APPLIED MECHANISM DESIGN FOR SOCIAL GOOD

JOHN P DICKERSON

Lecture #15 – 10/18/2016

CMSC828M
Tuesdays & Thursdays
12:30pm – 1:45pm
THIS CLASS:

DYNAMIC FAIR DIVISION & ALLOCATING FOOD TO FOOD BANKS

PART I: JOHN DICKERSON
PART II: NIDHI SHAH

Thanks to: Nisarg Shah (NS), Nick Mattei (NM)
TODAY’S PROBLEM

Like most lectures in this class:

• $m$ items (initially divisible, later indivisible)

• $k$ agents with private values for bundles of items

Either the agents, the items, or both arrive over time.

This class:

• Start with fair allocation of multiple divisible resources in a dynamic setting [Kash Procaccia Shah JAIR-2014]

• Move to fair dynamic allocation of indivisible items via a restricted bidding language [Aleksandrov et al. IJCAI-2015]

• Wrap up with a richer bidding language based on funny money [Prendergast w.p. 2015]
ALLOCATING OF DIVISIBLE RESOURCES WITHOUT MONEY

Allocating computational resources (CPU, RAM, HDD, etc)

- Organizational clusters (e.g., our new Horvitz cluster)
- Federated clouds
- NSF Supercomputing Centers

We’ll focus on fixed bundles (slots)

- Allocated using single resource abstraction

Highly inefficient when users have heterogeneous demands
DOMINANT RESOURCE FAIRNESS (DRF) MECHANISMS

Idea: Assume structure on user demands

Proportional demands (a.k.a. Leontief preferences)

\[ u(x_1, \ldots, x_m) = \min \left\{ \frac{x_1}{w_1}, \ldots, \frac{x_m}{w_m} \right\} \]

Example:

- User wishes to execute multiple instances of a job
- Each instance needs (1 unit RAM, 2 units CPU)
- Indifferent between (2, 4) and (2, 5)
- Happier with (2.1, 4.2)
STATIC DRF MECHANISM
Dominant Resource Fairness = equalize largest shares
(a.k.a. dominant shares)
PROBLEM WITH DRF
[Kash Procaccia Shah JAIR-14]

Assumes all agents are present from the beginning and all the job information is known upfront

Can relax this to dynamic setting:

• Agents arriving over time
• Job information of an agent only revealed upon arrival

This paper initiated the study of dynamic fair division

• Huge literature on fair division, but mostly static settings
• Still very little work on fair division in dynamic environments!
FOCAL DYNAMIC MODEL

Resources are known beforehand
Agents arrive at different times (steps), do not depart
  • Total number of agents known in advance
Agents’ demands are proportional, revealed at arrival
  • Each agent requires every resource

Simple dynamic allocation mechanism:
  • At every step $k$
    • Input: $k$ reported demands
    • Output: An allocation over the $k$ present agents
  • Terminate after final agent arrives

Irrevocability of resources!
## DESIDERATA

Properties of DRF, aims for a dynamic generalization

<table>
<thead>
<tr>
<th>Property</th>
<th>Static (DRF)</th>
<th>Dynamic (Desired)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Envy freeness</td>
<td>EF: No swaps.</td>
<td>EF: No swaps at any step.</td>
</tr>
<tr>
<td>Sharing incentives</td>
<td>SI: At least as good as equal split.</td>
<td>SI: At least as good as equal split to every present agent at all steps.</td>
</tr>
<tr>
<td>Strategyproofness</td>
<td>SP: No gains by misreporting.</td>
<td>SP: No gains at any step by misreporting.</td>
</tr>
<tr>
<td>Pareto optimality</td>
<td>PO: No “better” allocation.</td>
<td>DPO: At any step $k$, no “better” allocation using $k/n$ share of each resource.</td>
</tr>
</tbody>
</table>
IMPOSSIBILITY RESULT

Envy freeness + Dynamic Pareto optimality = Impossible

- DPO requires allocating too much
- Later agents might envy earlier agents

Dropping either of them completely \(\rightarrow\) trivial mechanisms!

