Fair Division: Computation and Communication

Jérôme Lang
LAMSADE, CNRS / Université Paris-Dauphine

Summer School on Fair Division
St Petersburg, August 8, 2017

Outline

Fair division: taxonomy of problems

Computation and communication
- Communication
- Combinatorial spaces and compact representation

Centralized mechanisms
- Cardinal, additive preferences
- Cardinal, non-additive preferences
- Ordinal, separable preferences
- Ordinal, nonseparable preferences

Decentralized mechanisms

Fair division problem (informal)

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

... a final allocation is found somehow

Fair division problem (informal)

- a set of *n agents*
- a set of *resources* to be allocated
- 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.

Partially decentralized mechanism interaction protocol + central authority

2. Divisible versus indivisible resources

Divisible resources (homogeneous or heterogeneous)

Indivisible resources or *items*

3. Ordinal versus cardinal preferences

(for centralized and semi-centralized mechanisms)

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

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

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

(for indivisible resources)

One-to-one allocation

Each agent gets exactly one item: *matching problem*

Many-to-one allocation

Each agent gets possibly several items.

5. Deterministic versus randomized

Deterministic allocation Output = single allocation (possibly using some tie-breaking mechanism)

Randomized allocation Output = probability distribution over allocations.

6. Money or not

Money Agents may pay and/or receive money

No money

7. Initial endowments or not

Initial endowments The agents initially own resources (other than money)

No initial endowments

Fair division

1. centralized or decentralized
2. divisible or indivisible
3. ordinal or cardinal
4. one-to-one or many-to-one
5. deterministic or randomized
6. money or no money
7. initial endowments or no initial endowments

Fair division

1. centralized or decentralized
2. divisible or indivisible
3. ordinal or cardinal
4. one-to-one or many-to-one
5. deterministic or randomized
6. money or no money
7. initial endowments or no initial endowments
8. goods only / goods and bads
9. nonshareable goods / partially shareable goods
10. (...)

for (almost) all the rest of the lecture

Centralized fair division

(From now on: indivisible goods)

Given

- a set of *m items* to be allocated,
- a set of *n agents*,
- *preferences* of agents over items,
- the final *allocation* being subject to some *feasibility constraints*,
- *criteria* for evaluating the quality of allocations

the central authority determines an allocation of items to agents

Decentralized fair division

Given

- a set of *m items* to be allocated
- a set of *n agents*
- the final *allocation* being subject to some *feasibility constraints*
- *criteria* for evaluating the quality of allocations,

an interaction protocol between the agents is performed, resulting in an allocation.

Semi-decentralized fair division

Given

- a set of *m items* to be allocated
- a set of *n agents*
- the final *allocation* being subject to some *feasibility constraints*
- *criteria* for evaluating the quality of allocations,

an interaction protocol between the agents and the central authority is performed, resulting in an allocation.

Outline

Fair division: taxonomy of problems

Computation and communication
- Communication
- Combinatorial spaces and compact representation

Centralized mechanisms
- Cardinal, additive preferences
- Cardinal, non-additive preferences
- Ordinal, separable preferences
- Ordinal, nonseparable preferences

Decentralized mechanisms

Computation and communication

Computation What are the computational resources (time and space) needed to determine the outcome of the mechanism? How should this outcome be computed in practice?

- an algorithm $\mathcal{A}$ *computes* a mechanism F if for any instance I, running $\mathcal{A}$ on I outputs $F(I)$.
- (very informally) cost of an algorithm: worst case number of resources, over all instances, used by the algorithm.
- computational complexity of mechanism: cost of the cheapest algorithm that computes it.

Computation and communication

|

Communication What is the length of the interaction between the agents (and possibly the central authority) needed to determine the outcome of the mechanism? How should this interaction be performed?

- a *communication protocol* $\mathcal{P}$ is similar to an algorithm, but the basic elements are communication acts instead of instructions for the computer.
- $\mathcal{P}$ is a protocol for a mechanism F if for any instance I, executing P on I outputs $F(I)$.
- cost of a protocol: worst case number of bits, over all instances, exchanged between the agents between themselves and/or with the central authority.
- communication complexity of mechanism: cost of its cheapest protocol.

Communication: centralized mechanisms

A centralized protocol works as follows:

1. each agent reports her preferences P_i (ordinal or cardinal), in some format
2. the central authority computes $F(P_1, \ldots, P_n)$

- communication complexity of F = number of bits needed to specify $P_1, \ldots, P_n$
- D space of all admissible preferences (preference relations or utility functions)
- each agent needs to send at most $\log |D|$ bits
- communication complexity: $O(n \log |D|)$
- this upper bound is not always reached
- find some mechanisms needing less communication

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 a left shoe then my value for a right shoe is higher than in the opposite case.

Fair division

1. centralized or decentralized
2. divisible or indivisible
3. ordinal or cardinal
4. one-to-one or many-to-one
5. deterministic or randomized
6. money or no money
7. initial endowments or no initial endowments
8. goods only / goods and bads
9. nonshareable goods / partially shareable goods
10. separable preferences or not

Separable preferences

- $\succeq$ preference relation over 2^O is *additively separable* if there are m values $v(o_1), \ldots, v(o_m)$ such that for all $A, B \subseteq O$,

$$A \succeq B \text{ if and only if } \sum_{o \in A} v(o) \geq \sum_{o \in B} v(o)$$

- $\succeq$ is *separable* if for each $A, B, C \subseteq O$ with $A \cap C = B \cap C = \varnothing$,

$$A \cup C \succeq B \cup C \text{ if and only if } A \succeq B$$

- $\succeq$ is *weakly separable* if for each $a, b, C \subseteq O$ with $a \notin C$, $b \notin C$,

$$\{a\} \cup C \succeq \{b\} \cup C \text{ if and only if } \{a\} \succeq \{b\}$$

- additively separability $\Rightarrow$ separability $\Rightarrow \succeq$ weak separability

Separable preferences

- $abc \succ ab \succ ac \succ bc \succ c \succ a \succ b \succ \varnothing$

Separable preferences

- $abc \succ ab \succ ac \succ bc \succ c \succ a \succ b \succ \varnothing$
  - not weakly separable
- $abcd \succ abc \succ abd \succ acd \succ bcd \succ ab \succ ac \succ bc \succ bd \succ ad \succ a \succ cd \succ b \succ c \succ d \succ \varnothing$

Separable preferences

- $abc \succ ab \succ ac \succ bc \succ c \succ a \succ b \succ \varnothing$
  - not weakly separable
- $abcd \succ abc \succ abd \succ acd \succ bcd \succ ab \succ ac \succ bc \succ bd \succ ad \succ a \succ cd \succ b \succ c \succ d \succ \varnothing$
  - weakly separable
  - not separable
- $\ldots \succ ad \succ bd \succ ae \succ af \succ be \succ cd \succ ce \succ bf \succ cf \succ \ldots$
  - (can be completed so as to be) separable
  - not additively separable

Separable preferences

- $abc \succ ab \succ ac \succ bc \succ c \succ a \succ b \succ \varnothing$
  - not weakly separable
- $abcd \succ abc \succ abd \succ acd \succ bcd \succ ab \succ ac \succ bc \succ bd \succ ad \succ a \succ cd \succ b \succ c \succ d \succ \varnothing$
  - weakly separable
  - not separable
- $\ldots \succ ad \succ bd \succ ae \succ af \succ be \succ cd \succ ce \succ bf \succ cf \succ \ldots$
  - (can be completed so as to be) separable
  - not additively separable

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

Preferences over bundles

- N set of agents
- $O = \{o_1, \ldots, o_m\}$ indivisible items

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 $[\varnothing | o_3 | o_1 o_2 o_4 o_5]$
- etc.

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

Preferences

Ordinal preferences

$\succeq$ weak order on $\mathcal{X}$

$$
\begin{array}{lcl}
x \succeq y & & x \text{ is at least as good as } y \\
x \succ y & \Leftrightarrow & x \succeq y \text{ and not } y \succeq x \ (\textit{strict preference}) \\
x \sim y & \Leftrightarrow & x \succeq y \text{ and } y \succeq x \ (\textit{indifference})
\end{array}
$$

Cardinal preferences

$u : \mathcal{X} \to \mathbb{R}$

Dichotomous preferences

$A \subseteq \mathcal{X}$ set of acceptable bundles

Monotonicity

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

Typically, preferences over bundles are *monotonic*: receiving one more item never makes an agent less happy. *(In particular: no bads, no chores).*

- 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 $u(S) > u(S')$

Combinatorial spaces...

- $O = \{o_1, \ldots, o_m\}$ indivisible items
- 2^O set of all *bundles of items*
- $\mathcal{X} \subseteq 2^O$ 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 items. . .

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

Combinatorial spaces. . .

The combinatorial trap. . .

