Cutting matroids like cakes

Laurent Gourvès, Jérôme Monnot, Lydia Tlilane

LAMSADE, CNRS - PSL, Université Paris Dauphine

January 19, 2015

Outline

Outline

Allocation of a divisible resource

...also known as cake cutting

- n agents $\{1, \ldots, n\}$
- a divisible resource (a cake)
- A solution is a partition of the resource $P_1 \cup \ldots \cup P_n$ such that agent i receives P_i
- Agent i's utility for P_i is denoted by $u_i(P_i)$

Normalization assumption: every agent has a utility of 1 for the entire resource

Allocation of a divisible resource

...also known as cake cutting

- n agents $\{1, \ldots, n\}$
- a divisible resource (a cake)
- A solution is a partition of the resource $P_1 \cup \ldots \cup P_n$ such that agent i receives P_i
- Agent i's utility for P_i is denoted by $u_i(P_i)$

Normalization assumption: every agent has a utility of 1 for the entire resource

Two desirable properties

Proportionality : $u_i(P_i) \geq 1/n$ for all $i \in [n]$

Envy-freeness : $u_i(P_i) \geq u_i(P_j)$ for all $i, j \in [n]$

It is possible to find an envy-free (and thus proportional) solution

2 agents : `Divide-and-Choose`

One agent divides the resource into what she believes are equal halves, and the other agent chooses the “half” she prefers

Two desirable properties

Proportionality : $u_i(P_i) \geq 1/n$ for all $i \in [n]$

Envy-freeness : $u_i(P_i) \geq u_i(P_j)$ for all $i, j \in [n]$

It is possible to find an envy-free (and thus proportional) solution

2 agents : `Divide-and-Choose`

One agent divides the resource into what she believes are equal halves, and the other agent chooses the “half” she prefers

Bibliographic notes

- 1940’s : proportional protocol for any number of agents
- late 1950’s : envy-free protocol for 3 agents
- 1995 : envy-free protocol for any number of agents

Two desirable properties

Proportionality : $u_i(P_i) \geq 1/n$ for all $i \in [n]$

Envy-freeness : $u_i(P_i) \geq u_i(P_j)$ for all $i, j \in [n]$

It is possible to find an envy-free (and thus proportional) solution

2 agents : `Divide-and-Choose`

One agent divides the resource into what she believes are equal halves, and the other agent chooses the "half" she prefers

Bibliographic notes

- 1940's : proportional protocol for any number of agents
- late 1950's : envy-free protocol for 3 agents
- 1995 : envy-free protocol for any number of agents

Allocation of indivisible goods

- n agents $\{1, \ldots, n\}$
- m indivisible items $\{1, \ldots, m\}$
- $u_i(j)$, nonnegative utility of agent i for item j
- A solution/allocation is a partition of the set of items $A_1 \cup \ldots \cup A_n$ so that agent i receives A_i
- Agent i's utility for A_i is denoted by $u_i(A_i)$ and defined as $\sum_{j \in A_i} u_i(j)$

Normalization assumption: $\sum_{j=1}^{m} u_i(j) = 1$ for all $i \in [n]$

Allocation of indivisible goods

- n agents $\{1, \ldots, n\}$
- m indivisible items $\{1, \ldots, m\}$
- $u_i(j)$, nonnegative utility of agent i for item j
- A solution/allocation is a partition of the set of items $A_1 \cup \ldots \cup A_n$ so that agent i receives A_i
- Agent i's utility for A_i is denoted by $u_i(A_i)$ and defined as $\sum_{j \in A_i} u_i(j)$