- Drop EF, trivial DPO mechanism ???????????
- Drop DPO, trivial EF mechanism ???????????

Relax one at a time …
1) RELAXING ENVY FREENESS

Envy impossible to avoid if efficiency (DPO) required

• But unfair if an agent is allocated resources while being envied

Dynamic Envy Freeness (DEF)

• If agent $i$ envies agent $j$, then $j$ must have arrived before $i$ did, and must not have been allocated any resources since $i$ arrived

Comparison to Forward EF [Walsh ADT-11]: An agent may only envy agents that arrived after her

• Forward EF is strictly weaker
• Trivial FEF mechanism ?????????????????
MECHANISM: DYNAMIC-DRF

1. Agent \( k \) arrives
2. Start with (previous) allocation of step \( k-1 \)
3. Keep allocating to all agents having the minimum “dominant” (largest) share at the same rate
   - Until a \( k/n \) fraction of at least one resource is allocated

(A constrained “water-filling” algorithm.)

Dynamic-DRF satisfies relaxed envy freeness (DEF) along with the other properties (DPO, SI, SP).
DYNAMIC-DRF ILLUSTRATED

3 agents, 2 resources

Total

(1,ε)

(ε,1)

(1,ε)
2) RELAXING DPO

Sometimes total fairness desired

Naïve approach: Wait for all the agents to arrive and then do a static envy free and Pareto optimal allocation

• Can we allocate more resources early?

Cautious Dynamic Pareto Optimality (CDPO)

• At every step, allocate as much as possible while ensuring EF can be achieved in the end irrespective of the future demands

• Cautious-LP: a constrained water-filling mechanism

Cautious-LP satisfies relaxed dynamic Pareto optimality (CDPO) along with the other properties (EF, SI, SP).
EXPERIMENTAL EVALUATION

Initial static DRF paper has had a big effect in industry.

Now: Dynamic-DRF and Cautious-LP under two objectives:

- Maximize the sum of dominant shares (utilitarian, maxsum)
- Maximize the minimum dominant share (egalitarian, maxmin)

Comparison with provable lower and upper bounds

Data: traces of real workloads on a Google compute cell

- 7-hour period in 2011, 2 resources (CPU and RAM)
- code.google.com/p/googleclusterdata/wiki/ClusterData2011_1
EXPERIMENTAL RESULTS

Maxsum objective
100 agents

Upper Bound
CautiousLP
DynamicDRF
Lower Bound

Maxmin objective
100 agents

Upper Bound
CautiousLP
DynamicDRF
Lower Bound
DISCUSSION

Relaxation: allowing zero demands

- Trivial mechanisms for SI+DPO+SP no longer work
- Open question: possibility of SI+DPO+SP in this case

Allowing agent departures and revocability of resources

- No re-arrivals $\rightarrow$ same mechanism (water-filling) for freed resources
- Departures with re-arrivals
  - Pareto optimality requires allocating resources freed on a departure
  - Need to revoke when the departed agent re-arrives
WHAT ABOUT INDIVISIBLE ITEMS?

[Aleksandrov et al. IJCAI-2015]

Recall: even in the static setting, an envy-free allocation may not exist (we’ll talk about this more next week):

• So: change our desiderata from previous part of lecture

New model:

• $k$ agents, each with private utility for each of $m$ items
• Items arrive one at a time
• Agents bid “like” or “dislike” on items when they arrive
• Mechanism must assign items when they arrive
THE LIKE MECHANISMS

LIKE Mechanism:

- Item arrives
- Some subset of agents bid “Like”
- Mechanism allocates uniformly at random amongst “Likers”

Bad properties ?????????