Four items. . .

$o_1 o_2 o_3 o_4 \succ o_1 o_2 o_4 \succ o_1 o_3 o_4 \succ o_2 o_3 o_4 \succ o_1 o_2 o_3 \succ o_1 o_3 \succ o_2 o_4 \succ o_3 o_4 \succ o_1 o_4 \succ o_1 \succ o_2 \succ o_4 \succ o_3 \succ \varnothing \rightarrow 16$ subsets

Combinatorial spaces...

The combinatorial trap...

Twenty items...

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

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

The dilemma

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

Outline

Fair division: taxonomy of problems

Computation and communication
- Communication
- Combinatorial spaces and compact representation

Centralized mechanisms
- Cardinal, additive preferences
- Cardinal, non-additive preferences
- Ordinal, separable preferences
- Ordinal, nonseparable preferences

Decentralized mechanisms

Centralized mechanisms: cardinal, additive preferences

- each agent i specifies a value $u_i(o_j)$ for each $o_j \in O$.
- for each $S \subseteq O$, $U_i(S) = \sum_{o \in S} u_i(o)$.

Two possible approaches:

1. fix an optimization criterion and output the optimal allocation
2. fix a set of properties that we want the allocation to satisfy, and check if there is one.

Centralized mechanisms: cardinal, additive preferences

- each agent i specifies a value $u_i(o_j)$ for each $o_j \in O$.
- for each $S \subseteq O$, $U_i(S) = \sum_{o \in S} u_i(o)$.

Two possible approaches:

1. fix an optimization criterion and output the optimal allocation
2. fix a set of properties that we want the allocation to satisfy, and check if there is one.

Centralized, additive preferences: maximizing social welfare

- utilitarian social welfare:

$$U(\pi) = \sum_{i=1}^{n} U_i(\pi(i))$$

- egalitarian social welfare:

$$U(\pi) = \max_{i=1,\ldots,n} U_i(\pi(i))$$

- egalitarian social welfare, leximin refinement:
  - if $U = (U_1, \ldots, U_n)$ then $U^{\uparrow} = (U_1^{\uparrow}, \ldots, U_n^{\uparrow}) = (U_{\sigma(1)}, \ldots, U_{\sigma(n)})$ where σ is a permutation of $\{1, \ldots, n\}$ such that $U_{\sigma(1)} \leq \ldots \leq U_{\sigma(n)}$.
  - $(U_1, \ldots, U_n)$ leximin-dominates $(V_1, \ldots, V_n)$ if there exists a k such that $U_k^{\uparrow} > V_k^{\uparrow}$ and for each $j < k$, $U_j^{\uparrow} = V_j^{\uparrow}$.
- Nash social welfare:

$$U(\pi) = \prod_{i=1}^{n} U_i(\pi(i))$$

Brief incursion into computational complexity

- search problem: given a relation F between instances and solutions: given instance I, output a solution s such that $F(I, s)$, if such a solution exists.
- decision problem: given a set of instances $\mathcal{I}$ and a subset $\mathcal{I}^+$ of positive instances: given instance I, is it true that $I \in \mathcal{I}^+$?
- a decision or search problem is in class P (*deterministic polynomial time*) if it can be solved by an algorithm running in an amount of time bounded by a polynomial function of the size of the instance.
- (informally) a decision problem is in NP (*nondeterministic polynomial time*) if given a solution of the problem, this solution can be verified in deterministic polynomial time.
- (informally) a decision or search problem is NP-hard if it is "at least as difficult" as all problems in NP.
- a decision problem is NP-complete if it is (a) in NP and (b) NP-hard.
- is is strongly believed that P is strictly contained in NP.

Centralized, additive preferences, utilitarian social welfare

- simply allocate each item to the agent who values it best

$$o_j \in \pi(i) \text{ if } o_j = \operatorname*{argmax}_{o \in O} u_i(o)$$

- little to do with fairness, even if values are normalized
- obviously in P (but becomes NP-hard if preferences are not additively separable)

Centralized, additive, egalitarian: the Santa Claus problem

- centralized mechanism; cardinal, additive preferences;
- egalitarian social welfare

$$U(\pi) = \max_{i=1,\ldots,n} U_i(\pi(i))$$

- return an allocation maximizing egalitarian social welfare.

The Santa Claus problem: computational complexity

1. Bad news: computing an optimal egalitarian allocation is NP-hard, even in the simple case of two agents with identical preferences.
2. Good news: the problem is *pseudo-polynomial* for a fixed number of agents.
3. (Rather) good news: polynomial approximation algorithms.
4. Good news: translation into integer linear programming.

The Santa Claus problem: NP-hardness

Bad news: NP-hardness, even for two agents with identical valuations.

- SUBSET SUM: given a collection of integers $S = (s_1, \ldots, s_p)$ and an integer K, is there a subset S' of S such that $\sum_{i \in S'} s_i = K$?
- $(S = (3, 3, 5), K)$: positive instance of SUBSET SUM for $K = 3, 5, 6, 8, 11$
- PARTITION subproblem of SUBSET SUM where $\sum_{i=1}^{m} s_i = 2K$.
- PARTITION, and a fortiori SUBSET SUM, are known to be NP-complete
- reducing PARTITION to OPTIMAL EGALITARIAN ALLOCATION:

The Santa Claus problem: NP-hardness

Bad news: NP-hardness, even for two agents with identical valuations.

- SUBSET SUM: given a collection of integers $S = (s_1, \ldots, s_p)$ and an integer K, is there a subset S' of S such that $\sum_{i \in S'} s_i = K$?
- $(S = (3, 3, 5), K)$: positive instance of SUBSET SUM for $K = 3, 5, 6, 8, 11$
- PARTITION subproblem of SUBSET SUM where $\sum_{i=1}^{m} s_i = 2K$.
- PARTITION, and a fortiori SUBSET SUM, are known to be NP-complete
- reducing PARTITION to OPTIMAL EGALITARIAN ALLOCATION:
- p items, two agents with identical preferences: $u_i(o_1) = s_1, \ldots, u_i(o_p) = s_p$;
- S is a positive instance of PARTITION if the optimal allocation has egalitarian social welfare equal to K.

The Santa Claus problem: pseudo-polynomiality

Good news: *pseudo-polynomiality* for a fixed number of agents

- let $W = \max_{i,j} u_i(o_j)$
- there exists an algorithm that finds an optimal allocation in time $O(W.2^n pol(n, m))$
- does this mean that the problem is in P if the number of agents is a constant?

The Santa Claus problem: pseudo-polynomiality

Good news: *pseudo-polynomiality* for a fixed number of agents

- let $W = \max_{i,j} u_i(o_j)$
- there exists an algorithm that finds an optimal allocation in time $O(W.2^n pol(n, m))$
- does this mean that the problem is in P if the number of agents is a constant? No, because the size needed to represent W is $\log W$.
- $O(W.2^n pol(n, m))$ is polynomial in the size of the input if both W and n are (bounded by a) constant.
- informally: if the valuations are small, and the number of agents is small too, then computing an optimal egalitarian allocation is easy.

The Santa Claus problem: pseudo-polynomiality

Good news: *pseudo-polynomiality* for a fixed number of agents

- let $W = \max_{i,j} u_i(o_j)$
- there exists an algorithm that finds an optimal allocation in time $O(W.2^n pol(n, m))$
- does this mean that the problem is in P if the number of agents is a constant? No, because the size needed to represent W is $\log W$.
- $O(W.2^n pol(n, m))$ is polynomial in the size of the input if both W and n are (bounded by a) constant.
- informally: if the valuations are small, and the number of agents is small too, then computing an optimal egalitarian allocation is easy.

We will now find such an algorithm

1. first in the simple case $n = 2$ and identical valuations.
2. then $n = 2$, any valuations
3. then in the general case: n constant > 2.

The Santa Claus problem: pseudo-polynomiality

1. $\mathbf{n} = 2$ and identical valuations

- *Input* $\alpha_1 = u_1(o_1) = u_2(o_1), \ldots, \alpha_m = u_1(o_m) = u_2(o_m)$
- $K = \left\lceil \left(\sum_{i=1}^{m} \alpha_i\right)/2 \right\rceil$
- we construct a table $T(w, i)$ for each $w = 0, \ldots, K$ and $i = 1, \ldots, m$
- $T(w, i)$ contains a subset of $\{\alpha_1, \ldots, \alpha_i\}$ of sum w if such a subset exists; otherwise, $T(w, i) = nil$.