Normalization assumption: $\sum_{j=1}^{m} u_i(j) = 1$ for all $i \in [n]$

Proportionality

Proportionality : $u_i(A_i) \geq 1/n$ for all $i \in [n]$

A proportional allocation is not guaranteed to exist

	item 1	item 2
agent 1	0.3	0.7
agent 2	0.4	0.6

The agent who does not receive item 2 has total utility $< \frac{1}{2}$

Hill's substitute

Ted Hill [The Annals of Probability, 1987] tried to characterize the value t_n such that, in any case, there exists an allocation A satisfying

$$u_i(A_i) \geq t_n \text{ for all } i \in [n]$$

The result relies on a parameter

$$\alpha := \max_{i\in[n], j\in[m]} u_i(j)$$

Hill's substitute

Ted Hill [The Annals of Probability, 1987] tried to characterize the value t_n such that, in any case, there exists an allocation A satisfying

$$u_i(A_i) \geq t_n \text{ for all } i \in [n]$$

The result relies on a parameter

$$\alpha := \max_{i \in [n], j \in [m]} u_i(j)$$

	item 1	item 2
agent 1	0.2	0.8
agent 2	0.1	0.9

$\alpha = 0.9$

There exists an allocation A such that $u_i(A_i) \geq 0.2$ for $i = 1, 2$

Hill's substitute

Ted Hill [The Annals of Probability, 1987] tried to characterize the value t_n such that, in any case, there exists an allocation A satisfying

$$u_i(A_i) \geq t_n \text{ for all } i \in [n]$$

The result relies on a parameter

$$\alpha := \max_{i \in [n], j \in [m]} u_i(j)$$

	item 1	item 2
agent 1	0.2	0.8
agent 2	0.1	0.9

$\alpha = 0.9$

There exists an allocation A such that $u_i(A_i) \geq 0.2$ for $i = 1, 2$

Hill's substitute

Ted Hill [The Annals of Probability, 1987] tried to characterize the value t_n such that, in any case, there exists an allocation A satisfying

$$u_i(A_i) \geq t_n \text{ for all } i \in [n]$$

The result relies on a parameter

$$\alpha := \max_{i \in [n], j \in [m]} u_i(j)$$

	item 1	item 2	item 3
agent 1	0.4	0.3	0.3
agent 2	0.4	0.5	0.1

$\alpha = 0.5$

There exists an allocation A such that $u_i(A_i) \geq 0.5$ for $i = 1, 2$

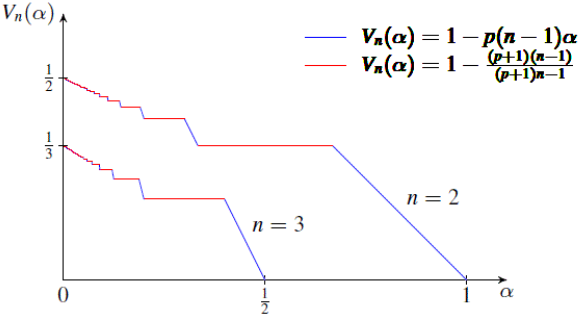
Hill's substitute

Hill's characterization is a non increasing function $V_n : [0,1] \to [0, n^{-1}]$

$$V_n(x) = \begin{cases} 1 - p(n-1)x, & x \in I(n,p) \\ 1 - \frac{(p+1)(n-1)}{(p+1)n-1}, & x \in NI(n,p) \end{cases}$$

where for any integer $p \geq 1$,
$I(n,p) = \left[\frac{p+1}{p((p+1)n-1)}, \frac{1}{pn-1}\right]$ and
$NI(n,p) = \left]\frac{1}{(p+1)n-1}, \frac{p+1}{p((p+1)n-1)}\right[$

Hill's substitute



If there are n agents and $\alpha := \max_{i \in [n], j \in [m]} u_i(j)$ then there must exist an allocation A such that $u_i(A_i) \geq V_n(\alpha)$ for all $i \in [n]$

An algorithmic approach

Markakis & Psomas [WINE 2011] proposed a centralized algorithm called ALLOCATE, polynomial in (n, m), that builds an allocation A such that $u_i(A_i) \geq V_n(\alpha)$

