APPLIED MECHANISM DESIGN FOR SOCIAL GOOD

JOHN P DICKERSON

Lecture #23 – 04/20/2021
Lecture #24 – 04/22/2021
Lecture #25 – 04/27/2021

CMSC828M
Tuesdays & Thursdays
2:00pm – 3:15pm

COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

ANNOUNCEMENTS

Please turn in a project checkup by April 29th

- Very simple – can just be a paragraph or two written in your project Slack channel, or a new PDF

Research videos:

- Please do make a 10-15 minute video covering a research paper, and/or your research project – something relevant in the space.
- This can be done with your group, too!

No exams:

- Counting as full %, per syllabus

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]

Thanks to: Nisarg Shah (NS), Nick Mattei (NM)

ALLOCATION 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

[Ghodsi et al. NSDI-11]

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)

DOMINANT RESOURCE FAIRNESS (DRF) MECHANISMS

Dominant resource: resource the agent has the biggest share of out of all resources available:

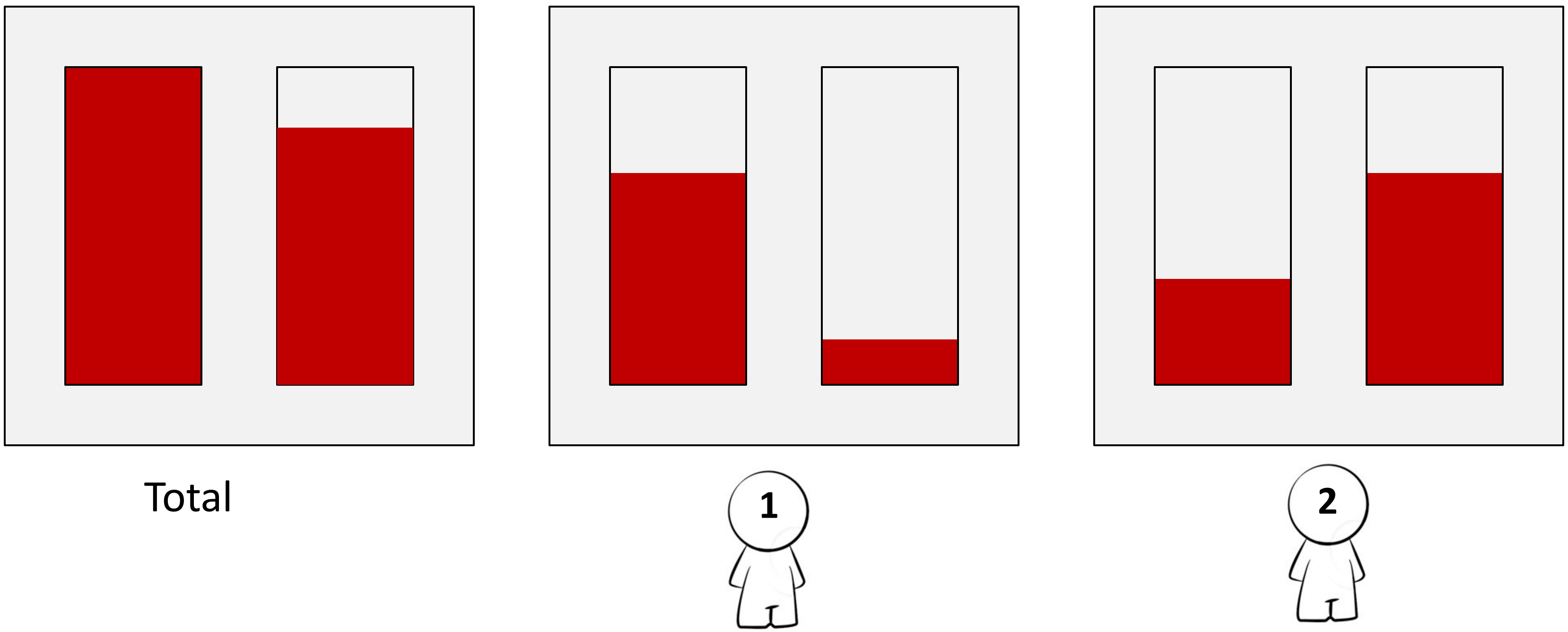
- 16 CPUs, 10 GB available, user allocated 4 CPUs, 8 GB
- Dominant resource is GB, because 4/16 CPU < 8/10 GB

Dominant share: fraction of dominant resource allocated

- Above, dominant share is 8/10 = 80%

STATIC DRF MECHANISM

Dominant Resource Fairness = equalize largest shares
(a.k.a. dominant shares)



Dominant resource (for an agent): resource for which the agent's task requires the largest fraction of total availability

NS

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!**

NS

FORMAL 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!

NS

DESIDERATA

Properties of DRF, aims for a dynamic generalization

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

NS

IMPOSSIBILITY RESULT

Envy freeness + Dynamic Pareto optimality = Impossible

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

Dropping either of them completely → 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 ???????????????

NS

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

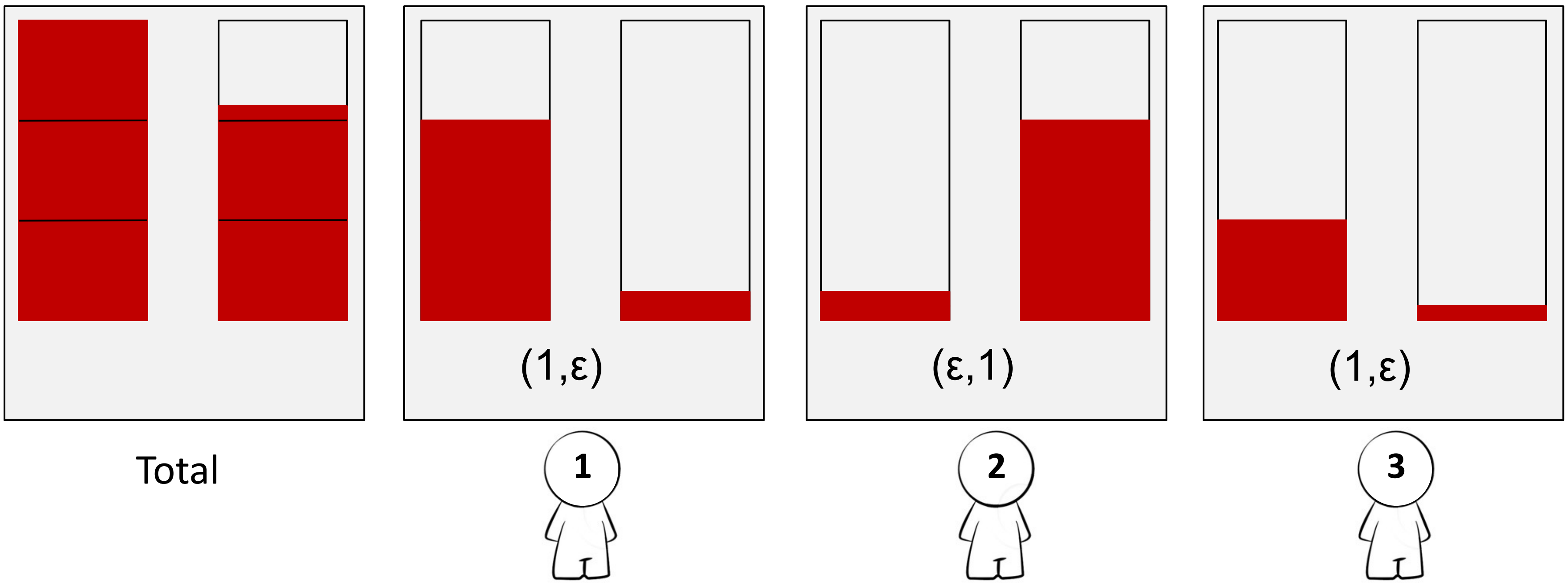
(A constrained "water-filling" algorithm.)

