APPLIED MECHANISM DESIGN FOR SOCIAL GOOD

JOHN P DICKERSON & MARINA KNITTEL

Lecture #14 – 04/20/2021
Lecture #15 – 04/22/2021
Lecture #16 – 04/27/2021

CMSC498T
Mondays & Wednesdays
2:00pm – 3:15pm

COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

ANNOUNCEMENTS

Projects!

- Due this Friday: a brief outline of what you would like to work on
- We strongly encourage working in a group

Paper Presentations:

- Due *tonight*: a brief written or recorded summary of a paper, book chapter, topic of interest, etc.
- Check out our website for some examples

CHECK-IN SURVEY

Two weeks ago, you guys filled out a survey about how we're doing.

Takeaways

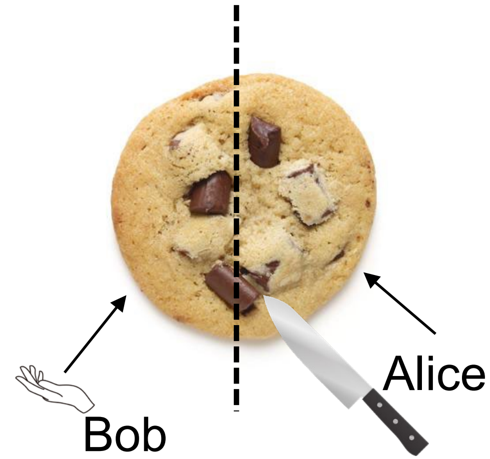
- Marking questions you got wrong
- Post readings and quiz solutions earlier
- You guys like applications/programming
- Where is this class going????

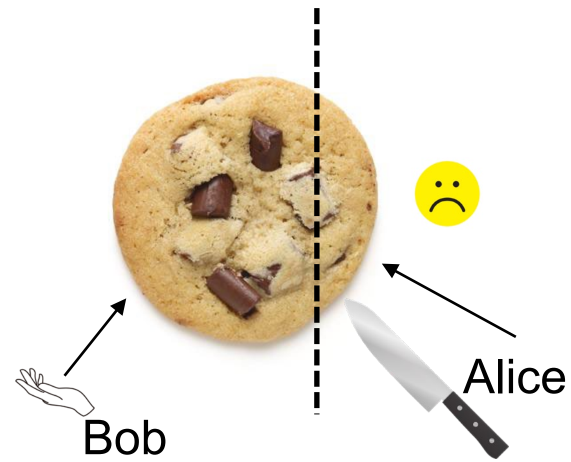
FAIRLY DIVIDING A COOKIE

How should Alice and Bob divide a cookie?

How do know this will give each person half the cookie (assuming Alice is sufficiently dexterous)?

How do we know giving Alice and Bob half the cookie each is fair?