Moreover, the allocation is such that $u_i(A_i) \geq V_n(\alpha_i) \geq V_n(\alpha)$ where

$$\alpha_i = \max_{j \in [m]} u_i(j)$$

and

$$\alpha := \max_{i \in [n], j \in [m]} u_i(j) = \max_{i \in [n]} \alpha_i$$

Note that the utility of the least happy agent remains the same

An algorithmic approach

Markakis & Psomas [WINE 2011] proposed a centralized algorithm called ALLOCATE, polynomial in (n, m), that builds an allocation A such that $u_i(A_i) \geq V_n(\alpha)$

Moreover, the allocation is such that $u_i(A_i) \geq V_n(\alpha_i) \geq V_n(\alpha)$ where

$$\alpha_i = \max_{j \in [m]} u_i(j)$$

and

$$\alpha := \max_{i \in [n], j \in [m]} u_i(j) = \max_{i \in [n]} \alpha_i$$

Note that the utility of the least happy agent remains the same

Centralized versus decentralized algorithms

Centralized

- A third party gathers the utilities of the agents for the different portions and computes an allocation
- An agent is often reluctant to disclose his utilities (is the third party trustworthy?)
- Issue with the communication of the utilities for every possible portion

Decentralized (Mechanism design)

- There is no third party to compute the allocation; each agent takes part of the construction of the final solution
- The communication problem is avoided

Centralized versus decentralized algorithms

Centralized

- A third party gathers the utilities of the agents for the different portions and computes an allocation
- An agent is often reluctant to disclose his utilities (is the third party trustworthy?)
- Issue with the communication of the utilities for every possible portion

Decentralized (Mechanism design)

- There is no third party to compute the allocation; each agent takes part of the construction of the final solution
- The communication problem is avoided

Outline

Matroids

A matroid $\mathcal{M} = (X, \mathcal{F})$ consists of a finite set of n elements X and a collection $\mathcal{F}$ of subsets of X such that:

(i) $\emptyset \in \mathcal{F}$,

(ii) if $F_2 \subseteq F_1$ and $F_1 \in \mathcal{F}$ then $F_2 \in \mathcal{F}$,

(iii) for every couple $F_1, F_2 \in \mathcal{F}$ such that $|F_1| < |F_2|$, $\exists\, e \in F_2 \setminus F_1$ such that $F_1 \cup \{e\} \in \mathcal{F}$.

Every $F \subseteq X$ s.t. $F \in \mathcal{F}$ is said independent

Every $C \subseteq X$ s.t. $C \notin \mathcal{F}$ is said dependent

A base B is an independent of maximal size

Matroids

A matroid $\mathcal{M} = (X, \mathcal{F})$ consists of a finite set of n elements X and a collection $\mathcal{F}$ of subsets of X such that:

(i) $\emptyset \in \mathcal{F}$,

(ii) if $F_2 \subseteq F_1$ and $F_1 \in \mathcal{F}$ then $F_2 \in \mathcal{F}$,

(iii) for every couple $F_1, F_2 \in \mathcal{F}$ such that $|F_1| < |F_2|$, $\exists\ e \in F_2 \backslash F_1$ such that $F_1 \cup \{e\} \in \mathcal{F}$.

Every $F \subseteq X$ s.t. $F \in \mathcal{F}$ is said independent

Every $C \subseteq X$ s.t. $C \notin \mathcal{F}$ is said dependent

A base B is an independent of maximal size

Example 1 : Forests

A forest in a graph G is a subset of its edges *without* any cycle

We can define a matroid $\mathcal{M} = (X, \mathcal{F})$ over a graph G

- X is the edges of G
- $\mathcal{F}$ is the set of all forests of G

A base is a spanning tree

Example 2 : Allocation of indivisible goods

n agents, m indivisible items

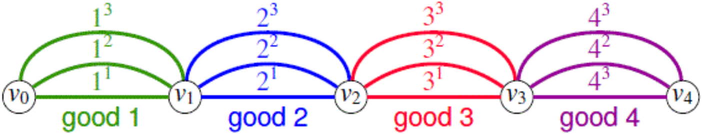
For every item j, build the set $E_j = \{j^1, j^2, \dots, j^n\}$

We can define a matroid $\mathcal{M} = (X, \mathcal{F})$ as follows

- $X = \bigcup_{j=1}^{m} E_j$
- $\mathcal{F} = \{F \subseteq X : |F \cap E_j| \leq 1, j \in [m]\}$

Example 2 : Allocation of indivisible goods

3 agents, 4 goods...