$T(\alpha_1, 1) \leftarrow \{o_1\}$; $T(0, 1) \leftarrow \varnothing$; for $w \notin \{0, \alpha_1\}$: $T(w, 1) \leftarrow nil$.
for $i = 2, \ldots, m$
  for $w = 0, \ldots, K$
    if $T(w, i-1) \neq nil$
    then $T(w, i) \leftarrow T(w, i-1)$
    else if $T(w - \alpha_i, i-1) \neq nil$
      then $T(w, i) \leftarrow T(w, i-1) \cup \{o_i\}$
      else $T(w, i) \leftarrow nil$ **endif**
  end for
end for
$w^* \leftarrow \max\{w, T(w, m) \neq nil\}$;
return the allocation $[T(w^*, m) \mid \overline{T(w^*, m)}]$

The Santa Claus problem: pseudo-polynomiality

2. $\mathbf{n} = 2$ and any valuations

- *Input*: for each $i = 1, \ldots, m$: $\alpha_i = u_1(o_i)$, $\beta_i = u_2(o_i)$
- $K = \left\lceil \left(\sum_{i=1}^{m} \max(\alpha_i, \beta_i) \right) / 2 \right\rceil$
- construct partial allocation $\pi(v, w, i)$ for each $(v, w) \in \{0, \ldots, K\}^2$ and each $i \leq m$
- $\pi(v, w, i)$ contains a partial allocation of $\{o_1, \ldots, o_i\}$ giving utility v to agent 1 and w to agent 2 if there is one; $\pi(v, w, i) = \mathit{nil}$ otherwise

The Santa Claus problem: pseudo-polynomiality

2. $\mathbf{n} = 2$ and any valuations

$\pi(0,0,1) \leftarrow [\varnothing|\varnothing]$;
for each $(v,w) \neq (0,0)$: $\pi(w,w,1) \leftarrow \mathit{nil}$ **end for**
for $i = 1,\ldots,m$
  for $(v,w) \in \{0,\ldots,K\} \times \{0,\ldots,K\}$
    if $\pi(v-\alpha_i, w, i-1) \neq \mathit{nil}$
    then $\pi(v,w,i) \leftarrow \pi(v,w,i-1) + (o_i|\varnothing)$;
    else if $\pi(v, w-\beta_i, i-1) \neq \mathit{nil}$
      then $\pi(v,w,i) \leftarrow \pi(v,w,i-1) + (\varnothing|o_i)$;
      else $\pi(w,i) \leftarrow \mathit{nil}$ **endif**
  end for
end for
let (v^*,w^*) s.t. $\min(v^*,w^*)$ is maximal and $\pi(v^*,w^*,m) \neq \mathit{nil}$;
return $\pi(v^*,w^*,m)$

The Santa Claus problem: pseudo-polynomiality

3. general case: **n** constant + any valuations

- *Input*: for each $i = 1, \ldots, n$, $j = 1, \ldots, m$: $u_i^j = u_1(o_j)$.
- $K = \left\lceil \left(\sum_{j=1}^{m} \max(u_1^j, \ldots, u_n^j) \right) / n \right\rceil$
- construct $\pi(v, w, i)$ for all $(v_1, \ldots, v_n) \in \{0, \ldots, K\}^n$ and all $i \leq m$
- $\pi(v, w, i)$ contains a partial allocation of $\{o_1, \ldots, o_i\}$ giving utility v_i to agent i if there is one; $\pi(v, w, i) = nil$ otherwise
- time needed to fill the table $O(m.K^n)$
- remains polynomial in the size of the input if K and n are constant
- works more generally for any notion of social welfare defined by a polynomially computable aggregation function $\star$

The Santa Claus problem: pseudo-polynomiality

3. general case: **n** constant + any valuations

- let $W = \max_{i,j} u_i(o_j)$
- there exists an algorithm that finds an optimal allocation in time $O(W.2^n pol(n, m))$
- the size needed to represent W is $\log W$, therefore $O(W.2^n pol(n, m))$ is polynomial in the size of the input if W and n are constant.
- informally: if the valuations are small, and the number of agents is small too, computing an optimal egalitarian allocation is easy.

The Santa Claus problem: polynomial approximation

Good news, 2: *polynomial approximation*

- first for $n = 2$ and identical valuations $\alpha_1, \ldots, \alpha_m$
- *Input* $\alpha_1 = u(o_1), \ldots, \alpha_m = u(o_m)$
- for each $S \subseteq \{1, \ldots, n\}$: $V(S) = \sum_{j \in S} \alpha_j$.
- Problem $P(k)$: find S s.t. $V(S)$ is maximal but no larger than k.
- a greedy approximation algorithm for $P(k)$:

$S \leftarrow \varnothing$
for $i = 1$ to n
  if $\alpha_i + V(S) \leq K$
  then $S \leftarrow S \cup \{j\}$
  endif
end for
return S

- let S^* be the optimal subset
- show that $V(S) \geq \frac{1}{2} V(S^*)$
- show that the $\frac{1}{2}$ bound is asympotically reached.

The Santa Claus problem: polynomial approximation

- $n = 2$ and identical valuations $\alpha_1, \ldots, \alpha_m$
- S^* optimal subset: $V(S^*)$ maximal subject to $V(S^*) \leq k$
- $\pi^* = [S^* \mid \overline{S^*}]$ optimal egalitarian solution
- $\pi = [S \mid \overline{S}]$ returned by the algorithm
- $\min_{i=1,2} u_i(\pi(i)) \leq \frac{1}{2} \min_{i=1,2} u_i(\pi^*(i))$
- the algorithm is a polynomial 2-approximation algorithm.
- can we get a better polynomial approximation? yes

The Santa Claus problem: polynomial approximation

- $n = 2$ and identical valuations $\alpha_1, \ldots, \alpha_m$
- problem P: optimize egalitarian social welfare
- exact algorithm: at each step i, construct the list of all reachable sums from $\{\alpha_1, \ldots, \alpha_i\}$: exponential time
- greedy algorithm: at each step, store exactly one number: linear time, approximation guarantee $= 2$
- *inbetween*: at each step, keep a list of reachable sums of "reasonable cardinality"
- given parameter δ, trim the current list of possible sums L into L' by removing elements as long as for every $y \in L$ there exists $z \in L'$ such that $(1 - \delta)y \leq z \leq y$.

The Santa Claus problem: polynomial approximation

We want a polynomial r-approximation algorithm for each $r > 1$.

- input: without loss of generality, $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_m$

  $\delta = \frac{r-1}{m}$

  $L_0 = \{0\}$

  for $j = 1$ to m **do**

  $L_i \leftarrow \mathit{MergeLists}(L_i, L_{i-1} + \alpha_i)$

  $L_i \leftarrow \mathit{Trim}(L_i, \delta)$

  remove from L all elements bigger than K

  return the partition corrresponding to the largest element from L_m

- the algorithm gives a r-approximation of the initial problem
- complexity: $O(m^2 \log(K)/1 - r)$: polynomial in the size of the input data and in $\frac{1}{1-r}$
- A is a fully polynomial-time approximation scheme for problem P.

The Santa Claus problem: polynomial approximation

- 2 agents with identical valuations: fully polynomial-time approximation scheme
- it does not carry on to the general case:
  - approximated to a factor $m - n + 1$
  - cannot be approximated to a factor < 2, unless P $=$ NP (Bezakova and Deni, 2005)

The Santa Claus problem: translation into ILP

Good news, 3: *translation into zero-one linear programming*

- Input: n agents, m items, valuations $u_i^j = u_i(o_j)$
- Goal: find an allocation maximizing $\min_i \sum_{j \in \pi(i)} v_i(o_j)$

- introduce nm variables x_i^j (agent i gets item o_j)
- solve the following problem:

$$\max(t)$$
$$s.t. \left| \begin{array}{ll} \sum_{j=1}^{m} x_i^j u_1^j \geq t & \forall i = 1, \ldots, n \\ \sum_{i=1}^{n} x_i^j = 1 & \forall j = 1, \ldots, m \\ x_i^j \in \{0, 1\} & \forall i, j \end{array} \right.$$

- zero-one linear programming is NP-hard but there exist now very good off-the-shelf solvers.

Centralized, additive preferences, leximin

- compute $w_1 = \max_\pi \min_i u_i(\pi(i))$
- for $k = 1, \ldots, n$, solve P_k:

$$\max(t) \quad s.t. \left| \begin{array}{l} \sum_{j=1}^{n} x_k^j = w_1 \\ \sum_{j=1}^{m} x_1^j u_1^j \geq t \quad \forall i \neq k \\ (\ldots) \end{array} \right.$$

- w_2 : maximal value obtained for $P_1, \ldots, P_n$.
- for $(k, k') = 1, \ldots, n, k \neq k'$, solve $P_{k,k'}$:

$$\max(t) \quad s.t. \left| \begin{array}{l} \sum_{j=1}^{n} x_k^j = w_1 \\ \sum_{j=1}^{n} x_{k'}^j = w_2 \\ \sum_{j=1}^{m} x_1^j u_1^j \geq t \quad \forall i \neq k, k' \\ (\ldots) \end{array} \right.$$

- w_3 : maximal value obtained for all $P_{k,k'}$.
- for all (k, k', k''), solve $P_{k,k',k''}$:
- etc.