T
H
E
O
R
E
M

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

DYNAMIC-DRF ILLUSTRATED

3 agents, 2 resources



NS

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

THEOREM

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

NS

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)

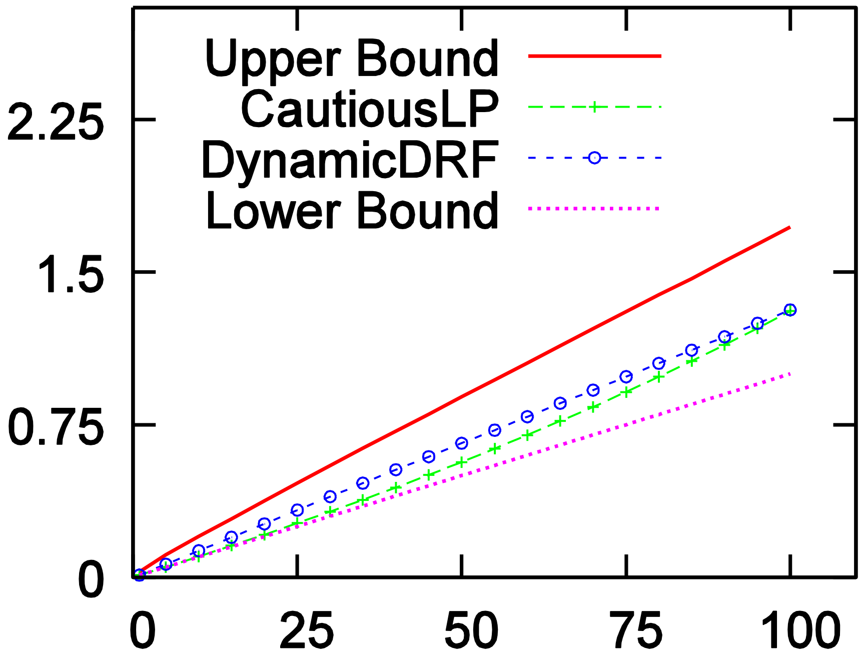
Comparison with provable lower and upper bounds

Data: traces of real workloads on a Google compute cell

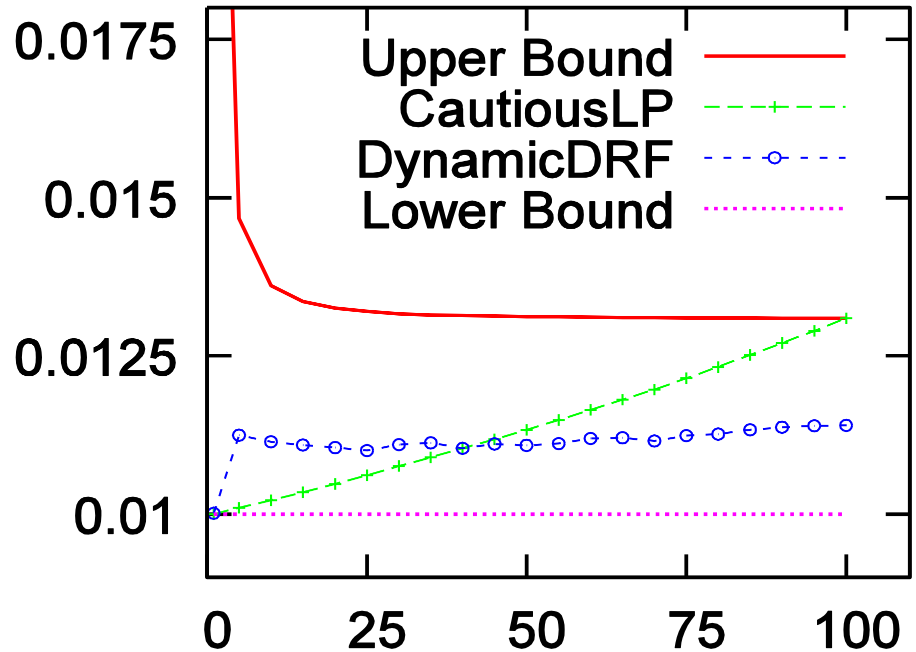
- 7-hour period in 2011, 2 resources (CPU and RAM)
- code.google.com/p/googleclusterdata/wiki/ClusterData2011_1

NS

EXPERIMENTAL RESULTS



Maxsum objective
100 agents



Maxmin objective
100 agents

NS

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 → 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]

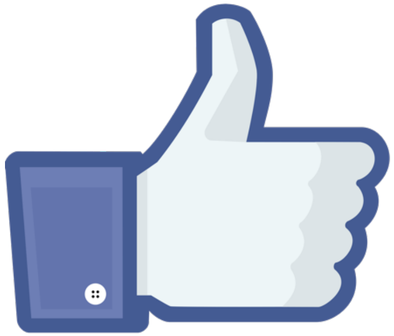
Recall: even in the static setting, an envy-free allocation may not exist (we'll talk about this more next week):

- 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

STRATEGY PROOFNESS

LIKE Mechanism ??????????

- Yes, always Like if utility is nonzero

THEOREM

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

Items	a	b	c
Agent 1	1	1	1
Agent 2	1	-	1
Agent 3	-	1	-

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

STRATEGY PROOFNESS

THEOREM

BALANCED-LIKE is SP with 2 agents and 0/1 utilities

THEOREM

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

THEOREM

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 $j \leq k$ 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 = ½^m > 0); unbounded envy!

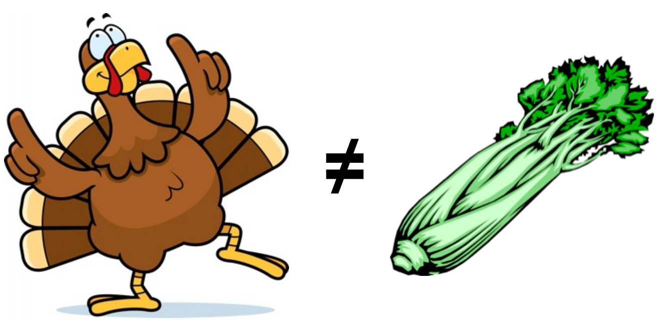
WHAT ABOUT ENVY?

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 then 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

COMBINATORIAL ASSIGNMENT PROBLEMS & COURSE MATCH

Thanks to: John Kubiatowicz (JK)

RECALL: DRF

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

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

Dominant resource: resource the agent has the biggest share of out of all resources available:

- 16 CPUs, 10 GB available, user allocated 4 CPUs, 8 GB
- Dominant resource is GB, because 4/16 CPU < 8/10 GB