An allocation is a spanning tree in this graph

Taking edge j^i means allocating item j to agent i

Example 3 : Seminar

- m speakers $\{1, \dots, m\}$
- ℓ days $\{d_1, \dots, d_\ell\}$

A speaker j is available for a subset of $\{d_1, \dots, d_\ell\}$

At most one speaker per day

Find a subset of speakers that we can assign to $\{d_1, \dots, d_\ell\}$

	d_1	d_2	d_3
1	1	1	0
2	0	1	0
3	1	0	1
4	0	1	0
5	1	0	1

One can define a matroid $(X, \mathcal{F})$

- $X = \{1, \dots, m\}$
- $F \in \mathcal{F}$ if there exists $f : \{1, \dots, m\} \to \{d_1, \dots, d_\ell\}$, every $j \in F$ is available on day $f(j)$, and $f(j) \neq f(j')$ for all $(j, j') \in F^2$

Example 3 : Seminar

- m speakers $\{1, \dots, m\}$
- ℓ days $\{d_1, \dots, d_\ell\}$

A speaker j is available for a subset of $\{d_1, \dots, d_\ell\}$

At most one speaker per day

Find a subset of speakers that we can assign to $\{d_1, \dots, d_\ell\}$

	d_1	d_2	d_3
1	1	1	0
2	0	1	0
3	1	0	1
4	0	1	0
5	1	0	1

One can define a matroid $(X, \mathcal{F})$

- $X = \{1, \dots, m\}$
- $F \in \mathcal{F}$ if there exists $f : \{1, \dots, m\} \to \{d_1, \dots, d_\ell\}$, every $j \in F$ is available on day $f(j)$, and $f(j) \neq f(j')$ for all $(j, j') \in F^2$

Matroids' background

Introduced by H. Whitney (1935), a matroid is a structure that generalizes the notion of linear independence in vector spaces

Many applications in combinatorial optimization:

- spanning trees
- assignment problem

Polynomial algorithms

- greedy: independent of maximum weight in a matroid
- maximum weight independent in the intersection of 2 matroids defined on the same set of elements

Matroids' background

Introduced by H. Whitney (1935), a matroid is a structure that generalizes the notion of linear independence in vector spaces

Many applications in combinatorial optimization:

- spanning trees
- assignment problem

Polynomial algorithms

- greedy: independent of maximum weight in a matroid
- maximum weight independent in the intersection of 2 matroids defined on the same set of elements

See *J. Oxley, Matroid Theory, Oxford University Press, 1992.* for more details

Matroids' background

Introduced by H. Whitney (1935), a matroid is a structure that generalizes the notion of linear independence in vector spaces

Many applications in combinatorial optimization:

- spanning trees
- assignment problem

Polynomial algorithms

- greedy: independent of maximum weight in a matroid
- maximum weight independent in the intersection of 2 matroids defined on the same set of elements

See *J. Oxley, Matroid Theory, Oxford University Press, 1992.* for more details

Outline

The setting

Input

- a matroid $\mathcal{M} = (X, \mathcal{F})$
- n agents
- a utility function $u_i : X \to \mathbb{R}^+$ for every agent i

Additive utility: Agent i has utility $u_i(F) := \sum_{x \in F} u_i(x)$ for every set $F \in \mathcal{F}$

The setting

Input

- a matroid $\mathcal{M} = (X, \mathcal{F})$
- n agents
- a utility function $u_i : X \to \mathbb{R}^+$ for every agent i

Additive utility: Agent i has utility $u_i(F) := \sum_{x \in F} u_i(x)$ for every set $F \in \mathcal{F}$