Centralized, additive preferences, leximin

- exponential number of 0-1 linear programs
- but polynomial number of n is a constant
- other algorithms (Bouveret and Lemaître, 2009)

Centralized, additive preferences, Nash

- maximize $U(\pi) = \prod_{i=1}^{n} U_i(\pi(i))$
- NP-hard
- pseudo-polynomial for constant number of agents
- constant factor approximation $2e^{1/e} \approx 2.9$ (Cole & Gkatzelis, 2017)
- solvable by a convex integer program

Centralized mechanisms: cardinal, additive preferences

Two possible approaches:

1. fix an optimization criterion and output the optimal allocation
2. fix a set of properties that we want the allocation to satisfy, and check if there is one.

Centralized, additive preferences, envy-freeness

- π is envy-free if for all i, j, $u_i(\pi(i)) \geq u_i(\pi(j))$
- envy-freeness has to be coupled with an efficiency criterion, in particular:
  - *completeness*: all items must be allocated
  - *Pareto-efficiency*: no allocation π' is at least as good as π for all agents and strictly better for at least one agent.

Centralized, additive preferences, envy-freeness

- envy-freeness + completeness

$$\left|\begin{array}{ll} \sum_{k=1}^{m} x_i^k u_i^k \geq \sum_{k=1}^{m} x_j^k u_i^k & \forall i,j = 1,\ldots,n \\ \sum_{i=1}^{n} x_1^j = 1 & \forall j = 1,\ldots,m \\ x_i^j \in \{0,1\} & \forall i,j \end{array}\right.$$

$$\left|\begin{array}{ll} \sum_{k=1}^{m} (x_i^k - x_j^k) u_i^k \geq 0 & \forall i,j = 1,\ldots,n \\ \sum_{i=1}^{n} x_1^j = 1 & \forall j = 1,\ldots,m \\ x_i^j \in \{0,1\} & \forall i,j \end{array}\right.$$

- existence of a complete EF allocation is NP-complete
- degree of envy of an allocation

$$e(\pi) = \max_{i,j} \max(0, u_i(\pi(j)) - u_i(\pi(i))$$

- (Lipton, Markakis, Mossel, Saberi, 2004):
  - Let $\alpha = \max_{i,o} u_i(o)$. There always exists an allocation π such that $e(\pi) \leq \alpha$, and such an allocation can be found in time $O(mn^3)$.
  - If the utilities are identical: [fully] polynomial-time approximation scheme [for n constant]

Centralized, additive preferences, envy-freeness

- envy-freeness + Pareto-efficiency
- verifying that an allocation is complete and envy-free: $O(n^2 m)$
- therefore: existence of an envy-freene + complete allocation is in NP
- does this extend to the existence of an envy-free and Pareto-efficient allocation?
  - checking whether an allocation is envy-free: polynomial.
  - checking whether an allocation is Pareto-efficient:

Centralized, additive preferences, envy-freeness

- envy-freeness + Pareto-efficiency
- verifying that an allocation is complete and envy-free: $O(n^2 m)$
- therefore: existence of an envy-freene + complete allocation is in NP
- does this extend to the existence of an envy-free and Pareto-efficient allocation?
  - checking whether an allocation is envy-free: polynomial.
  - checking whether an allocation is Pareto-efficient: coNP-complete. A problem $\mathcal{I}, \mathcal{I}^+$ is in coNP if its complement problem $\mathcal{I}, \mathcal{I}^-$ is in NP.

Centralized, additive preferences, envy-freeness

- envy-freeness + Pareto-efficiency
- verifying that an allocation is complete and envy-free: $O(n^2 m)$
- therefore: existence of an envy-freene + complete allocation is in NP
- does this extend to the existence of an envy-free and Pareto-efficient allocation?
  - checking whether an allocation is envy-free: polynomial.
  - checking whether an allocation is Pareto-efficient: coNP-complete.
  - existence of an envy-free and Pareto-efficient allocation: in Σ_2^p
  - $\Sigma_2^p = \mathrm{NP}^{\mathrm{NP}}$ set of problems solvable by a nondeterministic polynomial algorithm equipped with NP-oracles.
  - strongly believed that NP $\neq$ coNP and that both are strictly contained in Σ_2^p.
  - de Keijzer, Bouveret, Klos and Zhang, 2009: Σ_2^p-complete.

Centralized, additive preferences, maxmin fair share

Given a fair division problem:

- for each agent i, the maximin fair share value of i is her value of the worst share of the best possible partition

$$MmFS(i) =: \max_{\pi} \min_{j} u_i(\pi(j))$$

- π satisfies the maxmin fair share property if for all i,

$$u_i(\pi(i)) \geq MmFS(i)$$

- weaker than envy-freeness
- under additive preferences, a maxmin fair share allocation always exists (a) for $n = 2$, and (b) for any n and identical preferences.
- in the general case, it is difficult to find a problem with no maxmin fair share allocation (Bouveret and Lemaitre 2014) but counterexamples exist (Procaccia and Wang 2014).

Centralized, additive preferences, maxmin fair share

- given i's preferences and the number of agents n, computing $MmFS(i)$ is NP-hard, even with $n = 2$(reduction from PARTITION).
- given problem P and allocation π, what is the complexity of *determining whether π satisfies the maxmin fair share property?*.

Centralized, additive preferences, maxmin fair share

- given i's preferences and the number of agents n, computing $MmFS(i)$ is NP-hard, even with $n = 2$(reduction from PARTITION).
- given problem P and allocation π, what is the complexity of *determining whether π satisfies the maxmin fair share property?*.
  - in NP:
    for each $i = 1, \ldots, n$
      guess π^i
      check that $u_i(\pi(i) \geq u_i(\pi^i(j))$ for each $j = 1, \ldots, m$
    end for
  - NP-hard: reduction from PARTITION.
  - NP-complete.

Centralized, additive preferences, maxmin fair share

- given i's preferences and the number of agents n, computing $MmFS(i)$ is NP-hard, even with $n = 2$(reduction from PARTITION).
- given problem P and allocation π, what is the complexity of *determining whether π satisfies the maxmin fair share property?* NP-complete.
- given problem P, compute a maxmin fair share allocation: NP-hard, even for two agents with identical preferences.
- given problem P, determine if there exists a maxmin fair share allocation:
  - in NP:
    guess $n+1$ allocations $\pi, \pi^1, \ldots, \pi^n$
    check that $u_i(\pi(i) \geq u_i(\pi^i(j))$ for each $j = 1, \ldots, m$
  - NP-hard?

Centralized, additive preferences, proportional and minmax fair share

Given a fair division problem:

- the proportional fair share value of i is $PFS(i) = \frac{1}{n} \sum_{j=1}^{m} u_i^j$
- π satisfies the *proportional fair share* property if for all i, $u_i(\pi(i)) \geq PFS(i)$
- the minmax fair share value of i is her value of the best share of the worst possible partition:

$$mMFS(i) = \min_{\pi} \max_{j} u_i(\pi(j))$$

- π satisfies the minmax fair share property if for all i, $u_i(\pi(i)) \geq mMFS(i)$
- envy-freeness $\Rightarrow$ minmax FS $\Rightarrow$ proportional FS $\Rightarrow$ FS
- complexity of verification, search and existence for proportional and minmax FS?

Outline

Fair division: taxonomy of problems

Computation and communication
- Communication
- Combinatorial spaces and compact representation

Centralized mechanisms
- Cardinal, additive preferences
- Cardinal, non-additive preferences
- Ordinal, separable preferences
- Ordinal, nonseparable preferences

Decentralized mechanisms

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*

Centralized mechanisms: dichotomous preferences

- 2^O set of all bundles of items
- each agent partitions the set of all bundles between *good* and *bad* bundles: $O = O^+ \cup O^-$
- for each $S \in Good$, $u(S) = 1$
- for each $S \in Bad$, $u(S) = 0$

- $O = \{b(eer), c(offee), s(ugar)\}$
- Emmanuel wants a beer, or else coffee with sugar.
  Two possible meanings:
  1. $Good_E = \{\{b\}, \{c, s\}\}$: *nonmonotonic*
  2. $Good_E = \{\{b, c\}, \{b, s\}, \{b, c\}, \{b, s\}, \{b, c, s\}\}$: *monotonic*
- Marine wants a beer.
- Jean-Luc wants a beer or a coffee.

How can we express such preferences in a succinct way?

Incursion to propositional logic

Let *PS* 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.

