consists of 10 pairs of shoes
- $u(S) = 10p + s$ if S contains a total of p matching pairs and in addition s single shoes
- u is 2-additive:
  - $u(\{left_i\}) = u(\{right_i\}) = 1$ for all i
  - $u(\{left_i, right_i\}) = 8$ for all i
- Exercise: express u as the sum of local values of sets of *exactly* two shoes.

k-additive utilities

Another example

Categorized domain: three attributes $N = \{main, dessert, wine\}$, and

$$\mathcal{X} = \{Meat, Fish, Veggie\} \times \{Apple, Cake\} \times \{Red, White\}$$

u_{main}

m	8
f	10
v	12

$u_{dessert}$

a	1
c	5

u_{wine}

r	1
w	0

$u_{main,wine}$

	r	w
m	5	-1
f	-1	5
v	0	0

$u_{main,dessert}$

	a	c
m	2	0
f	0	0
v	0	3

$u_{dessert,wine}$

	a	c
r	0	0
w	0	0

$$\begin{aligned} u(vrc) &= u_M(v) + u_D(c) + u_W(r) + u_{MW}(vr) + u_{MD}(vc) + u_{WD}(rc) \\ &= 12 + 5 + 0 + 0 + 3 + 0 = 18 \end{aligned}$$

- Exercise: find the optimal alternative

Incursion into computational complexity

Two key notions from computational complexity theory:

- a problem is in the class P if it can be solved by an algorithm running in an amount of time bounded by a polynomial function of the size of the input data.
- a decision problem (= checking that a property holds) is in NP (*nondeterministic polynomial time*) if given a solution of the problem, this solution can be verified in polynomial time
- a problem is NP-hard if it is "at least as difficult" as all problems in NP
- a decision problem is NP-complete if (a) it is in NP and (b) it is NP-hard
- is is strongly believed that P is strictly contained in NP (therefore: for solving an NP-complete problem, so far we only have exponential-time algorithms).

k-additive form: complexity

For any $k \geq 2$:

given a k-additive representation...

- and an alternative x, computing $u(x)$ is in P
- and a number α, checking that there exists an alternative x such that $u(x) \geq \alpha$ is NP-complete
- finding x with $u(x)$ maximal is NP-hard (except of course if we know beforehand that preference are monotonic...)

Generalized Additive Independence

GAI-decomposability

Let $\mathcal{X}_1, \dots, \mathcal{X}_k$ be a family of subsets of N such that $\bigcup_i \mathcal{X}_i = N$.
u is GAI-decomposable with respect to $\mathcal{X}_1, \dots, \mathcal{X}_k$ if there exist k subutility functions

$$u_i : \mathcal{X}_i \to \mathbb{R}$$

such that

$$u(x) = \sum_{i=1}^{k} u_i(x_{\mathcal{X}_i})$$

- k-additivity = GAI-decomposability, with $|\mathcal{X}_i| \leq k$ for all i.

Generalized Additive Independence

$$N = \{first, main, dessert, wine\}$$

$$X = \{Soup, Pasta\} \times \{Meat, Fish, Veggie\} \times \{Apple, Cake\} \times \{Red, White\}$$

$$\mathcal{X}_1, \ldots, \mathcal{X}_k = \{\{first\}, \{main, wine\}, \{main, dessert\}\}$$

u_{first}

s	3
p	1

$u_{main,wine}$

	r	w
m	13	7
f	9	15
v	12	12

$u_{main,dessert}$

	a	c
m	2	0
f	0	0
v	0	3

- Dominance is in P
- Optimisation is NP-hard in the general case

Background on propositional logic

Let *ATM* be a set of *propositional symbols*. The propositional language generated from *PS* is the set of *formulas* L_{PS} defined as follows:

- every propositional symbol is a formula;
- $\top$ and $\bot$ are formulas;
- if φ is a formula then $\neg\varphi$ is a formula;
- if φ and ψ are formulas then $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \rightarrow \psi$, and $\varphi \leftrightarrow \psi$ are formula;

$\top$ (`true`) and $\bot$ (`false`): logical constants
$\neg$ (not): unary connective
$\wedge$ (and), $\vee$ (or), $\rightarrow$ (implies), $\leftrightarrow$ (equivalent) : binary connectives.

Background on propositional logic

An *interpretation* (or *valuation*) is a mapping from PS to $\{0,1\}$. An interpretation I is extended to formulas by the following rules:

- $I(\top) = 1$;
- $I(\bot) = 0$;
- $I(\neg\varphi) = 1 - I(\varphi)$;
- $I(\varphi \vee \psi) = \max(I(\varphi), I(\psi))$;
- $I(\varphi \wedge \psi) = \min(I(\varphi), I(\psi))$;
- $I(\varphi \rightarrow \psi) = I(\neg\varphi \vee \psi)$;
- $I(\varphi \leftrightarrow \psi) = I((\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi))$.