Dominant share: fraction of dominant resource allocated

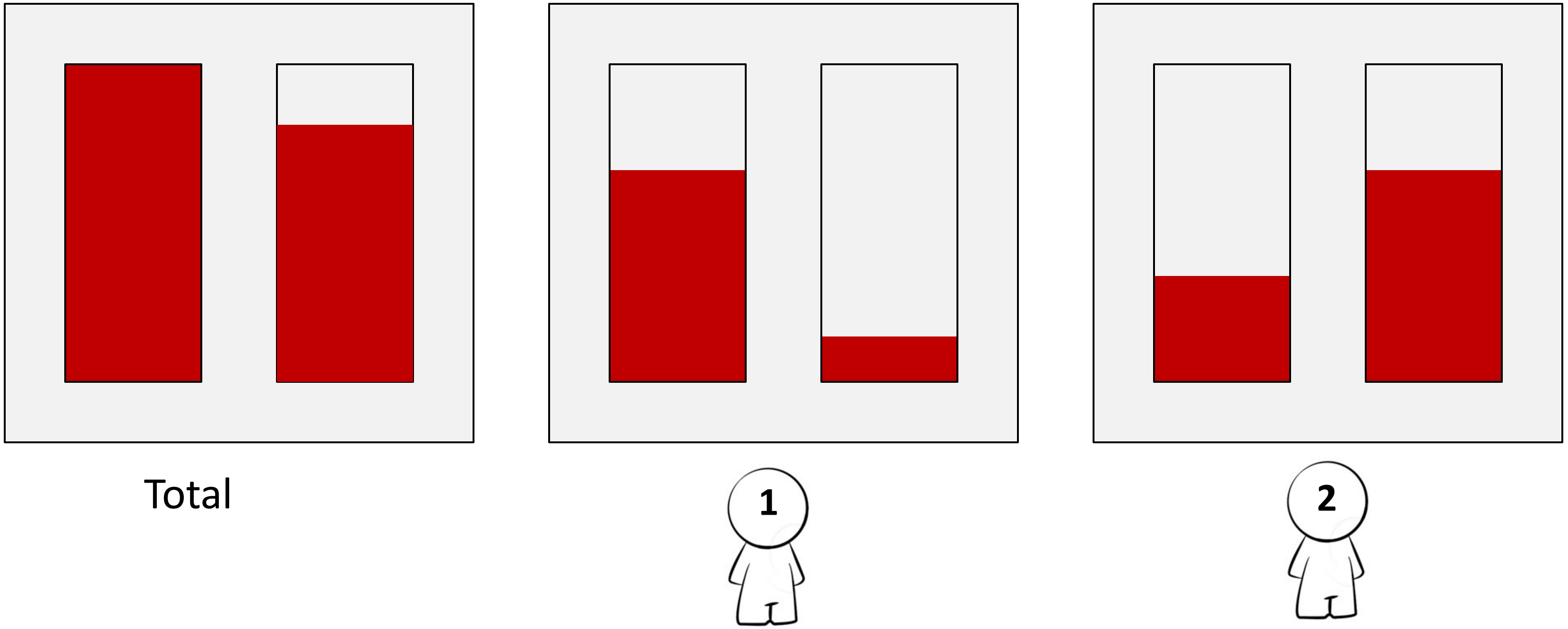
- Above, dominant share is 8/10 = 80%

DRF: application of max-min fairness to dominant shares

- Equalize the dominant share amongst agents

STATIC DRF MECHANISM

Dominant Resource Fairness = equalize largest shares
(a.k.a. dominant shares)



NS

ALTERNATIVE: MAKE A MARKET

Competitive Equilibrium from Equal Incomes (CEEI):

- **Agents report their preferences over sets of items**
- **Give agents an equal budget of funny money**
- **Computer finds prices that clear the market**
    - That is, prices such that when each agent chooses its most favored set that it can afford, the market clears
- **Assign all resources to agents based on their demands and these computed prices**

CEEI EXAMPLE: DIVISIBLE RESOURCES

Supply: {1 cake, 1 doughnut}

Two agents, both with \$1 (funny money), capacity of 1

- ***A*: cake = 1/2, doughnut = 1**
- ***B*: cake = 1/4, doughnut = 1**

Market clearing prices: cake = \$2/5, doughnut = \$8/5

- *A* wants to max $1/2c + 1d$ s.t. $c + d \leq 1$, $p_c c + p_p d <= 1$ ⇒ ?????????? Max: ½ cake, ½ doughnut
- *B* wants to max $1/4c + 1d$ s.t. $c + d \leq 1$, $p_c c + p_p d <= 1$ ⇒ Max: ½ cake, ½ doughnut (and many others – clearinghouse chooses!)

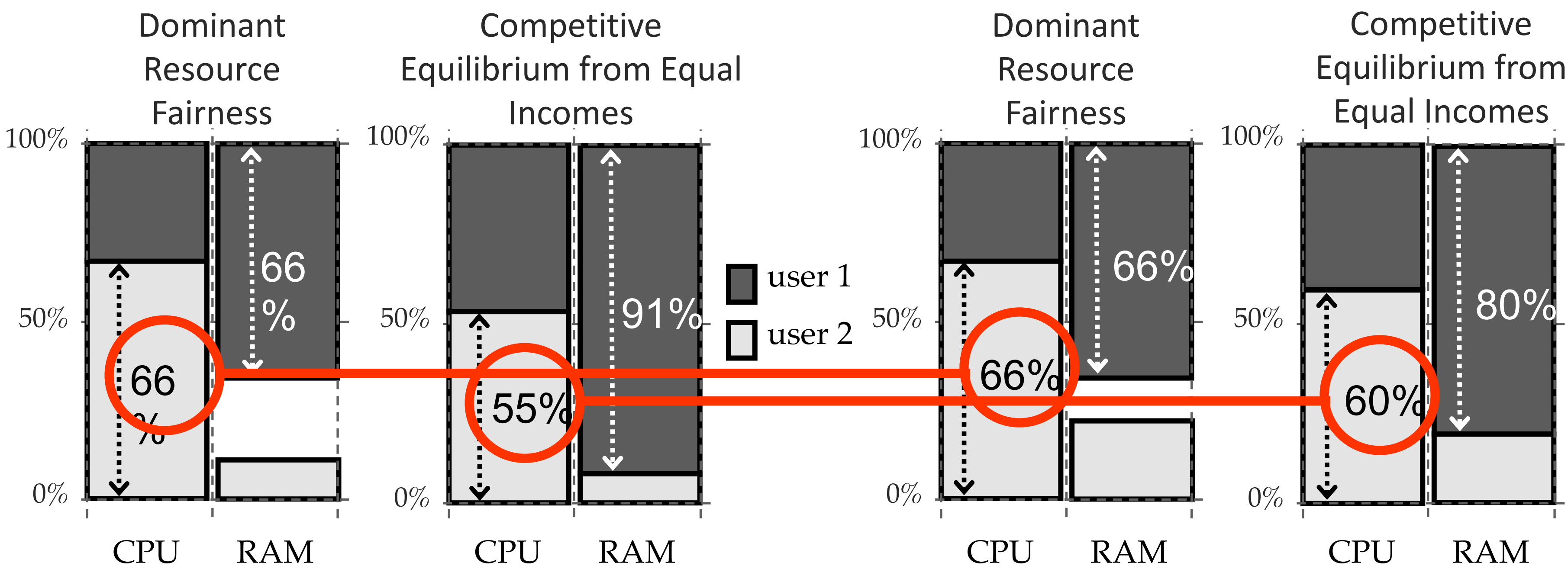
CEEI PROPERTIES

- **Envy-free** ????????
  - Yes! Given the prices, you bought the best bundle you could afford
  - If you envy somebody else's bundle, you could've purchased it!
- **Pareto-efficient** ????????
  - Yes! Market is cleared → taking a Pareto step involves taking a resource from one agent and giving it to somebody new … but this lowers their utility by above
- **Strategy proof** ????????
  - No! Intuition: CEEI clears the market → can game the system by requesting more underutilized resources

DRF VS CEEI

A1: <1 CPU, 4 GB> A2: <3 CPU, 1 GB>

- DRF more fair, CEEI better utilization



A1: <1 CPU, 4 GB> A2: <3 CPU, 2 GB>

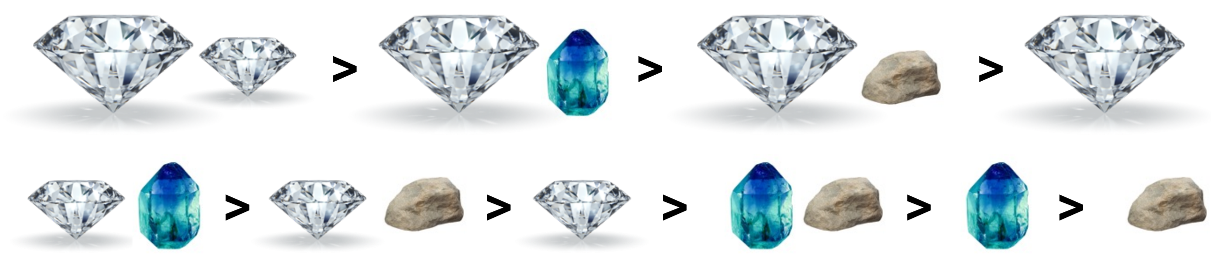
- A2 increased her share of both CPU and memory

CEEI FOR INDIVISIBLE ITEMS?

Two agents

Capacity: 2

Both agents will share the same preference profile:



Market clearing prices ????????