Goal : Build a "collective solution" $F \in \mathcal{F}$ and consider $(u_i(F))_{i \in [n]}$, the profile of the agents' utilities

The setting

Input

- a matroid $\mathcal{M} = (X, \mathcal{F})$
- n agents
- a utility function $u_i : X \to \mathbb{R}^+$ for every agent i

Additive utility: Agent i has utility $u_i(F) := \sum_{x \in F} u_i(x)$ for every set $F \in \mathcal{F}$

Goal : Build a “collective solution” $F \in \mathcal{F}$ and consider $(u_i(F))_{i \in [n]}$, the profile of the agents' utilities

The setting

Remark

Since the utility for every element is nonnegative, one can restrict ourselves to the **bases** of $\mathcal{M}$

Normalization to 1

For every agent i and $F \in \mathcal{F}$ we have $0 \leq u_i(F) \leq 1$

$u_i(B_i^*) = 1$ where $B_i^* \in \mathcal{F}$ is the base that agent i likes the most

No loss of generality (rescaling)

The setting

Remark

Since the utility for every element is nonnegative, one can restrict ourselves to the **bases** of $\mathcal{M}$

Normalization to 1

For every agent i and $F \in \mathcal{F}$ we have $0 \leq u_i(F) \leq 1$

$u_i(B_i^*) = 1$ where $B_i^* \in \mathcal{F}$ is the base that agent i likes the most

No loss of generality (rescaling)

Example : Seminar

- n agents $\{a_1, \ldots, a_n\}$
- m speakers $\{1, \ldots, m\}$
- ℓ days $\{d_1, \ldots, d_\ell\}$

A speaker is available for a subset of $\{d_1, \ldots, d_\ell\}$, at most one speaker per day, find a subset of speakers that we can assign to $\{d_1, \ldots, d_\ell\}$

a_1	a_2
0.3	0.1
0.4	0.3
0.1	0.2
0.3	0.4
0.3	0.4

	d_1	d_2	d_3
1	1	1	0
2	0	1	0
3	1	0	1
4	0	1	0
5	1	0	1

$n = 2$

Example : Seminar

- n agents $\{a_1, \ldots, a_n\}$
- m speakers $\{1, \ldots, m\}$
- ℓ days $\{d_1, \ldots, d_\ell\}$

A speaker is available for a subset of $\{d_1, \ldots, d_\ell\}$, at most one speaker per day, find a subset of speakers that we can assign to $\{d_1, \ldots, d_\ell\}$

a_1	a_2
0.3	0.1
0.4	0.3
0.1	0.2
0.3	0.4
0.3	0.4

	d_1	d_2	d_3
1	1	1	0
2	0	1	0
3	1	0	1
4	0	1	0
5	1	0	1

$n = 2$

Example : Seminar

- n agents $\{a_1, \ldots, a_n\}$
- m speakers $\{1, \ldots, m\}$
- ℓ days $\{d_1, \ldots, d_\ell\}$

A speaker is available for a subset of $\{d_1, \ldots, d_\ell\}$, at most one speaker per day, find a subset of speakers that we can assign to $\{d_1, \ldots, d_\ell\}$

a_1	a_2
0.3	0.1
0.4	0.3
0.1	0.2
0.3	0.4
0.3	0.4

	d_1	d_2	d_3
1	1	1	0
2	0	1	0
3	1	0	1
4	0	1	0
5	1	0	1

$n = 2$

Example : Seminar

- n agents $\{a_1, \ldots, a_n\}$
- m speakers $\{1, \ldots, m\}$
- ℓ days $\{d_1, \ldots, d_\ell\}$