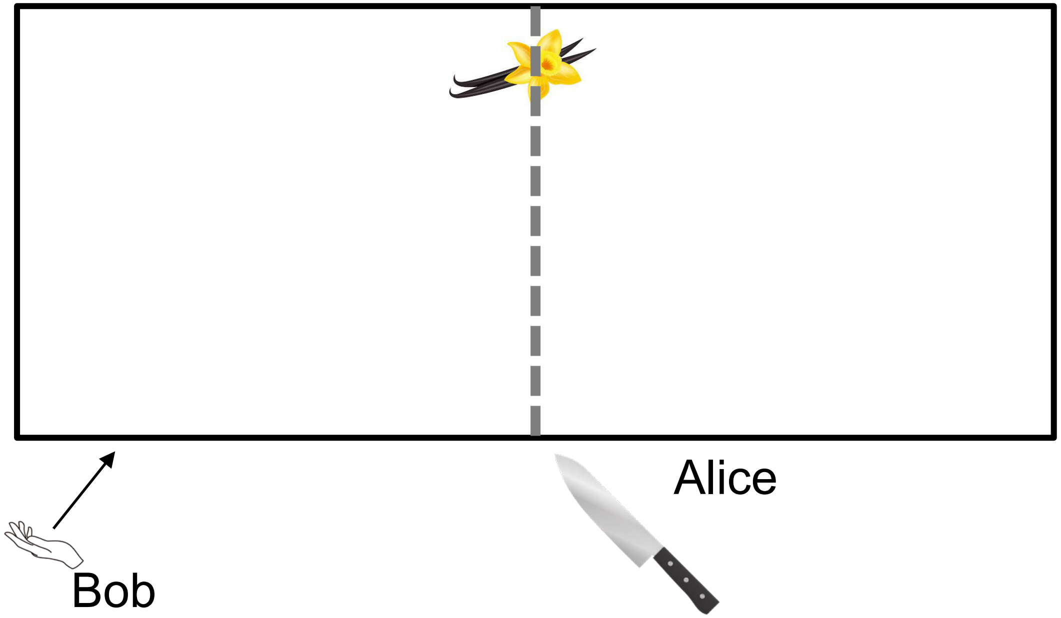
FAIRLY DIVIDING A SHEET CAKE

The cake is a metaphor.

How should Alice and Bob divide a vanilla sheet cake?

$\mathbf{u_{Alice}(vla) = 10}$

$\mathbf{u_{Bob}(vla) = 20}$

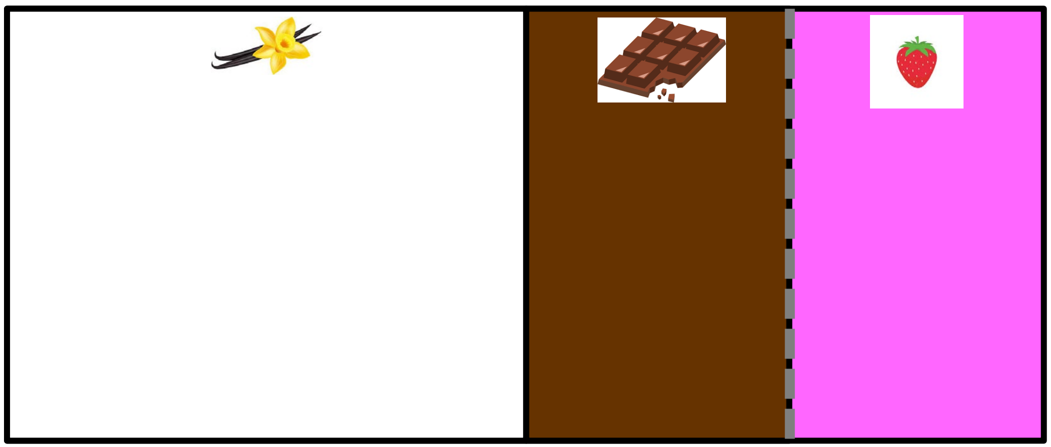


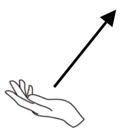
FAIRLY DIVIDING A MULTI-FLAVOR SHEET CAKE

How should Alice and Bob divide a multi flavored sheet cake?

$u_{Alice}(vla) = 10$ **$u_{Alice}(chc) = 10$** **$u_{Alice}(str) = 30$**

$u_{Bob}(vla) = 20$ **$u_{Bob}(chc) = 20$** **$u_{Bob}(str) = 20$**





Bob

$u_{Alice}(left) = 10 * ½ + 10 * ¼ = 7.5$
$u_{Alice}(right) = 30 * ¼ = 7.5$
$u_{Bob}(left) = 20 * ½ + 20 * ¼ = 15$
$u_{Bob}(right) = 20 * ¼ = 5$

Alice

Bob gets more utility than Alice.

Bob strictly favors his piece.

Alice weakly favors her piece.

Is this fair???

Alice could lie knowing Bob's utilities.