- Don't exist! For any price, for any item, either both agents demand that item or both do not.
- Small changes in price can cause big changes in demand

APPROXIMATE-CEEI

Can we tiebreak somehow?

Idea: give agents slightly different, but roughly equal budgets

- For each agent, draw budget from $[1, 1 + B)$
- $0 < B < \min(1/m, 1/(k-1))$ – k is capacity of agent
- Note: if $B = 0$, this is just CEEI

Still "feels fair" – random winners and losers in the budget draw, and the playing ground is still roughly equal.

A-CEEI FOR INDIVISIBLE ITEMS

Two agents

Capacity: 2

Agent 1's budget: $1.2

Agent 2's budget: $1

 **= $1.10**

 **= $0.80**

 **= $0.20**

 **= $0.10**

????????????

Agent 1:  

Agent 2:  

A-CEEI: PROPERTIES

Always exists if *B* > 0 (need unequal budgets)

The market approximately clears:

- There exist prices that clear the market to within an error of at most √k*m/2
  ??????????????
- Error does not depend on the number of participants → error goes to zero as a fraction of the underlying endowment

Approximately strategy proof

- **“Strategy-proof in the large"**

Bounded envy free

Very difficult to compute!

WHEN DO FAIR ALLOCATIONS EXIST AND HOW DO WE FIND THEM?

Thanks to: Yonatan Aumann (YA), Ariel Procaccia (AP), Shengyu Zhang (SZ)

CUTTING A DIVISIBLE CAKE: MODEL

Division of a heterogeneous divisible good

The cake is the interval [0,1]

Set of agents $N = \{1,\ldots,n\}$

Each agent has a valuation function V_i over pieces of cake

- Additive: if $X \cap Y = \emptyset$ then $V_i(X) + V_i(Y) = V_i(X \cup Y)$
- $\forall i \in N,\ V_i([0,1]) = 1$

Find an allocation $A = A_1,\ldots,A_n$

The cake is a metaphor.

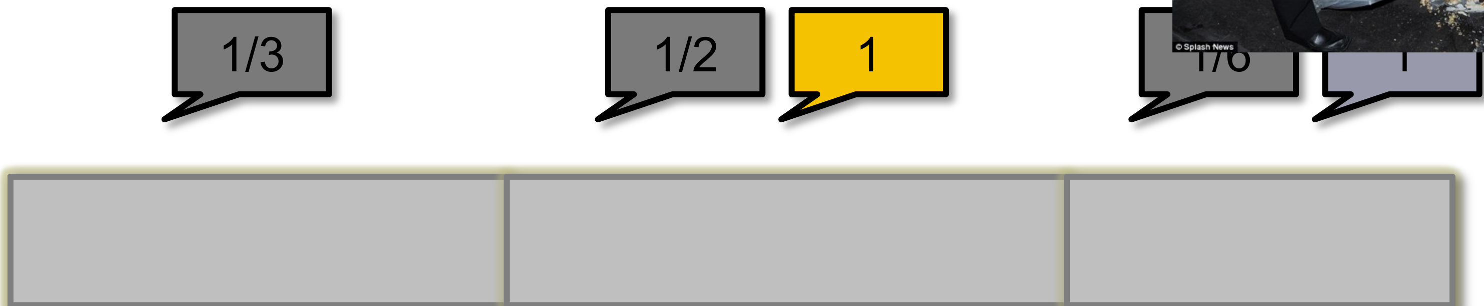
FAIRNESS DEFINITIONS

Proportionality: $\forall i \in N$, $V_i(A_i) \geq 1/n$

Envy-freeness: $\forall i,j \in N$, $V_i(A_i) \geq V_i(A_j)$

Assuming free disposal the two properties are incomparable

- Envy-free but not proportional: ????????????
  - Throw away cake!
- Proportional but not envy-free:



DETERMINISTIC ALGORITHMS

Current research in cake cutting: design truthful, envy free, proportional, and tractable cake cutting algorithms

Requires restricting the valuation functions

- Lower bounds for envy-free cake cutting (see, e.g., [Procaccia, 2009, 2014])

Valuation V_i is piecewise uniform if agent i is uniformly interested in a piece of cake

- E.g., interested uniformly in [0,0.5] but not (0.5,1.0]

THEOREM

Assuming that the agents have piecewise uniform valuations, then there is a deterministic algorithm that is truthful, proportional, envy-free, and polynomial-time.

RANDOMIZED ALGORITHMS

A randomized algorithm is universally envy-free (resp., universally proportional) if it always returns an envy-free (resp., proportional) allocation

A randomized algorithm is truthful in expectation if an agent cannot gain in expectation by lying

→ Looking for universal fairness and truthfulness in expectation

A RANDOMIZED CAKE CUTTING PROTOCOL

A partition $X_1,\ldots,X_n$ is perfect if for every i, k, $V_i(X_k)=1/n$

Algorithm:

1. Find a perfect partition $X_1,\ldots,X_n$
2. Give each player a random piece

Observation [Mossel&Tamuz 2010]: algorithm is truthful in expectation, universally E-F and universally proportional

- Proof: if agent i lies it may lead to a partition $Y_1,\ldots,Y_n$, but ??????????? $\Sigma_k\ (1/n)V_i(Y_k) = (1/n)\ \Sigma_k\ V_i(Y_k) = 1/n$

It is known that a perfect partition always exists [Alon 1987]

- Lemma: if agents have piecewise linear valuations then a perfect partition can be found in polynomial time

COUNTING CUTS & QUERIES

Algorithms for different variants of the problem:

- Finite Algorithms
- "Moving knife" algorithms

Lower bounds on the number of steps required for divisions

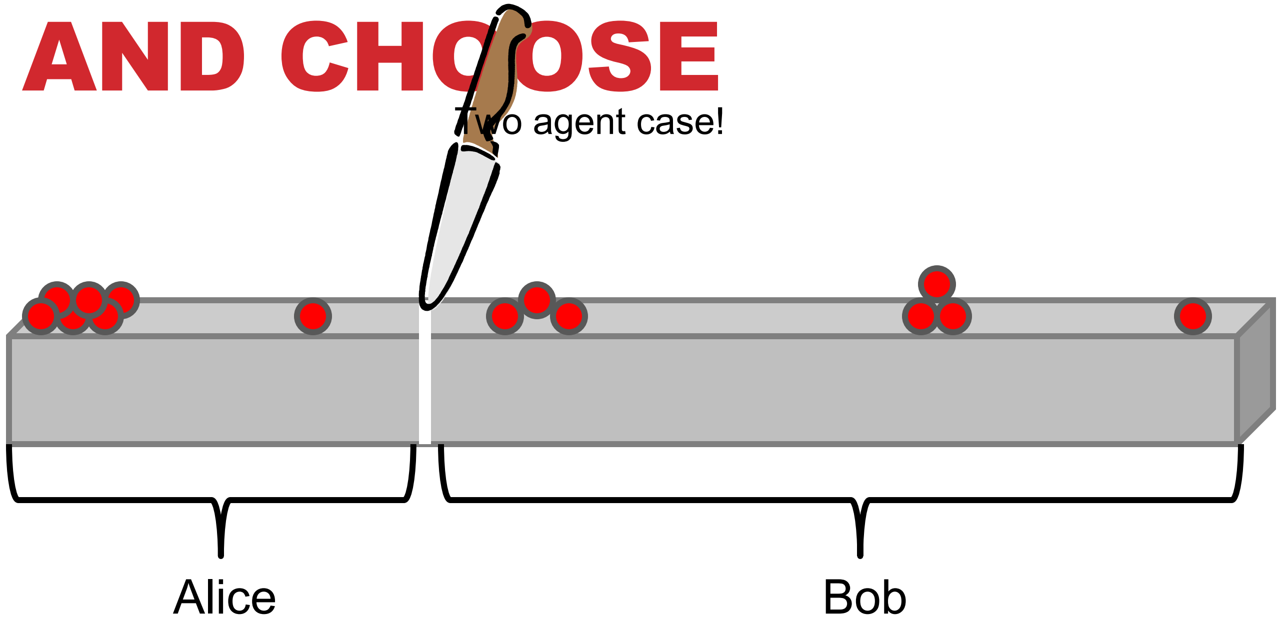
- (see [Procaccia CACM-14] for an easy-to-read discussion)

Until very recently it was unknown if there was a bounded (in terms of queries to agents' valuation functions, and in terms of cuts) and E-F cake cutting algorithm for 4 or more players

- [Aziz and Mackenzie STOC-16]: bounded (231 cuts) for 4 players
- [Aziz and Mackenzie FOCS-16]: bounded (O(n^n^n^n^n^n) queries) for n players

CUT AND CHOOSE

Two agent case!