A speaker is available for a subset of $\{d_1, \ldots, d_\ell\}$, at most one speaker per day, find a subset of speakers that we can assign to $\{d_1, \ldots, d_\ell\}$

a_1	a_2
0.3	0.1
0.4	0.3
0.1	0.2
0.3	0.4
0.3	0.4

	d_1	d_2	d_3
1	1	1	0
2	0	1	0
3	1	0	1
4	0	1	0
5	1	0	1

$n = 2$

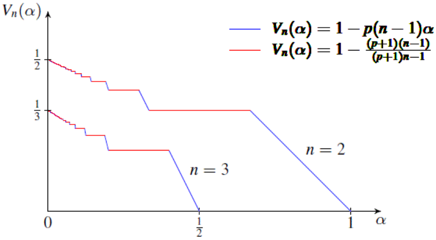
Example : Seminar

maximal feasible sets	u_1	u_2	$\min\{u_1, u_2\}$
$\{1, 2, 3\}$	0.8	0.6	0.6
$\{1, 2, 5\}$	1	0.8	0.8
$\{1, 3, 4\}$	0.7	0.7	0.7
$\{1, 3, 5\}$	0.7	0.7	0.7
$\{1, 4, 5\}$	0.9	0.9	0.9
$\{2, 3, 5\}$	0.8	0.9	0.8
$\{3, 4, 5\}$	0.7	1	0.7

Centralized algorithm

One can extend ALLOCATE, the centralized algorithm by Markakis & Psomas, to matroids

For every number of agents n, and every matroid $\mathcal{M}$, there exists a centralized algorithm which outputs a base B of $\mathcal{M}$ such that $u_i(B) \geq V_n(\alpha_i)$ for all agent i.



G, Monnot, Tlilane IJCAI 2013

Outline

Divide-and-Choose on a matroid (2 agents)

The input is a matroid $(X, \mathcal{F})$

1. Agent 1 computes a base and cuts it in two parts $S_1 \cup T_1$
2. Agent 2 finds S_2 and T_2 s.t. $S_1 \cup S_2$ and $T_1 \cup T_2$ are bases
3. Agent 2 chooses the base that she prefers between $S_1 \cup S_2$ and $T_1 \cup T_2$

The viewpoint of Agent 1

Lemma

Given a base B of a matroid partitioned in $S \cup T$ and another base B^* with maximal utility, one can always partition B^* in $S^* \cup T^*$ such that $\min\{u(S^*), u(T^*)\} \geq \min\{u(S), u(T)\}$

Hence, in the DaC mechanism, the first agent should compute a base which maximizes her utility (greedy algorithm) and partition it

Partitioning such that the lightest part is maximized is an **NP**-hard problem (SUBSET-SUM); but one can use ALLOCATE which is polynomial

The viewpoint of Agent 1

Lemma

Given a base B of a matroid partitioned in $S \cup T$ and another base B^* with maximal utility, one can always partition B^* in $S^* \cup T^*$ such that $\min\{u(S^*), u(T^*)\} \geq \min\{u(S), u(T)\}$

Hence, in the `DaC` mechanism, the first agent should compute a base which maximizes her utility (greedy algorithm) and partition it

Partitioning such that the lightest part is maximized is an **NP**-hard problem (SUBSET-SUM); but one can use ALLOCATE which is polynomial

By doing so the first agent is guaranteed to have a utility of $V_2(\alpha_1)$

The viewpoint of Agent 1

Lemma

Given a base B of a matroid partitioned in $S \cup T$ and another base B^* with maximal utility, one can always partition B^* in $S^* \cup T^*$ such that $\min\{u(S^*), u(T^*)\} \geq \min\{u(S), u(T)\}$

Hence, in the `DaC` mechanism, the first agent should compute a base which maximizes her utility (greedy algorithm) and partition it

Partitioning such that the lightest part is maximized is an **NP**-hard problem (SUBSET-SUM); but one can use ALLOCATE which is polynomial

By doing so the first agent is guaranteed to have a utility of $V_2(\alpha_1)$

Divide-and-Choose on a matroid (2 agents)

The input is a matroid $(X, \mathcal{F})$

1. Agent 1 computes a base and cut its in two parts $S_1 \cup T_1$
2. Agent 2 finds S_2 and T_2 s.t. $S_1 \cup S_2$ and $T_1 \cup T_2$ are bases
3. Agent 2 chooses the base that she prefers between $S_1 \cup S_2$ and $T_1 \cup T_2$