I is a *model* of φ, denoted $I \vDash \varphi$, iff $I(\varphi) = 1$

$Mod(\varphi) = \{I \mid I \vDash \varphi\}$

Background on propositional logic

Validity φ is *valid* if $I(\varphi) = 1$ for every interpretation I

$\models \varphi$

Satisfiability φ is *satisfiable* if $I(\varphi) = 1$ for at least one interpretation I

Logical consequence

ψ is a *logical consequence* of φ if every model of φ is a model of ψ

$\varphi \models \psi$

Logical equivalence

φ and ψ are *equivalent* if they are logical consequences of each other

$\varphi \equiv \psi$

Background on propositional logic

Some classes of formulas:

- *literals*: atomic formulas or negations of atomic formulas

$$a \qquad \neg b \qquad \dots$$

- *clauses*: disjunctions of literals, including the empty clause $\perp$

$$a \vee \neg b \vee c \qquad d \vee \neg d \qquad \perp \qquad \dots$$

  - k-clauses: disjunctions of at most k literals

- *cubes*: conjunctions of literals, including the empty cube $\top$

$$a \wedge \neg b \wedge c \qquad d \wedge \neg d \qquad \top \qquad \dots$$

  - k-clauses: conjunctions of at most k literals

- *positive formulas*: formulas in which the only connectives appearing are $\wedge$ and $\vee$

$$a \wedge (b \vee c) \qquad a \vee (b \wedge c) \qquad \dots$$

Binary variables

Particular case: binary variables $\rightarrow D_i = \{\top, \bot\}$ for all i.
Can be used to represent subsets of elements.

Binary variables

Particular case: binary variables $\to D_i = \{\top, \bot\}$ for all i.
Can be used to represent subsets of elements.

A set of elements $\mathcal{O} = \{o_1, \ldots, o_m\} \to$ binary variables $\{o_1, \ldots, o_m\}$, where each variable O_i stands for the presence or absence of o_i.

$\to$ each instantiation / interpretation represents a **subset** π of $\mathcal{O}$

Example of application: allocation of indivisible goods

Example: $o_1 \bar{o}_2 \bar{o}_3 o_4 \bar{o}_5$ represents the subset $\{o_1, o_4\}$.

Of logic and goals

Logic-based languages suit well when we have to deal with **binary variables** (*e.g.* resource allocation problems).

- A propositional syntax $L_{\mathcal{O}}\dots$
  - set of propositional symbols $\mathcal{O}$,
  - usual connectives

Of logic and goals

Logic-based languages suit well when we have to deal with **binary variables** (*e.g.* resource allocation problems).

- A propositional syntax $L_{\mathcal{O}}\dots$
  - set of propositional symbols $\mathcal{O}$,
  - usual connectives

Example

- $\mathcal{O} = \{$  $\}$.
- Set of requests for one agent:
  - ,
  - .

Dichotomous preferences...

What to do with all these goals ?

Dichotomous preferences...

What to do with all these goals ?

A first (simplistic) example: dichotomous preferences.

Example

Variables $\mathcal{O} = \{o_1, o_2, o_3\}$

$$o_2 \wedge (o_1 \vee o_3)$$

represents the dichotomous preference relation

$$\{o_1, o_2, o_3\} \sim \{o_1, o_2\} \sim \{o_2, o_3\} \succ \text{ all others subsets}$$

Weighted logics

Language L_W:

- G = a set of pairs $\langle \varphi_i, w_i \rangle$ where
  - φ_i is a propositional formula;
  - w_i is a real number
- $I_L(G) = u_G$ defined by: for all $x \in 2^{PS}$,

$$u_G(x) = \bigoplus \{ w_i \mid \langle \varphi_i, w_i \rangle \in G \text{ and } x \models \varphi \}$$

- $\oplus$ non-decreasing, symmetric function
- two usual choices: $\oplus = +$ and $\oplus = \max$.
- rest of the talk: $\oplus = +$

Expressing preferences over sets of goods

Binary variables fit well to *resource allocation problems* with *indivisible goods*

- an attribute i = an indivisible object o_i
- $O = \{o_1, \ldots, o_n\}$
- an alternative = a bundle of goods $b_i \subseteq O$

$$o_i \in b \text{ iff } b_i = 1$$

- each agent has to express a utility function over the set of possible bundles 2^O

Expressing preferences over sets of goods

Binary variables fit well to *resource allocation problems* with *indivisible goods*

- an attribute i = an indivisible object o_i
- $O = \{o_1, \ldots, o_n\}$
- an alternative = a bundle of goods $b_i \subseteq O$

$$o_i \in b \text{ iff } b_i = 1$$

- each agent has to express a utility function over the set of possible bundles 2^O

Example

- $\mathcal{O} = \{$  $\}$.
- Set of requests for one agent:
  - ,
  - .