BALANCED-LIKE Mechanism:

- Same as LIKE, but allocates randomly amongst “Likers” that have received the fewest overall number of items
- Guarantees agent receives at least 1 item per every $k$ she Likes
LIKE Mechanism
• Yes, always Like if utility is nonzero

LIKE is strategy proof for general utility functions
**STRATEGY PROOFNESS**

**BALANCED-LIKE Mechanism ?????????**

**THEOREM**

**BALANCED-LIKE is not SP, even for 0/1 utilities**

---

**True private utilities**

<table>
<thead>
<tr>
<th>Items</th>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent 1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Agent 2</td>
<td>1</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>Agent 3</td>
<td>-</td>
<td>1</td>
<td>-</td>
</tr>
</tbody>
</table>

**Arrivals: a → b → c**

**EV of truthful A1 vs. truthful A2 and A3 ???????**

- 0.5: a → not b → not c, 0.5*1 = 1/2
- 0.5: not a → ...
  - 0.5: not b → c = 0.5*0.5*1 = 1/4
  - 0.5: b → 0.5 c = 0.5*0.5*(1 + 0.5*1) = 3/8
- EV = 1/2 + 1/4 + 3/8 = 9/8

**Manipulation:**

- Don’t bid on item a → Agent 2 gets a
- Bid on b → 0.5: get b = 1/2
- Bid on c → have b? → 0.5: get c; not b? → c
- EV = 1/2 + 1/2 + 1/4 = 5/4 > 9/8
BALANCED-LIKE is SP with 2 agents and 0/1 utilities

BALANCED-LIKE is not SP with 2 agents and general utilities (even for the case of only 2 items)

(See the paper.)
SO THE SYSTEM CAN BE GAMED ...

What does this do social welfare? Fairness?

• Authors were motivated by working with Food Bank Australia, where unsophisticated dispatchers bid on food

• Strong case to be made to care about both objectives!

In general, bidding strategically is quite bad for social welfare:

• Compare sincere behavior against set of Nash profiles

There are instances with 0/1 utilities and $k$ agents where:
the \{egalitarian, utilitarian\} welfare with sincere play under
\{LIKE, BALANCED-LIKE\} …

… is $k$ times the corresponding welfare under a Nash profile.
WHAT ABOUT ENVY?

**Ex-ante** envy freeness: over all possible outcomes, do I expect to be envious?

**Ex-post** envy freeness: after items are allocated, am I envious?

Is LIKE ex-ante E-F under 0/1 utilities ???????????

- **Yes.** Each item’s allocation is independent of past allocations.
- Assume first $m$-1 allocations are EF. Item $m$ arrives. Each of $j < k$ agents with utility 1 receives item in $1/j$ of possible worlds. Still EF.

Is LIKE ex-post E-F under 0/1 utilities ???????????

- **No.** 2 agents, utility 1 for all $m$ items. Agent 1 gets lucky and receives all $m$ items ($P = \frac{1}{2}^m > 0$); **unbounded** envy!
WHAT ABOUT ENVY?

Using similar arguments, paper shows that BALANCED-LIKE under 0/1 utilities is:

- **Ex-ante** envy free
- Bounded **ex-post** envy free (with at most 1 unit of envy)

Quick summary:

- Effect of strategic behavior can be very bad for efficiency!
- Under sincere play, mechanisms seem pretty fair …
  - ... under unit preferences for items
WHAT TO DO?

Motivated by a food bank problem:

• Participants may be altruistic, social-welfare-minded, and relatively unsophisticated → sincere behavior?

Bundle items so participants value them roughly equally

• Equivalent to 0/1 utilities, can leverage fairness properties

Problems:

• Bidders still have self interest
• Bundling items takes time (and produce spoils quickly)
• Bundling items may not always be possible
NEXT UP:
USING FUNNY MONEY TO BID FOR FOOD (OR OTHER ITEMS)

PRESENTER: NIDHI SHAH
NEXT CLASS:
COMBINATORIAL ASSIGNMENT PROBLEMS