The viewpoint of Agent 2

Theorem [Brylawski '73, Greene '73, Woodall '74]

Given two bases B_1 and B_2 of a matroid $\mathcal{M}$, and a partition $B_1 = X_1 \cup Y_1$, there is a partition $B_2 = X_2 \cup Y_2$ such that $X_1 \cup Y_2$ and $X_2 \cup Y_1$ are two bases of $\mathcal{M}$

Hence, in the DaC mechanism, the second agent can compute a base B_2^* which maximizes her utility and find the partition $S_2 \cup T_2$ of it such that $S_1 \cup S_2$ and $T_1 \cup T_2$ are both bases

- the first step is polynomial (greedy algorithm)
- the second step is also polynomial (matroid intersection)

The viewpoint of Agent 2

Theorem [Brylawski '73, Greene '73, Woodall '74]

Given two bases B_1 and B_2 of a matroid $\mathcal{M}$, and a partition $B_1 = X_1 \cup Y_1$, there is a partition $B_2 = X_2 \cup Y_2$ such that $X_1 \cup Y_2$ and $X_2 \cup Y_1$ are two bases of $\mathcal{M}$

Hence, in the `DaC` mechanism, the second agent can compute a base B_2^* which maximizes her utility and find the partition $S_2 \cup T_2$ of it such that $S_1 \cup S_2$ and $T_1 \cup T_2$ are both bases

- the first step is polynomial (greedy algorithm)
- the second step is also polynomial (matroid intersection)

By doing so the second agent is guaranteed to have a utility of $\max\{u_2(S_1 \cup S_2), u_2(T_1 \cup T_2)\} \geq \max\{u_2(S_2), u_2(T_2)\} \geq u(B_2^*)/2 = 1/2 > V_2(\alpha_2)$

The viewpoint of Agent 2

Theorem [Brylawski '73, Greene '73, Woodall '74]

Given two bases B_1 and B_2 of a matroid $\mathcal{M}$, and a partition $B_1 = X_1 \cup Y_1$, there is a partition $B_2 = X_2 \cup Y_2$ such that $X_1 \cup Y_2$ and $X_2 \cup Y_1$ are two bases of $\mathcal{M}$

Hence, in the DaC mechanism, the second agent can compute a base B_2^* which maximizes her utility and find the partition $S_2 \cup T_2$ of it such that $S_1 \cup S_2$ and $T_1 \cup T_2$ are both bases

- the first step is polynomial (greedy algorithm)
- the second step is also polynomial (matroid intersection)

By doing so the second agent is guaranteed to have a utility of $\max\{u_2(S_1 \cup S_2), u_2(T_1 \cup T_2)\} \geq \max\{u_2(S_2), u_2(T_2)\} \geq u(B_2^*)/2 = 1/2 > V_2(\alpha_2)$

Summary

With the matroid extension of Divide-and-Choose, Agent 1 and 2 can guarantee to themselves $V_2(\alpha_1)$ and $0.5 > V_2(\alpha_2)$, respectively.

1	2
0.3	0.1
0.4	0.3
0.1	0.2
0.3	0.4
0.3	0.4

	d_1	d_2	d_3
1	1	1	0
2	0	1	0
3	1	0	1
4	0	1	0
5	1	0	1

Summary

With the matroid extension of Divide-and-Choose, Agent 1 and 2 can guarantee to themselves $V_2(\alpha_1)$ and $0.5 > V_2(\alpha_2)$, respectively.