Incursion to 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:

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

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

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

Incursion to 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

$\varphi \equiv \psi$

Incursion to propositional logic

Some classes of formulas:

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

$$a \qquad \neg b \qquad \ldots$$

- *clauses*: disjunctions of literals, including $\bot$

$$a \vee \neg b \vee c \qquad d \vee \neg d \qquad \bot \qquad \ldots$$

  - k-clauses: disjunctions of at most k literals

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

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

  - k-clauses: conjunctions of at most k literals

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

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

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

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

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

Example of application: allocation of indivisible items

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 $L_{\mathcal{O}}$...
  - 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 $L_{\mathcal{O}}$...
  - set of propositional symbols $\mathcal{O}$,
  - usual connectives

Example

- $\mathcal{O} = \{$  ,  ,  ,  ,  ,  ,  $\}$.
- Set of requests for one agent:
  -  $\wedge \Big($ ( $\wedge$ ) $\vee$  $\Big)$,
  -  $\wedge$  .

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

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

Dichotomous preferences

- three items: one cup of coffee, one glass of beer, one sugar cube
- three agents: E(mmanuel), M(arine), J(ean-Luc), with *dichotomous* preferences:
  - Emmanuel wants a beer, or else coffee with sugar.
  - Marine wants a beer.
  - Jean-Luc wants a beer or a coffee.
- can they all be satisfied?
  - $b_J \vee c_J$ where b_J means: *the beer is allocated to Jean-Luc*
  - $b_E \vee (c_E \wedge s_E)$
  - b_M
  - constraints: $b_E \rightarrow \neg b_M \wedge \neg b_J$; etc. (an item is given to at most one agent)
  - (and possibly): $b_E \vee b_M \vee b_J$ etc. (every item must be allocated)
- allocations satisfying a maximum number of agents: *model-preserving translation* to MAXSAT

  $[c|-|b]$ $[c|b|-]$ $+ s$ to anybody (or to nobody, if allowed)
  $[-|cs|b]$ $[b|cs|-]$

Dichotomous preferences

Dichotomous preferences for fair division

- $\mathcal{X} = \{o_1, \ldots, o_m\}$ set of items
- $A \subseteq \mathcal{X}$ set of acceptable bundles
- agent i partitions the set of bundles A into two sets: *acceptable* and *unacceptable* bundles
- $b_E \vee (c_E \wedge s_E)$: Emmanuel is happy with $\{b\}$, $\{c, s\}$, $\{b, s\}$, $\{b, c\}$ and $\{b, c, s\}$, and unhappy with $\{c\}$, $\{s\}$ and $\varnothing$
- each set of acceptable bundles A is representable by a propositional formula φ_E
- a set of acceptable bundles A is *monotonic* if for all $X \subseteq Y$, $X \in A$ implies $Y \in A$
- **Remark** A is monotonic iff φ_E is a positive formula (can be written with only $\wedge$, $\vee$, but with no $\neg$)
  - $b \wedge \neg c$ (agent allergic to the smell of coffee): nonmonotonic

Dichotomous preferences

- Emmanuel wants a beer, or else coffee with sugar: $b_E \vee (c_E \wedge s_E)$
- Marine wants a beer: b_M
- Jean-Luc wants a beer or a coffee: $b_J \vee c_J$

An allocation π is *envy-free* if every agent is at least happy with her share than with any other agent's share

- $\pi_1 = [b| - |c]$: Marine is envious of Emmanuel.
- $\pi_2 = [cs|b|-]$: Jean-Luc is envious of both Emmanuel and Marine.
- $\pi_3 = [-| - |c]$: envy-free, but not *Pareto-efficient*: $[b| - |c]$ does at least as well as π_3 for all agents and strictly better for one (Emmanuel).

Here: no allocation is both envy-free and Pareto-efficient

Dichotomous preferences

Preferences slightly change: Emmanuel does not like beer anymore.

- Emmanuel wants a coffee with sugar: $c_E \wedge s_E$
- Marine wants a beer: b_M
- Jean-Luc wants a beer or a coffee: $b_J \vee c_J$
- $[-|b|c]$ and $[s|b|c]$: envy-free and Pareto-efficient

Dichotomous preferences

- Emmanuel wants a coffee with sugar: $c_E \wedge s_E$
- Marine wants a beer: b_M
- Jean-Luc wants a beer or a coffee: $b_J \vee c_J$
- EF:

$$\begin{array}{ll} & (b_J \vee c_J) \vee (\neg(b_E \vee c_E) \wedge \neg(b_M \vee c_M)) \quad \text{Jean-Luc not envious} \\ \wedge & (c_E \wedge s_E) \vee (\neg(c_M \wedge s_M) \wedge \neg(c_J \wedge s_J)) \quad \text{Emmanuel not envious} \\ \wedge & b_M \vee (\neg b_E \wedge \neg b_J) \quad \text{Marine not envious} \end{array}$$

- Γ: an item should not be given to more than one person

$$c_E \rightarrow (\neg c_M \wedge \neg c_J) \wedge \ldots$$

- Pareto-efficiency: satisfy a maximal subset of

$$\Delta = \{b_J \vee c_J, c_E \wedge s_E, b_M\}$$

- EF-PE allocation $\leftrightarrow$ maximal subset of Δ consistent with $EF \wedge \Gamma$
- model-preserving translation into default logic.
- practical computation: use efficient off-the shelf *Answer Set Programming* solvers

Dichotomous preferences

- EXISTENCE OF AN EFFICIENT, ENVY-FREE ALLOCATION is Σ_2^P-complete.
- complexity can fall down if
  - the number of agents is fixed to a small number, especially 2.
  - the agents are assumed to have *identical preferences*.
  - the agents' goals are restricted to a *specific subclass of formulas*.
  - Pareto-efficiency is replaced by the weaker *completeness* condition

Dichotomous preferences

- EXISTENCE OF AN EFFICIENT, ENVY-FREE ALLOCATION is Σ_2^P-complete.
- complexity can fall down if
  - the number of agents is fixed to a small number, especially 2.
  - the agents are assumed to have *identical preferences*.
  - the agents' goal are restricted to a *specific subclass of formulas*.
  - Pareto-efficiency is replaced by the weaker *completeness* condition

Dichotomous preferences

Retsircting the agents' goal to a specific subclass of formulas.

- if agents' preferences are represented by *cubes* then EXISTENCE OF AN EFFICIENT, ENVY-FREE ALLOCATION is NP-complete.
- if agents' preferences are represented by *clauses* then EXISTENCE OF AN EFFICIENT, ENVY-FREE ALLOCATION under this restriction is in P.

Compact representation languages

- $O = \{o_1, \ldots, o_m\}$ 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

Compact representation languages

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.

Compact representation languages

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:  $\wedge \Big($($($  $\wedge$  $)$ $\vee$  $\Big)$

Compact representation languages

- $O = \{o_1, \ldots, o_m\}$ 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 2: (obvious) language for additive utility functions:

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

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

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

Compact representation languages

- $O = \{o_1, \ldots, o_m\}$ 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 $\mathcal{X}$.

Compact 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?

Compact 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) language L_1 is at least as succinct as language L_2 is any preference structure expressible in L_2 can be expressed in L_1 *without any exponential growth of size*.

Compact 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) language L_1 is at least as succinct as language L_2 is any preference structure expressible in L_2 can be expressed in L_1 *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 L?

Compact 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) language L_1 is at least as succinct as language L_2 is any preference structure expressible in L_2 can be expressed in L_1 *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 L?
- *Easiness of elicitation*
  Preference elicitation = interaction with a user, so as to acquire her preferences, encoded in a language L.
  Is it easy to construct protocols for eliciting the agent's preferences in L?

Compact representation languages: 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 $\mathcal{X}$.

L_{exp} is more expressive than L_{add}.

Compact representation languages: succinctness

Relative notion:

L_1 is at least as succinct as L_2 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:

- $L_{exp,add}$ = 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

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

Compact representation languages: computational complexity

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

Given an input Φ in the language L, ...

- 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.

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: $u : \{S \subseteq O, |S| \leq k\} \rightarrow \mathbb{R}$

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

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 $|S| = k$: $\begin{pmatrix} m \\ k \end{pmatrix}$ 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

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 $\mathcal{X}_1, \ldots, \mathcal{X}_k$ be a family of subsets of N such that $\bigcup_i \mathcal{X}_i = N$. u is GAI-decomposable with respect to $\mathcal{X}_1, \ldots, \mathcal{X}_k$ 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

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 $x \in 2^{PS}$,

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

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

Weighted logics

Example

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

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

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

Weighted logics

Example

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

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

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

Weighted logics

Example

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

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

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

Weighted logics

Example

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

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

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

Weighted logics

Example

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

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

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

Weighted logics: expressive power

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

- 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 logics: 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 logics: 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)

Outline

Fair division: taxonomy of problems

Computation and communication
- Communication
- Combinatorial spaces and compact representation

Centralized mechanisms
- Cardinal, additive preferences
- Cardinal, non-additive preferences
- Ordinal, separable preferences
- Ordinal, nonseparable preferences

Decentralized mechanisms

Centralized mechanisms, ordinal preferences

- each agent i has a private preference relation $\succ_i$ over 2^O.
- the usual communication dilemma:
  1. either i specifies $\succ_i$ *in extenso*: exponential communication cost.
  2. or domain restriction, the most common being *separability*.