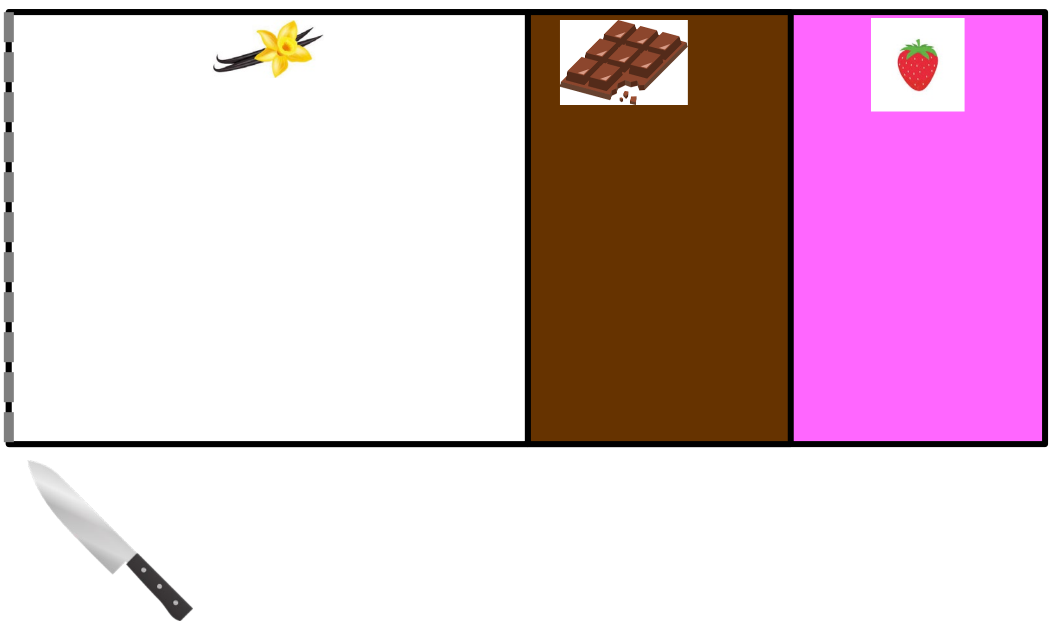
TREATING PLAYERS EQUALLY

This is another algorithm for dividing divisible goods (say, a sheet cake) between two players.

A referee moves the knife along the cake. Once either Alice or Bob views the slices as fair, they say “stop” (say in this example, Alice says stop).

A second knife is placed. Move the knives in parallel, keeping the cut in between them proportional according to Alice. Once Bob views the middle slice as proportional, he says “stop”.

Randomly assign one of them the outer slices and one of them the inner slice.



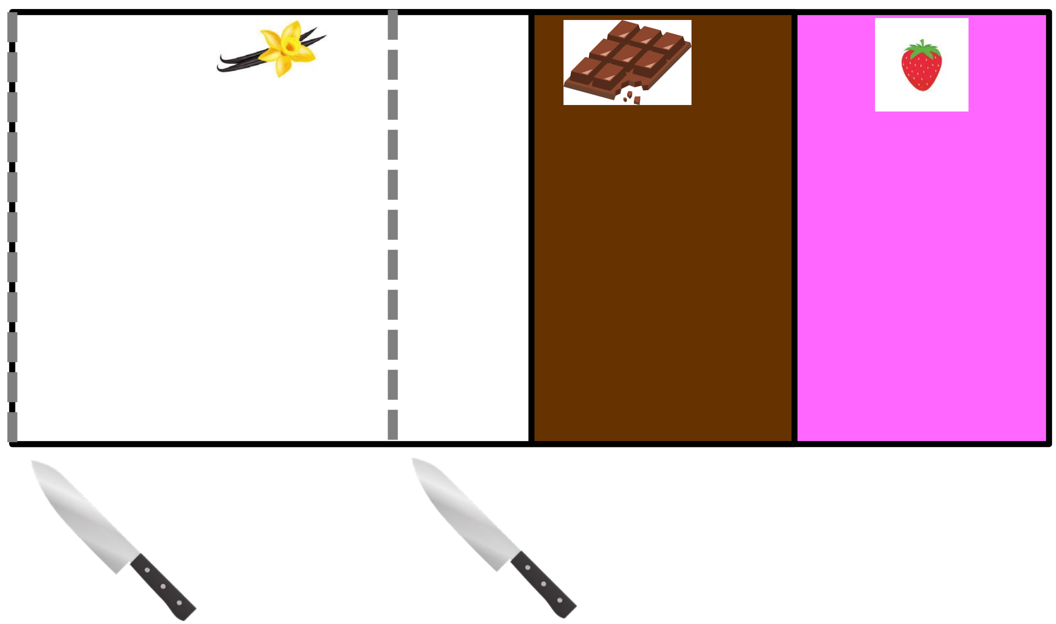
TREATING PLAYERS EQUALLY

This is another algorithm for dividing divisible goods (say, a sheet cake) between two players.