Expressing preferences over sets of goods

Example

- $\mathcal{O} = \{$ , , , , , ,  $\}$.
- Agent 1's requests:
  - $\Big\langle$  $\wedge \Big(($  $\wedge$  $) \vee$  $\Big), 110 \Big\rangle$,
  - $\Big\langle$  $, -10 \Big\rangle$,
  - $\Big\langle$  $\wedge$  $, 50 \Big\rangle$.

Computation of individual utility $(\oplus = +)$:

$\pi_1 = \{$ , , ,  $\}$

Expressing preferences over sets of goods

Example

- $\mathcal{O} = \{$, , , , , , $\}$.
- Agent 1's requests:
  - $\Big\langle$  $\wedge \Big(($  $\wedge$  $) \vee$  $\Big), 110 \Big\rangle$,
  - $\Big\langle$  $, -10 \Big\rangle$,
  - $\Big\langle$  $\wedge$  $, 50 \Big\rangle$.

Computation of individual utility $(\oplus = +)$:

$\pi_1 = \{$, , , $\} \Rightarrow u_1(\pi_1) =$

 $\wedge (($  $\wedge$  $) \vee$  $)$

110

Expressing preferences over sets of goods

Example

- $\mathcal{O} = \{$ , , , , , ,  $\}$.
- Agent 1's requests:
  - $\Big\langle$  $\wedge \Big(($  $\wedge$  $) \vee$  $\Big), 110 \Big\rangle$,
  - $\Big\langle$  $, -10 \Big\rangle$,
  - $\Big\langle$  $\wedge$  $, 50 \Big\rangle$.

Computation of individual utility $(\oplus = +)$:

$\pi_1 = \{$ , , ,  $\} \Rightarrow u_1(\pi_1) = 110$  -10

Expressing preferences over sets of goods

Example

- $\mathcal{O} = \{$, , , , , , $\}$.
- Agent 1's requests:
  - $\Big\langle$  $\wedge \Big(($  $\wedge$  $) \vee$  $\Big), 110 \Big\rangle$,
  - $\Big\langle$  $, -10 \Big\rangle$,
  - $\Big\langle$  $\wedge$  $, 50 \Big\rangle$.

Computation of individual utility $(\oplus = +)$:

$\pi_1 = \{$, , , $\} \Rightarrow u_1(\pi_1) = 110 - 10 +$ ~~ $\wedge$ ~~ 0

Expressing preferences over sets of goods

Example

- $\mathcal{O} = \{$  ,  ,  ,  ,  ,  ,  $\}$.
- Agent 1's requests:
  - $\Big\langle$  $\wedge \Big(($  $\wedge$  $) \vee$  $\Big), 110 \Big\rangle$,
  - $\Big\langle$  $, -10 \Big\rangle$,
  - $\Big\langle$  $\wedge$  $, 50 \Big\rangle$.

Computation of individual utility ($\oplus = +$) :

$\pi_1 = \{$  ,  ,  ,  $\} \Rightarrow u_1(\pi_1) = 110 - 10 + 0 = 100$

Weighted goals: expressive power

Depends on the formulas and the weights allowed in the pairs $\langle \varphi, w \rangle$.
Examples:

- positive cubes + all weights: fully expressive
- literals + all weights: additive functions
- 2-cubes + all weights: 2-additive functions
- cubes + positive weights: non-negative functions
- clauses + positive weights: a proper subset of all nonnegative functions

  Hint $u(\{o_1, o_2\}) = 1$, $u(\{o_1\}) = u(\{o_2\}) = 0$: not expressible!
- positive formulas + positive weights: monotonic non-negative functions
- positive cubes + positive weights: a proper subset of all monotonic non-negative functions

Weighted goals: succinctness

- all formulas + all weights: fully expressive
- positive cubes + all weights: fully expressive

But

all formulas + all weights
more succinct than
positive cubes + all weights

Hint: try to express u defined by

$$u(x) = \max_{i=1,\ldots,n} x_i$$

Weighted goals: computational complexity

- comparing two alternatives: can be solved in polynomial time
- finding an optimal alternative: NP-complete in the general case, even for dichotomous utilities
- finding an optimal alternative: polynomial for some restrictions of the language
  - monotonic fragment (no negation, positive weights)
  - additive fragment (literals only)