Centralized mechanisms: ordinal, separable preferences

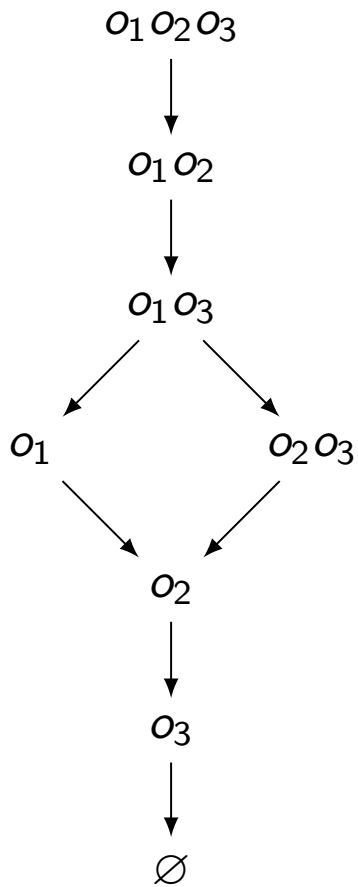
- each agent i reports a preference relation $\rhd_i$ over O
- $\rhd_i$ is extended to 2^O:
- $\succ_i$ is the *monotonic and separable extension* of $\rhd$ to 2^O, that is, the smallest preference relation $\succ$ over 2^O such that
  - $\succ$ extends $\rhd$: for all $o_i, o_j \in O$, $o_i \rhd o_j$ implies $\{o_i\} \succ \{o_j\}$
  - $\succ$ is separable
  - $\succ$ is monotonic
- $\succ$ sometimes called the *responsive extension* of $\rhd$.

Monotonic separable extension

$m = 2,\ o_1 \rhd o_2$

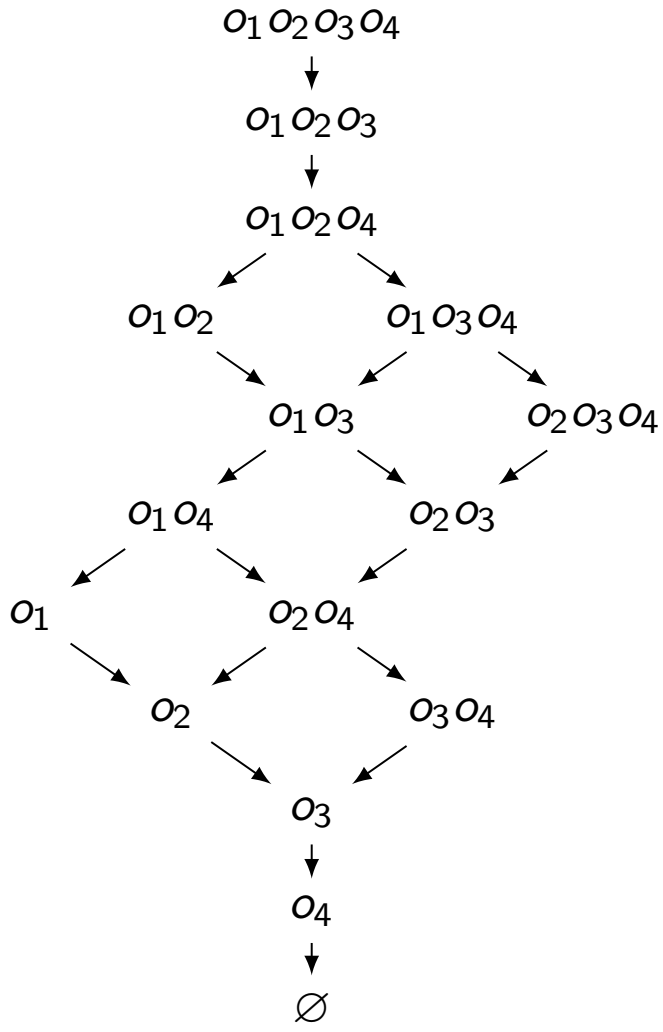
Monotonic separable extension

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



Monotonic separable extension

$m = 4,\ o_1 \rhd o_2 \rhd o_3 \rhd o_4$



Centralized mechanisms: ordinal, separable preferences

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?

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

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

- produces a (very) partial order

Modal notions of efficiency and fairness

- $P = \langle \rhd_1, \ldots, \rhd_n \rangle$ where each $\rhd_i$ is a ranking over O.
- $P^* = \langle \succ_1, \ldots, \succ_n \rangle$ where each $\succ_i$ is the monotonic separable extension of $\rhd_i$.
- completion of P^*:

$$T = \langle T_1, \ldots, T_n \rangle$$

  where each T_i is a linear order extending $\succ_i$.
- For a fairness or efficiency concept Γ:

Possible Γ	an allocation π is *possibly* Γ if there exists a completion T of P^* such that π satisfies Γ under T.
Necessary Γ	an allocation π is *necessarily* Γ if for every completion T of P^*, π satisfies Γ under T.

Modal notions of efficiency and fairness

- $P = (\rhd_1, \ldots, \rhd_n)$
  $\mapsto R = (\succ_1, \ldots, \succ_n)$ collection of partial orders on 2^O
- π *possibly envy-free*
  if it is envy-free for some complete extension of R
- π *necessarily envy-free*
  if it is envy-free for all complete extensions of R
- *possible Pareto efficiency*, *necessary Pareto efficiency*, etc: defined similarly.

Ranking single items

Simple characterization:

- π is necessarily envy-free if for all agents i, j, and all $k \leq |\pi_i|$, i prefers his kth best item in π_i to the kth best item in π_j
- π is possibly envy-free if for all i, j, either $|\pi_i| > |\pi_j|$ or for some $k \leq |\pi_i|$, i prefers his kth best item in π_i to the kth best item in π_j.

$$
\begin{aligned}
\rhd_1&: \quad \mathbf{a} \rhd b \rhd c \rhd d \rhd e \rhd f \\
\rhd_2&: \quad a \rhd \mathbf{d} \rhd b \rhd c \rhd e \rhd \mathbf{f} \\
\rhd_3&: \quad \mathbf{b} \rhd a \rhd c \rhd d \rhd f \rhd e \\
\rhd_4&: \quad b \rhd a \rhd \mathbf{c} \rhd \mathbf{e} \rhd f \rhd d
\end{aligned}
$$

possibly efficient, possibly envy-free
not necessarily envy-free
not necessarily efficient

Ranking single items

Simple characterization:

- π is necessarily envy-free if for all agents i, j, and all $k \leq |\pi_i|$, i prefers his kth best item in π_i to the kth best item in π_j
- π is possibly envy-free if for all i, j, either $|\pi_i| > |\pi_j|$ or for some $k \leq |\pi_i|$, i prefers his kth best item in π_i to the kth best item in π_j.

$$\begin{aligned} \rhd_1&: \quad a \rhd b \rhd c \rhd d \rhd e \\ \rhd_2&: \quad a \rhd d \rhd b \rhd c \rhd e \\ \rhd_3&: \quad b \rhd a \rhd c \rhd d \rhd e \\ \rhd_4&: \quad b \rhd a \rhd c \rhd e \rhd d \end{aligned}$$

- is there a complete, possibly envy-free (PEF) allocation?
- if some agent receives nothing, she necessarily envies all those receiving something
- PEF $\Rightarrow$ agent receives two items, three agents receive one item
- if 1 or 2 receive only one item, one of them will not get a and will necessarily envy someone; similarly for 3 and 4.
- if k distinct items are top-ranked by some agent, there exists a complete possibly envy-free allocation if and only if $m \geq 2n - k$. [Remains true if *complete* is replaced by *possibly Pareto-efficient*.]

Ranking single items

Simple characterization:

- π is necessarily envy-free if for all agents i, j, and all $k \leq |\pi_i|$, i prefers his kth best item in π_i to the kth best item in π_j
- π is possibly envy-free if for all i, j, either $|\pi_i| > |\pi_j|$ or for some $k \leq |\pi_i|$, i prefers his kth best item in π_i to the kth best item in π_j.

1 and 4 leave:

$$\rhd_2: \quad \mathbf{a} \rhd \mathbf{d} \rhd b \rhd c \rhd \mathbf{e} \rhd f$$
$$\rhd_3: \quad \mathbf{b} \rhd a \rhd \mathbf{c} \rhd d \rhd \mathbf{f} \rhd e$$

necessarily envy-free
not necessarily efficient

Modal notions of efficiency and fairness

The following problems are NP-complete:

- existence of a complete necessarily envy-free allocation.
- existence of a possibly Pareto-optimal, necessarily envy-free allocation.
- existence of a necessarily Pareto-optimal, necessarily envy-free allocation.