A referee moves the knife along the cake. Once either Alice or Bob views the slices as fair, they say "stop" (say in this example, Alice says stop).

A second knife is placed. Move the knives in parallel, keeping the cut in between them proportional according to Alice. Once Bob views the middle slice as proportional, he says "stop".

Randomly assign one of them the outer slices and one of them the inner slice.



Both Bob and Alice get what they perceive as exactly ½ the utility of the cake.

HOW DO WE QUANTIFY FAIRNESS?

There are n players.

Proportionality (Prop): All players get what they perceive as 1/n of the total available value.

Envy-freeness (EF): No player thinks that another player received a better share than they did.

Pareto-optimality (PO): There is no other allocation that *pareto dominates* it

- **Pareto-dominance:** An allocation pareto dominates another if it is strictly better for at least one player and at least as good for all others

FAIRLY DIVIDING A MULTI-FLAVOR SHEET CAKE

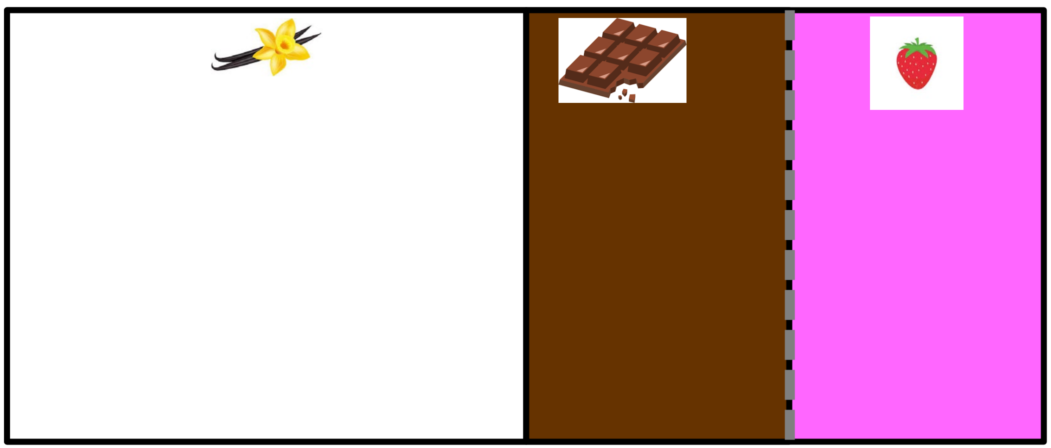
How should Alice and Bob divide a vanilla sheet cake?

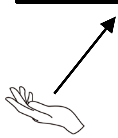
$u_{Alice}(vla) = 10$	**$u_{Alice}(chc) = 10$**	**$u_{Alice}(str) = 30$**
$u_{Bob}(vla) = 20$	**$u_{Bob}(chc) = 20$**	**$u_{Bob}(str) = 20$**

Prop? **Yes.**

EF? **Yes.**

PO? **Yes, but not generally.**





Bob

$u_{Alice}(left) = 10 * ½ + 10 * ¼ = 7.5$
$u_{Alice}(right) = 30 * ¼ = 7.5$
$u_{Bob}(left) = 20 * ½ + 20 * ¼ = 15$
$u_{Bob}(right) = 20 * ¼ = 5$

Alice

PROP AND EF ON 2 PLAYERS

P1 = Player 1
P2 = Player 2
A1 = P1's allocation
A2 = P2's allocation
M = the entire cake

Is one stronger than the other? Are they equivalent? Are they incomparable?

Prop => EF:

Assume P1 gets what they perceive to be a proportional allocation. $u_{P1}(A1) \geq u_{P1}(M)/2$. Note that $u_{P1}(A1) + u_{P1}(A2) = u_{P1}(M)$. Thus $u_{P1}(A2) \leq u_{P1}(M)/2$. Thus P1 cannot *envy* P2. The same argument can be made for P2.

EF => Prop:

Assume P1 does not envy P2. Then $u_{P1}(A1) \geq u_{P1}(A2)$. Since $u_{P1}(A1) + u_{P1}(A2) = u_{P1}(M)$, then $u_{P1}(A1) \geq u_{P1}(M)/2$. Thus P1 gets a proportional allocation. The same argument can be made for P2.