Alice likes the candies

Bob likes the base

Proportional ??????
Envy free ??????

- ✓ Proportional
- ✓ Envy free
- × Equitable

1. Alice cuts in “her” middle
2. Bob chooses

CUT AND CHOOSE

Three agent case!

Stage 0: Player 1 divides into three equal pieces

- (According to her valuation)

Player 2 trims the largest piece s.t. the remaining is the same as the second largest.

The trimmed part is called Cake 2; the other forms Cake 1

STAGE 1: DIVISION OF CAKE 1

Player 3 chooses the largest piece ("his" largest)

If Player 3 didn't choose the trimmed piece:

- **Player 2 chooses it**

Otherwise:

- **Player 2 chooses one of the two remaining pieces**

Either Player 2 or Player 3 receives the trimmed piece; call that player T

- Call the other player by T'

Player 1 chooses the remaining (untrimmed) piece

STAGE 2: DIVISION OF CAKE 2

T' divides Cake 2 into three equal pieces

- (According to her valuation)

Players T, 1, and T' choose the pieces of Cake 2, in that order.

WHOLE PROCESS

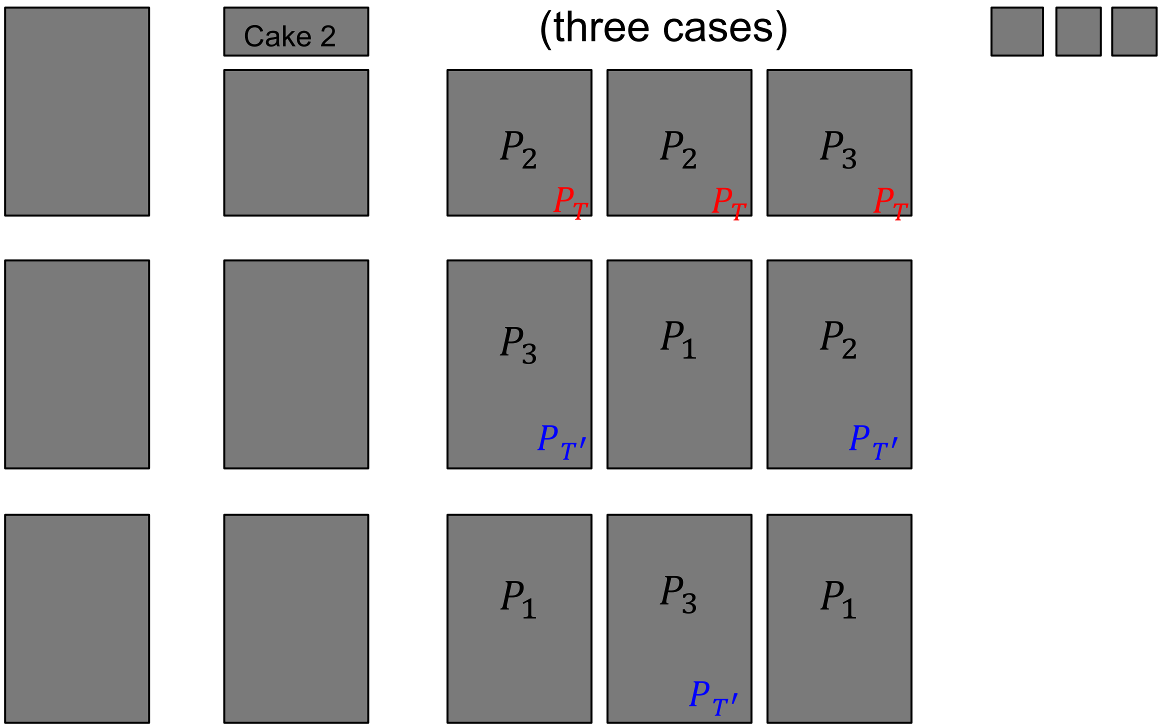
P_1 cuts

P_2 trims

$P_3 \to P_2 \to P_1$ choose cake 1 (three cases)

$P_{T'}$ cuts cake 2

$P_T \to P_1 \to P_{T'}$ choose cake 2



ENVY-FREENESS

The division of Cake 1 is envy-free:

- Player 3 chooses first so he doesn’t envy others.
- Player 2 likes the trimmed piece and another piece equally, both better than the third piece. Player 2 is guaranteed to receive one of these two pieces, thus doesn’t envy others.
- Player 1 is indifferent judging the two untrimmed pieces and indeed receives an untrimmed piece.

ENVY-FREENESS OF CAKE 2

The division of Cake 2 is envy-free:

- Player T goes first and hence does not envy the others.
- Player T' is indifferent weighing the three pieces of Cake 2, so he envies no one.
- Player 1 does not envy T': Player 1 chooses before T'
- Player 1 doesn't envy T: Even if T the whole Cake 2, it's just 1/3 according to Player 1's valuation.

GENERAL n?

An algorithm using recursion

1. Let $P_1, \ldots, P_{n-1}$ divide the cake

- How? Recursively.

2. Now P_n comes.

- Each of $P_1, \ldots, P_{n-1}$ divides her share into n equal pieces
- P_n takes a largest piece from each of $P_1, \ldots, P_{n-1}$

FAIRNESS AND COMPLEXITY

THEOREM

The protocol is (proportional) fair

Proof.

- For $P_1, \dots, P_{n-1}$: each gets $\geq \frac{1}{n-1} \cdot \frac{n-1}{n} = \frac{1}{n}$.
- P_n: gets $\geq \frac{a_1}{n} + \cdots + \frac{a_{n-1}}{n} = \frac{1}{n}$
  - a_i: P_n's value of P_i's share in Step 1.

Complexity? Let $T(n)$ be the number of pieces.

- Recursion: $T(n) = n \cdot T(n-1)$
- $T(1) = 1$, and $T(n) = n!$ for general n.

MOVING KNIFE PROTOCOLS

[Dubins-Spanier 1961]

Continuously move a knife from left to right.

1. A player yells out "STOP" as soon as knife has passed over $1/n$ of the cake

- (By her valuation function)

2. The player that yelled out is assigned that piece. (And she is out of the game; $n \leftarrow n - 1$)

- Break ties arbitrarily

3. The procedure continues until everyone gets one piece

FAIRNESS AND COMPLEXITY

THEOREM

The protocol is (proportional) fair

Proof.

- For the first who yells out: she gets $1/n$
- For the rest: each things that the remaining part has value at least $\frac{n-1}{n}$, and $n-1$ people divide it
  - Recursively: each gets $\frac{1}{n-1}\frac{n-1}{n} = \frac{1}{n}$.