Particular case of two agents ($n = 2$):

- a complete allocation π is necessarily envy-free if and only if
  - m is even
  - for $i = 1, 2$ and every $k = 1, \ldots, \frac{m}{2}$, π gives agent i at least k of her $2k - 1$ most preferred items.
- a complete allocation necessarily envy-free, if there exists ine, can be computed in polynomial time.

Outline

Fair division: taxonomy of problems

Computation and communication
- Communication
- Combinatorial spaces and compact representation

Centralized mechanisms
- Cardinal, additive preferences
- Cardinal, non-additive preferences
- Ordinal, separable preferences
- Ordinal, nonseparable preferences

Decentralized mechanisms

Centralized mechanisms: ordinal, nonseparable preferences

- How can we express succinctly ordinal, nonseparable preferences over 2^O
- a first possibility: conditionally lexicographic preferences
- a second possibility: *conditional importance networks*

Conditional importance networks

- allow to express conditional importance statements such as

$$a\overline{b} : cde \rhd fg$$

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

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

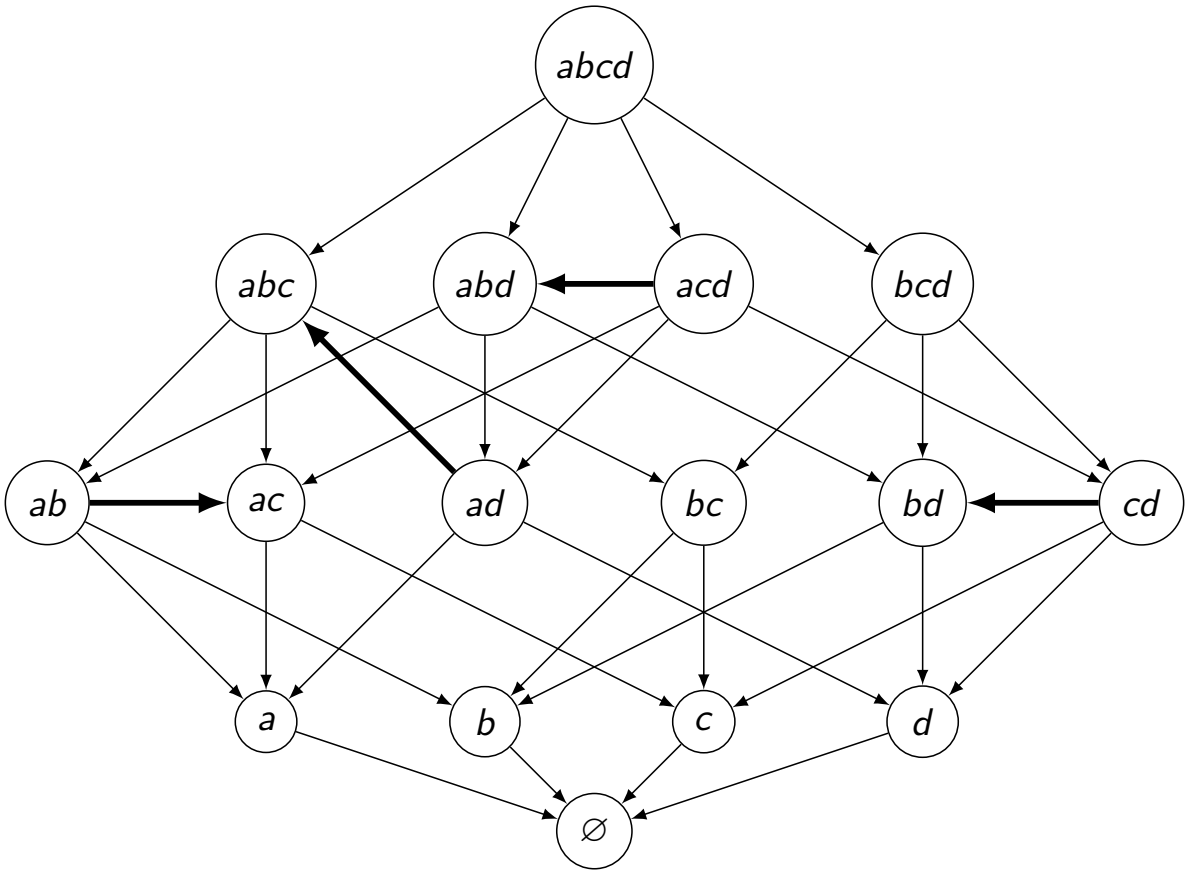
Preference relation induced from a CI-net

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

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

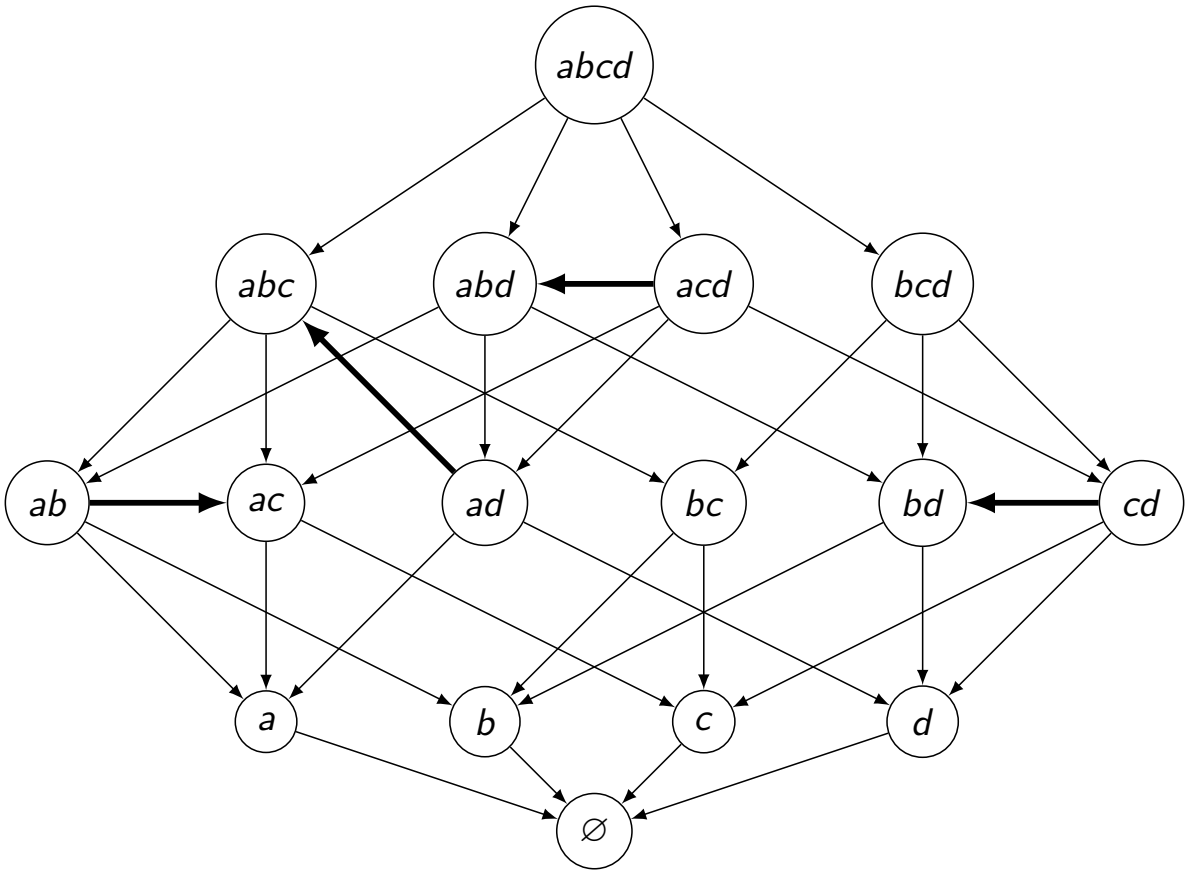
Conditional importance networks

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



Conditional importance networks

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



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

Conditional importance networks

- we recover the separable, monotonic extension when the CI-net is of the form

$$\varnothing, \varnothing : o_1 \rhd o_2$$
$$\varnothing, \varnothing : o_2 \rhd o_3;$$
$$\ldots$$
$$\varnothing, \varnothing : o_{m-1} \rhd o_m$$

- 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 $\{a \rhd c, b \rhd c, e \rhd d\}$).