PROP AND EF ON N>2 PLAYERS

Pi = Player I
Ai = Pi's allocation
M = the entire cake

Does the equivalence hold when n>2?

Prop DOES NOT IMPLY EF:

Consider 3 agents. Say, from P1's perspective, $u_{P1}(A1) = 1/3$, $u_{P1}(A2) = 1/2$, and $u_{P1}(A3) = 1/6$. Clearly P1 views their share as proportional, but they envy P2.

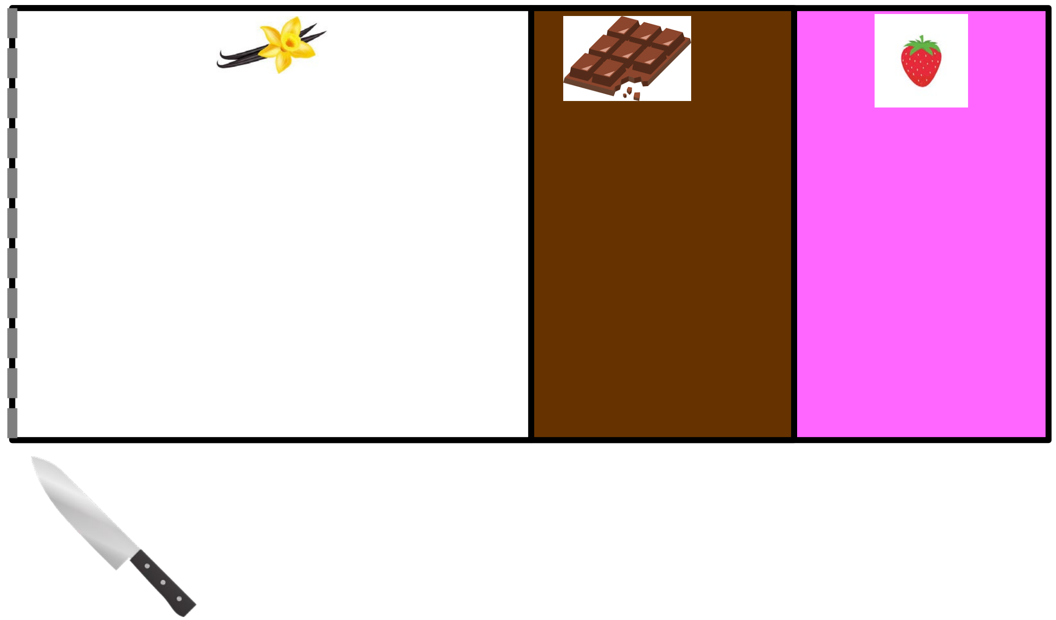
EF => Prop:

Assume P1 does not envy P2. Then $u_{P1}(A1) \geq u_{P1}(Ai)$ for all i. Since $u_{P1}(A1) + \ldots + u_{P1}(An) = u_{P1}(M)$, then $u_{P1}(A1) \geq u_{P1}(M)/n$. Thus P1 gets a proportional allocation. The same argument can be made for Pi for all i.

PROPORTIONALITY FOR N>2 VIA KNIFE MOVING

A referee moves the knife along the cake. Once any player views the slices as a proportional slice, they say "stop". They then take the piece.