1	2
0.3	0.1
0.4	0.3
0.1	0.2
0.3	0.4
0.3	0.4

	d_1	d_2	d_3
1	1	1	0
2	0	1	0
3	1	0	1
4	0	1	0
5	1	0	1

Agent 1 proposes $\{\{1,5\},\{2\}\}$. Agent 2 can partition his base $\{3,4,5\}$ into $\{\{4\},\{3,5\}\}$ such that $\{1,4,5\}$ and $\{2,3,5\}$ are two bases. Agent 2 likes them equally; suppose $\{1,4,5\}$ is returned. The two agents have utility 0.9

Summary

With the matroid extension of Divide-and-Choose, Agent 1 and 2 can guarantee to themselves $V_2(\alpha_1)$ and $0.5 > V_2(\alpha_2)$, respectively.

1	2
0.3	0.1
0.4	0.3
0.1	0.2
0.3	0.4
0.3	0.4

	d_1	d_2	d_3
1	1	1	0
2	0	1	0
3	1	0	1
4	0	1	0
5	1	0	1

Agent 1 proposes $\{\{1,5\},\{2\}\}$. Agent 2 can partition his base $\{3,4,5\}$ into $\{\{4\},\{3,5\}\}$ such that $\{1,4,5\}$ and $\{2,3,5\}$ are two bases. Agent 2 likes them equally; suppose $\{1,4,5\}$ is returned. The two agents have utility 0.9

Outline

A protocol for 3 agents on a matroid

1. Agent 1 computes a base and cuts it in 3 parts $A_1 \cup A_2 \cup A_3$
2. Agent 2 chooses one part, say A_i, and asks Agent 3 if he agrees to "give" this part to Agent 1
3. If Agent 3 agrees then A_i is in the final solution and Agent 2 and 3 apply Divide-and-Choose on the contraction of $\mathcal{M}$ by A_i
4. Otherwise, Agents 1 and 2 apply Divide-and-Choose on the contraction of $\mathcal{M}$ by A_i; R denotes the resulting independent set
5. Agent 3 completes R into a base of $\mathcal{M}$

Theorem

Using Divide-Ask-and-Choose, Agents 1, 2 and 3 can guarantee to themselves $V_3(\alpha_1)$, $\frac{2}{3}V_2(\alpha_2) \geq V_3(\alpha_2)$ and $1/3 \geq V_3(\alpha_3)$, respectively.

This result partially relies on the following lemma

Multiple exchange property (Greene & Magnanti 1975)

Let A and B be bases of a matroid $\mathcal{M}$ such that A is partitioned into $\{A_1, \ldots, A_n\}$. Then there exists a partition of B into $\{B_1, \ldots, B_n\}$ such that $(A - A_i) \cup B_i$ is a base for all $i \in [n]$.

Theorem

Using Divide-Ask-and-Choose, Agents 1, 2 and 3 can guarantee to themselves $V_3(\alpha_1)$, $\frac{2}{3}V_2(\alpha_2) \geq V_3(\alpha_2)$ and $1/3 \geq V_3(\alpha_3)$, respectively.

This result partially relies on the following lemma

Multiple exchange property (Greene & Magnanti 1975)

Let A and B be bases of a matroid $\mathcal{M}$ such that A is partitioned into $\{A_1, \ldots, A_n\}$. Then there exists a partition of B into $\{B_1, \ldots, B_n\}$ such that $(A - A_i) \cup B_i$ is a base for all $i \in [n]$.

Outline

The fair allocation of (in)divisible goods is a challenging problem with a lot of applications

We work with matroids, a structure that extends the allocation of indivisible goods

The fair allocation of (in)divisible goods is a challenging problem with a lot of applications

We work with matroids, a structure that extends the allocation of indivisible goods

Known results for the worst case utility of the agents in the problem of allocating indivisible goods apply to matroids

- centralized (any number of agents)
- decentralized (limited number of agents, $n \leq 8$)

The fair allocation of (in)divisible goods is a challenging problem with a lot of applications

We work with matroids, a structure that extends the allocation of indivisible goods

Known results for the worst case utility of the agents in the problem of allocating indivisible goods apply to matroids

- centralized (any number of agents)
- decentralized (limited number of agents, $n \leq 8$)

THANK YOU FOR YOUR ATTENTION