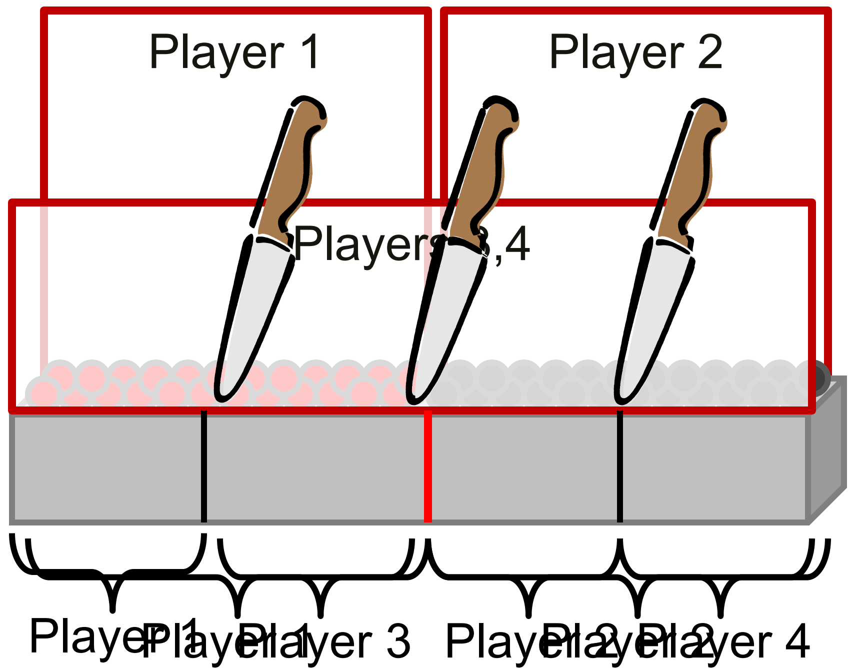
Complexity ???????

- Only $n-1$ cuts into n pieces
- Query complexity ???????

Envy free ???????

WHAT ABOUT FAIRNESS VS SOCIAL WELFARE?

E-F allocation ????????



Social welfare maximizing allocation ???????

Total: 1.5

Total: 2

Fairness ≠ Maximum Utility

THE PRICE OF FAIRNESS IN CAKE CUTTING

Given an instance:

$$\text{PoF} = \frac{\text{max } \textbf{welfare} \text{ using any division}}{\text{max } \textbf{welfare} \text{ using fair division}}$$

utilitarian

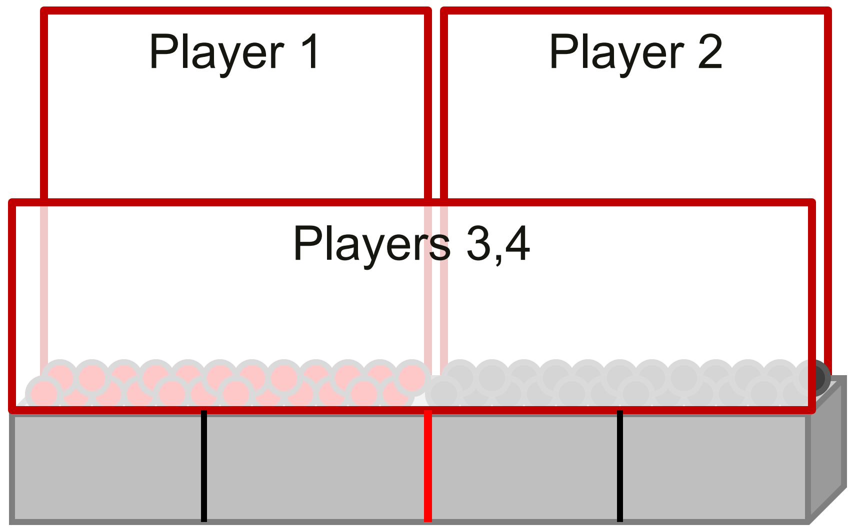
egalitarian (= minimum of players' utilities)

Price of envy-freeness

Price of equitability

Price of proportionality

PRICE OF E-F: CONTINUED EXAMPLE



Envy-free

Total: 1.5

Utilitarian optimum

Total: 2

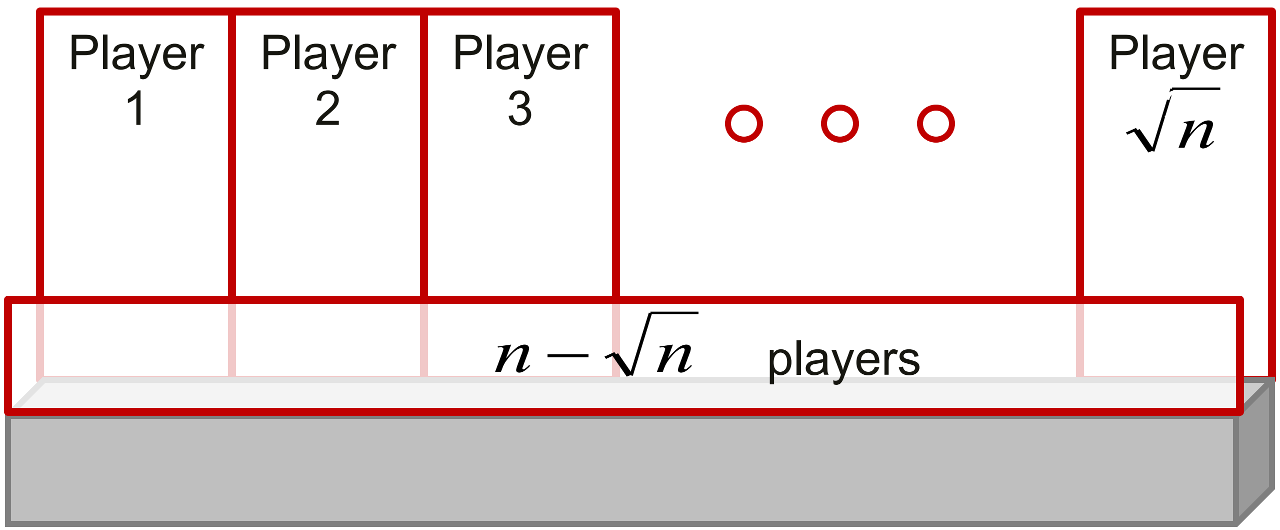
Utilitarian Price of Envy-Freeness: 4/3

“PRICE OF” BOUNDS

[Aumann & Dombb]

Price of …	Proportionality	Envy freeness	Equitability
Utilitarian	$\frac{\sqrt{n}}{2}+O(1)$		$n+O(1)$
Egalitarian	1	$\frac{n}{2}$	1

UTILITARIAN PRICE OF E-F: LOWER BOUND



Best possible utilitarian: $\sqrt{n}$

Best proportional/envy-free utilitarian: $\frac{1}{n}\cdot\left(n-\sqrt{n}\right)+1<2$

Utilitarian Price of envy-freeness:
$\approx \sqrt{n}/2$

CEEI FOR MULTIPLE DIVISIBLE ITEMS

[Varian 1974]

Endow all players with a budget of $1

Competitive equilibrium is:

- (Virtual) prices such that …
- … when each player buys their most valuable bundle at those prices ...
- ... the market clears.

Tough to compute

Envy free allocation

- (I can afford any other player's bundle, but chose my own)

RECALL: CEEI FOR INDIVISIBLE ITEMS?

Two agents

Capacity: 2

Both agents will share the same preference profile:

Market clearing prices

- Don't exist! For any price, for any item, either both agents demand that item or both do not.

Got around this via "A-CEEI," slightly different budgets for agents, envy free up to 1 good, ~SP in the large …

ENVY-FREENESS UP TO ONE GOOD

Recall: an allocation $A_1, \ldots, A_n$ is envy free up to one good (EF1) if for all i, j,

$$v_i(A_i) \geq v_i(A_j) - \max_{g \in A_j} v_i(g)$$

A round-robin allocation is EF1:

MAXIMUM NASH WELFARE

However, round robin is not Pareto efficient

Can we find a mechanism that is both EF1 and Pareto efficient?

Idea: Maximize the Nash welfare $\prod_i v_i(A_i)$

For homogeneous divisible goods:

- Envy free and Pareto efficient
- Coincides with CEEI and proportional fairness

For indivisible goods:

- Rounding does not work

THEOREM

Maximizing Nash welfare satisfies EF1 and Pareto efficiency

[Caragiannis et al. EC-2016]

WHEN DO TRULY E-F ALLOCATIONS EXIST?

Can we characterize when an EF allocation of indivisible goods exists (with high probability)?

[A1]: utilities are drawn I.I.D.