Outline

Fair division: taxonomy of problems

Computation and communication
- Communication
- Combinatorial spaces and compact representation

Centralized mechanisms
- Cardinal, additive preferences
- Cardinal, non-additive preferences
- Ordinal, separable preferences
- Ordinal, nonseparable preferences

Decentralized mechanisms

Decentralized mechanisms

1. the Descending Demand Protocols
2. Picking Sequences Protocols
3. the Undercut Protocol
4. Local Exchange Protocols

The Descending Demand Protocols

- Herreiner and Puppe (2002)
- Basic protocol:
  - at each step k, each agents gives her best k subset of items
  - stop when we reach a k such that there exists an allocation π such that $\pi(i)$ is among the top k subsets of agent i.
- 2 agents (Ann and Bob), 4 items (a, b, c, d)

step	Ann	Bob
1	*abcd*	*abcd*
2	*bcd*	*abc*
3	*acd*	*abd*
4	*cd*	*bcd*
5	*abc*	*acd*
6	*abd*	*bd*
7	*bc*	**ac**
8	**bd**	*bc*

- stop and output $(bd|ac)$

The Descending Demand Protocols

- allows for nonseparable preferences
- resulting allocation maximizes egalitarian social welfare
- resulting allocation may not be envy-free, even when there exists one
- in $(bd|ac)$: Bob envies Ann.
- modified DDP:

step	Ann	Bob
1	*abcd*	*abcd*
2	*bcd*	*abc*
3	*acd*	*abd*
4	**cd**	*bcd*
5	*abc*	*acd*
6	*abd*	*bd*
7	*bc*	*ac*
8	*bd*	*bc*
9	*ab*	**ab**

The Descending Demand Protocols

Many nice properties but **exponential communication complexity**

- with two agents, in the worst case, each agent may have to report half of her preferences
- this worst case is likely to be typical, especially if agents have similar preferences.

Picking Sequences Protocols

- strict alternation protocol:
  - agent 1 takes an item
  - agent 2 takes one of the remaining items
  - ...
  - agent n takes one of the remaining items
  - agent 1 takes one of the remaining items
  - etc.
- example for 2 agents and 8 items: 12121212
- other sequences may be fairer, for instance 12212112
- what about 3 agents and 17 items?
- to each sequence σ: a picking sequence protocol P_σ.

Picking Sequences Protocols

- difficult for the (even nonstrategic) agents to pick the next item if they have nonseparable preferences
  - picking sequences 1221
  - agent 1: $o_1o_2 \succ o_3o_4 \succ o_1o_3 \succ o_2o_3 \succ o_2o_4 \succ o_1o_4$
  - agent 2: $o_1o_2 \succ o_1o_3 \succ o_1o_4 \succ o_2o_3 \succ o_2o_4 \succ o_3o_4$
  - what should she pick first?
  - the resulting allocation could be $(o_1o_4|o_2o_3)$, Pareto-dominated.
- when all agents have separable preferences:
  - the resulting allocation is Pareto-efficient
  - no guarantee of maxmin fairness or envy-freeness (even when there exists an envy-free efficient allocation)
- communication complexity at most $m \log m$
- more on picking sequences on Thursday.

The Undercut Protocol

- Brams, Kilgour and Klamler (2012)
- Two agents (Ann and Bob), 6 items.
- Ann: $e \succ a \succ b \succ f \succ c \succ d$
- Bob: $f \succ e \succ a \succ b \succ c \succ d$
- first phase:
  - agents name their best items
  - if they name different items, each gets the item she names
  - otherwise the item goes into the *contested pile CP*
  - Ann names and gets e, Bob names and gets f.
  - Ann and Bob both name a, which goes to the *contested pile CP*.
  - Similarly, b, c and d go to the contested pile: $CP = \{a, b, c, d\}$

The Undercut Protocol

- Ann: $e \succ a \succ b \succ f \succ c \succ d$
- Bob: $f \succ e \succ a \succ b \succ c \succ d$
- first phase: Ann gets e, Bob gets f, $CP = \{a, b, c, d\}$.
- second phase:
  - let $>_i$ be the monotonic separable extension of $\succ_i$
  - S is a *minimal bundle* for agent i if
    (a) $S \succeq_i CP \setminus S$, and
    (b) for every $T <_i S$, $CP \setminus T \succ_i T$.
  - each agent i communicates to the central authority, privately, her set of minimal bundles
- Ann:
  - Ann's preference relation on 2^{CP}:

    $abcd \succ abc \succ abd \succ acd \succ bcd \succ ab \succ ac \succ bc \succ ad \succ bd \succ cd \ldots$

  - Ann's only minimal bundle: bc
  - $bc \succ ad$
  - $bd \prec ac$, $cd \prec ab$, $b \prec acd$, $c \prec abd$, $d \prec abc$

The Undercut Protocol

- Ann: $e \succ a \succ b \succ f \succ c \succ d$
- Bob: $f \succ e \succ a \succ b \succ c \succ d$
- first phase: Ann gets e, Bob gets f, $CP = \{a, b, c, d\}$.
- second phase:
  - let $>_i$ be the monotonic separable extension of $\succ_i$
  - S is a *minimal bundle* for agent i if
    (a) $S \succeq_i CP \setminus S$, and
    (b) for every $T <_i S$, $CP \setminus T \succ_i T$.
  - each agent i communicates to the central authority, privately, her set of minimal bundles
- Bob:
  - Bob's preference relation on 2^{CP}:

    $abcd \succ abc \succ abd \succ ab \succ acd \succ ac \succ bcd \succ ad \succ a \succ bc \succ bd \ldots$

  - Bob's minimal bundles: bcd, ad

The Undercut Protocol

- Ann: $e \succ a \succ b \succ f \succ c \succ d$
- Bob: $f \succ e \succ a \succ b \succ c \succ d$
- Ann gets e, Bob gets f, $CP = \{a, b, c, d\}$.
- Ann's minimal bundles: $MB_A = \{bc\}$
- Bob's minimal bundles: $MB_B = \{bcd, ad\}$
- Case 1: $MB_A \neq MB_B$.
  - The central authority proposes (for instance) Bob's most preferred minimal bundles S in $MB_B \setminus MB_A$ to Ann.
  - It proposes bcd, to Ann.
  - Ann can choose to take $CP \setminus S$, or to *undercut* S by taking her most preferred bundle among those that are $>_A$-dominated by S
  - Ann's preference relation on 2^{CP}:

    $abcd \succ abc \succ abd \succ acd \succ bcd \succ ab \succ ac \succ bc \succ ad \succ bd \succ cd \ldots$

  - Ann can undercut bcd into bc
  - since Ann prefers bc to a, Ann takes bc and Bob receives ad
  - final allocation $[bce|adf]$.

The Undercut Protocol

- Ann: $e \succ a \succ b \succ f \succ c \succ d$
- Bob: $f \succ e \succ a \succ b \succ c \succ d$
- Ann gets e, Bob gets f, $CP = \{a, b, c, d\}$.
- Ann's minimal bundles: $MB_A = \{bc\}$
- Bob's minimal bundles: $MB_B = \{bcd, ad\}$
- Case 1: $MB_A \neq MB_B$.
- Case 2a: $MB_A = MB_B$, and there exists $S \in MB_A$ such that $S \sim_1 CP \setminus S$ (and equivalently, $S \sim_2 CP \setminus S$).
  - Ann receives S and Bob receives $CP \setminus S$ (or *vice versa*)

The Undercut Protocol

- Ann: $e \succ a \succ b \succ f \succ c \succ d$
- Bob: $f \succ e \succ a \succ b \succ c \succ d$
- Ann gets e, Bob gets f, $CP = \{a, b, c, d\}$.
- Ann's minimal bundles: $MB_A = \{bc\}$
- Bob's minimal bundles: $MB_B = \{bcd, ad\}$
- Case 1: $MB_A \neq MB_B$.
- Case 2a: $MB_A = MB_B$, and there exists $S \in MB_A$ such that $S \sim_1 CP \setminus S$.
- Case 2b: $MB_A = MB_B$, and no such S exists.
  - a minimal bundle S is chosen arbitrarily and Ann has to choose between accepting $CP \setminus S$ and undercutting S.
- in cases 1 and 2a, the resulting allocation is envy-free, provided the agent's preferences are separable.

The Undercut Protocol

- communication complexity is exponential
- but this is a worst-case bound, and in practice it will often be very small.

Protocols Based on Local Exchanges

- works with a prior allocation (initial endowment or default allocation or randomly selected allocation)
- the agents may contract local exchanges, respecting some rationality criteria
- possibility (or not) of side payments
- depending on several parameters (number of agents and items involved in an exchange, possibility of side payments, possible restrictions on preferences), the process may or may not converge towards a fair allocation, for some given fairness criteria.
- for instance: (Chevaleyre, Endriss, Estivie, Maudet, 2007): if utility functions are supermodular, then a Paretp-efficient envy-free allocation can always be reached.