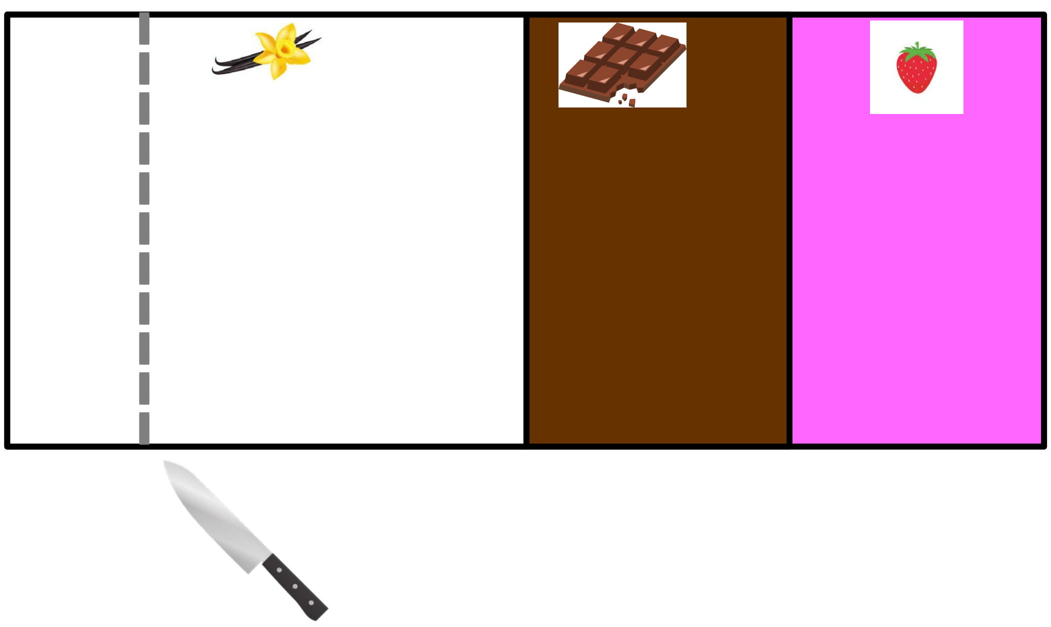
Repeat this procedure until all slices are allocated.



PROPORTIONALITY FOR N>2 VIA KNIFE MOVING

A referee moves the knife along the cake. Once any player views the slices as a proportional slice, they say "stop". They then take the piece.

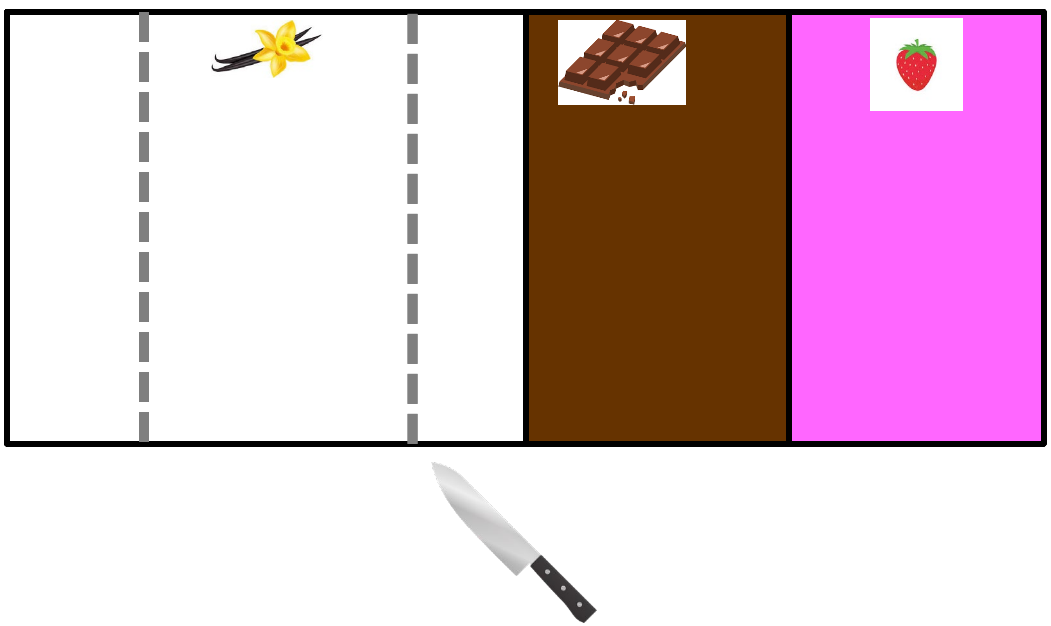
Repeat this procedure until all slices are allocated.



PROPORTIONALITY FOR N>2 VIA KNIFE MOVING

A referee moves the knife along the cake. Once any player views the slices as a proportional slice, they say “stop”. They then take the piece.

Repeat this procedure until all slices are allocated.



PROPORTIONALITY FOR N>2 VIA KNIFE MOVING

Quick proof:

Say the order of calling “stop” is P1, P2, … Pn.

P1 obviously gets a proportional slice.