[A2]:

- each agent equally likely to want *g* the most
- difference between the expected utility of the agent most wanting *g* and any other agent is at least some constant μ

Uniform distribution satisfies [A1] and [A2]

Goods with intrinsic base values → only [A2]

A SMALL NUMBER OF GOODS

THEOREM

Even when the number of goods is larger than the number of agents by a **linear fraction**, an EF allocation probably won't exist.

[Dickerson et al. AAAI-14]

Note: if $m < n$, clearly no EF allocation exists.

- How many additional goods beyond $m=n$ are needed?

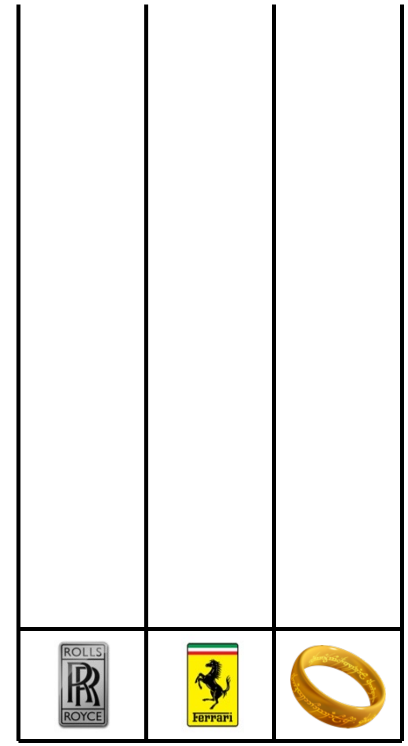
Formally: under [A1], for small constant δ:

- if the probability that EF allocation exists is $1-\delta$
- then $m \geq (1+c(\delta))n$, with $c(\delta)>0$

A SMALL NUMBER OF GOODS

Thought: If two agents want the same good the most, require at least three goods for an envy-free allocation

Count such collisions; are there too many?



A LARGE NUMBER OF GOODS

THEOREM

When the number of goods is larger than the number of agents by a **logarithmic factor**, an EF allocation probably exists.

Formally: under [A2], with $n = O(m/\ln m)$:

- An EF allocation exists (w.p.1) as $m \to \infty$

Idea: give each good to the agent who wants it the most

- This produces EF allocations with high probability

A LARGE NUMBER OF GOODS

Proof of the theorem uses a natural mechanism that also maximizes social welfare over the space of allocations

Alternate theorem statement:

THEOREM

"When the number of goods is larger than the number of agents by a logarithmic factor, **the social welfare-maximizing allocation is EF**."

EXPERIMENTAL VALIDATION

Both theorem statements hide constants

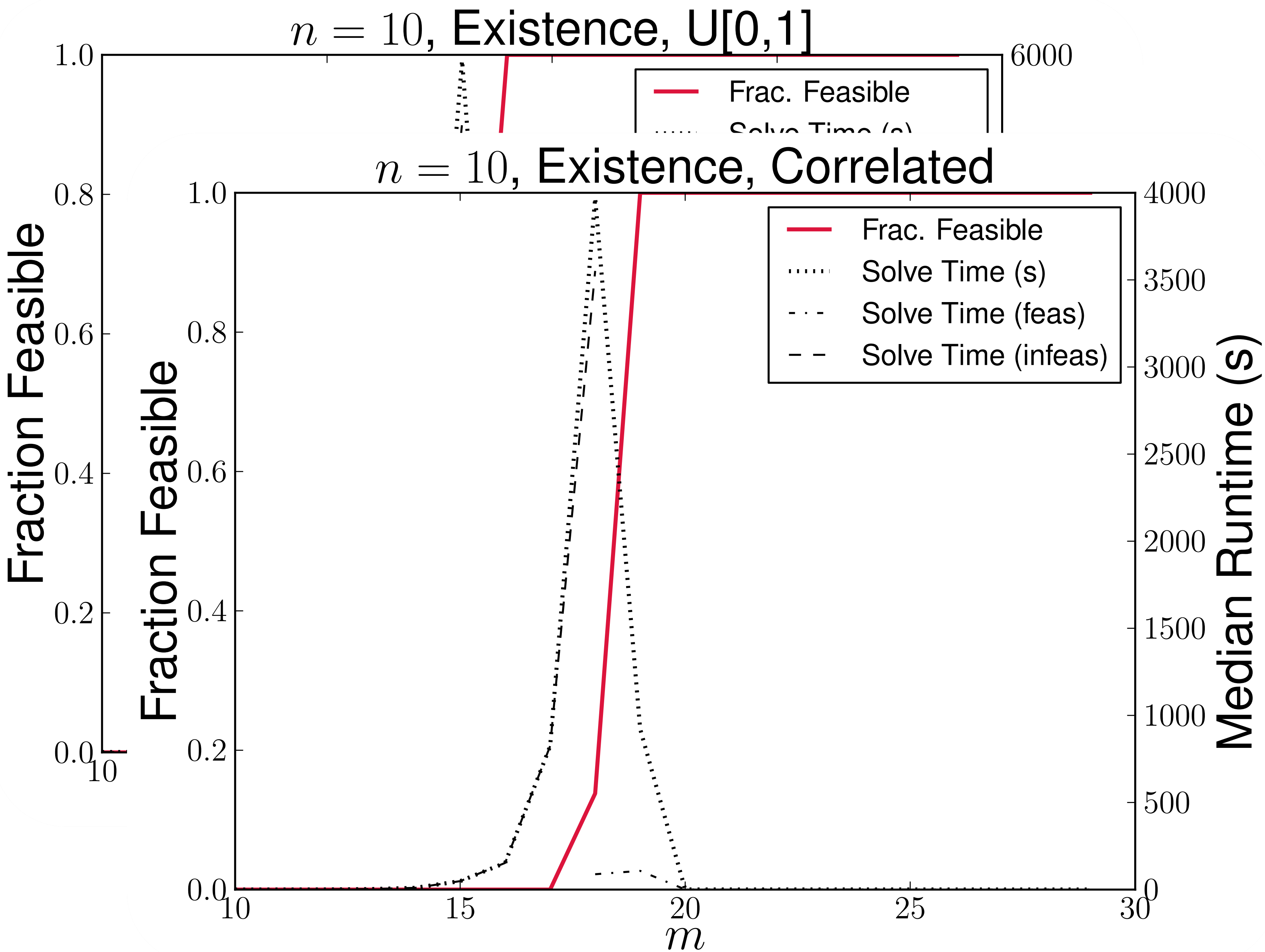
- When do these results "kick in"?

We test under two distributions:

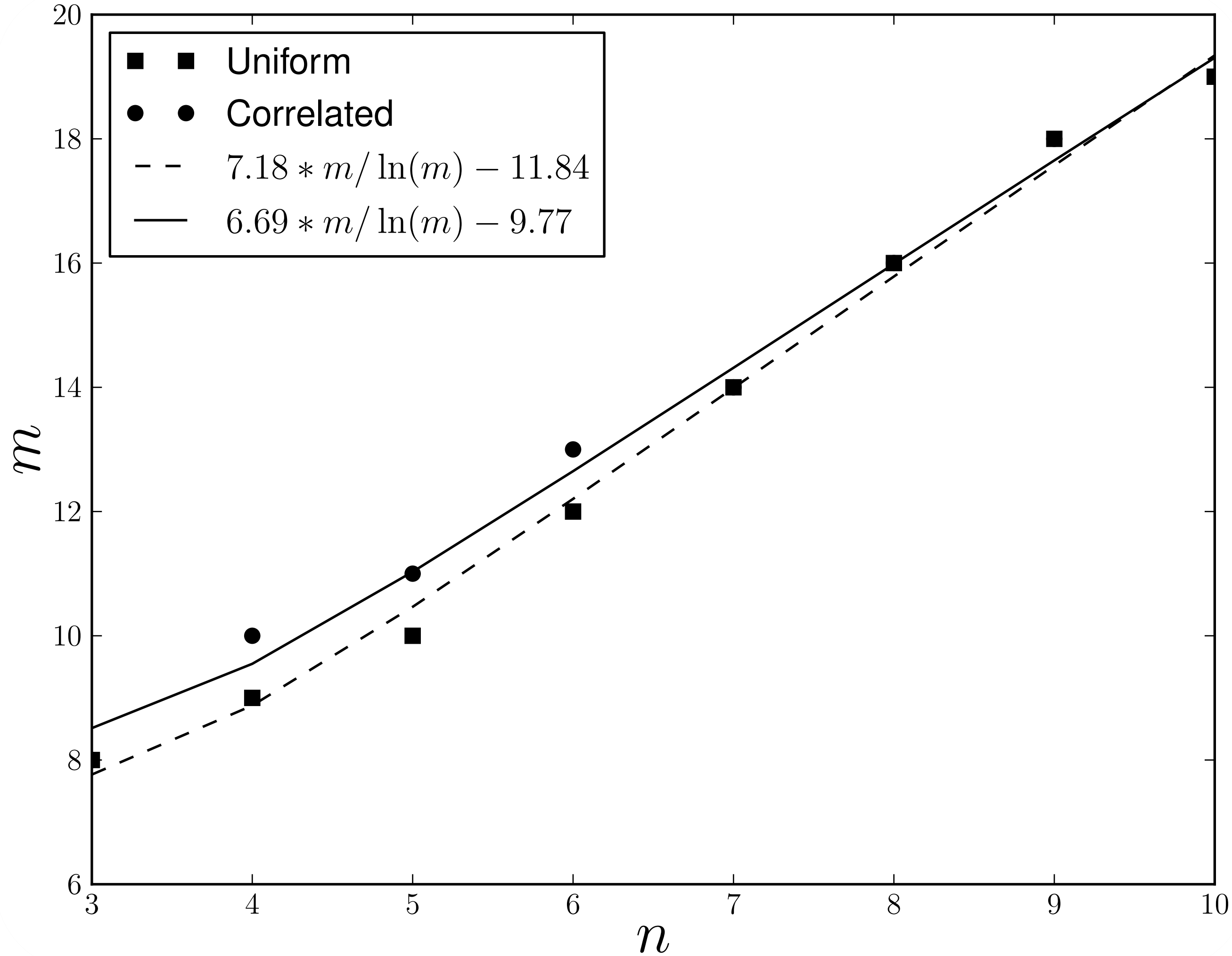
- Uniform
  - Satisfies [A1] and [A2] and thus both theorems
- Correlated (goods have intrinsic values)
  - Satisfies [A2] and thus Theorem 2

Hold *n* constant, vary *m*, see when EF allocations exist

- And how long it takes to find them (or prove otherwise)



$n = 10$, Existence, U[0,1]
Frac. Feasible
Solve Time (s)
Fraction Feasible
$n = 10$, Existence, Correlated
Frac. Feasible
Solve Time (s)
Solve Time (feas)
Solve Time (infeas)
Fraction Feasible
Median Runtime (s)
m



- Number of agents (x-axis) vs. number of items (y-axis) before at least 99% of the instances had an EF allocation, for each of the Uniform and Correlated distributions.
- Theorem 2: w.h.p. occurs when $n = O(m/\ln m)$ – **aligns with results**.

EXPLORING THE PHASE TRANSITION

Is the runtime spike an artifact of the model?

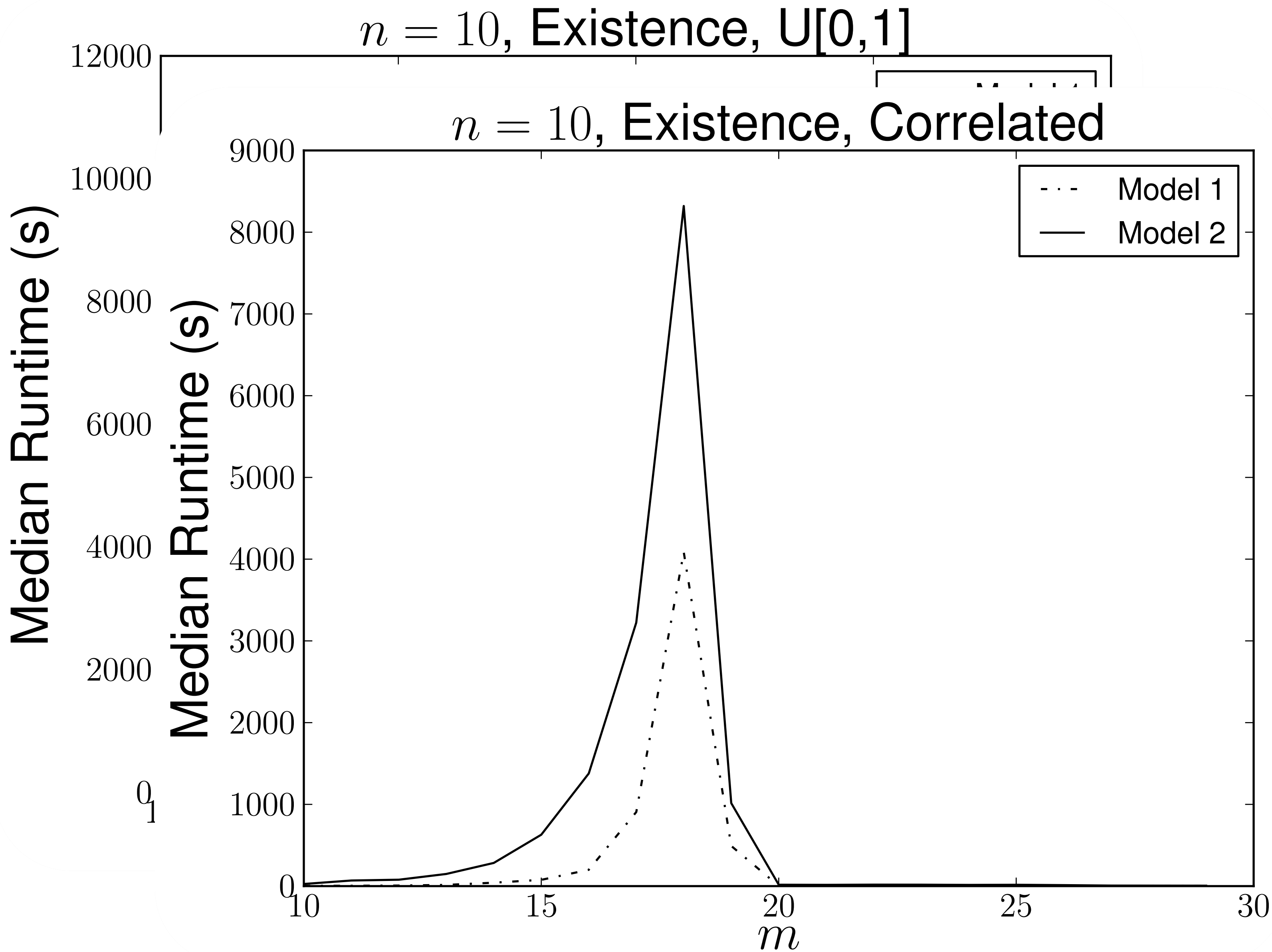
Tried two models in the paper:

- Feasibility problem (Model #1)
- Optimization problem (Model #2)

Motivation: state-of-the-art IP solvers treat feasibility and optimization problems differently

- Some evidence that adding objective can help (e.g., the “MIP Nash” paper [Sandholm Gilpin Conitzer 2005])

$$
\begin{array}{lll}
\text{find} & x_{ig} \quad \forall i \in N, g \in G & \\
\text{s.t.} & \sum_{i \in N} x_{ig} = 1 & \forall g \in G \\
& \sum_{g \in G} v_{ig} x_{i'g} - \sum_{g \in G} v_{ig} x_{ig} \leq 0 & \forall i \neq i' \in N \\
& x_{ig} \in \{0, 1\} & \forall i \in N, g \in G
\end{array}
$$


$n = 10$, Existence, U[0,1]
$n = 10$, Existence, Correlated
Median Runtime (s)
Median Runtime (s)
Model 1
Model 2
m
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
12000
10
15
20
25
30