According to player to P2, P1 got at most $u_{P2}(A1) \leq u_{P2}(M)/n$. Thus the amount left is at least (n-1) * $u_{P2}(M)/n$ according to P2.

When P2 says stop, they get a 1/(n-1) fraction of the remaining value. That is $[(n-1) * u_{P2}(M)/n\ ]/(n-1) \geq u_{P2}(M)/n$. Therefore, P2 gets a proportional allocation.

A similar argument can be made going down the line of players.

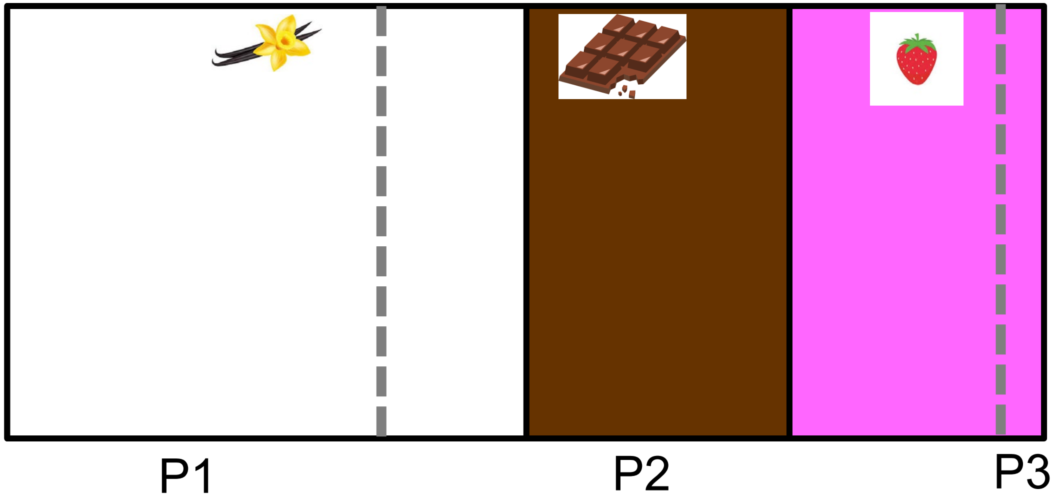
Also: Notice that we made n-1 cuts. This is the minimum possible number of cuts to divide it into n pieces.

But: This is not envy-free. It also prefers later players.

GREEDY ALGS AREN'T ENVY-FREE

Assume we have any greedy algorithm for dividing a cake. That is, an agent P1 gets assigned a piece and then has no influence on the division of the rest of the cake.

If P1 gets less than half the cake, they have no idea how the rest could be allocated. Maybe P2 gets practically the rest of the cake. Thus, they envy P2.



AN ENVY-FREE PROTOCOL FOR N=3

1. P1 cuts into 3 proportional pieces.

2. P2 "trims" the largest piece only if one is the largest. Set aside trimmings, L.

3. P3, then P2, then P1 choose their favorite pieces. If P2 "trimmed" they must pick the trimmed piece if free. If no trimming happened, we stop here.

NOTE: P2 or P3 got the trimmed part. We say Pi (P2 or P3) got it, Pj (the other) did not.

4. Pj cuts L into 3 proportional pieces.

5. Pi, then P1, then Pj choose their favorite pieces.

NOTE: Each player may have 2 pieces.

ENVY-FREENESS BEYOND N=3

It's um… well it's complicated. And we don't need to discuss it.

In some ways it is an extension of the n=3 algorithm, with a fair amount of added steps. See: reading on course website (Brams and Taylor) for the n=4 version.

Notable difference: the number of cuts is unbounded, whereas the n=3 algorithm only ever requires 5 cuts.

But hey! At least it means that envy-freeness always exists ☺ (this was shown non-constructively much earlier)

Or does it….?

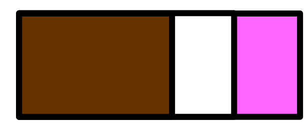
Guess you'll have to wait until after Spring Break…

WHAT DO WE KNOW ABOUT DIVISIBLE GOODS?

- **When n=2:** EF ⇔ Prop
- **When n>2:** EF => Prop
- There exist EF and Proportional algorithms for any n agents.

Other notions to consider:

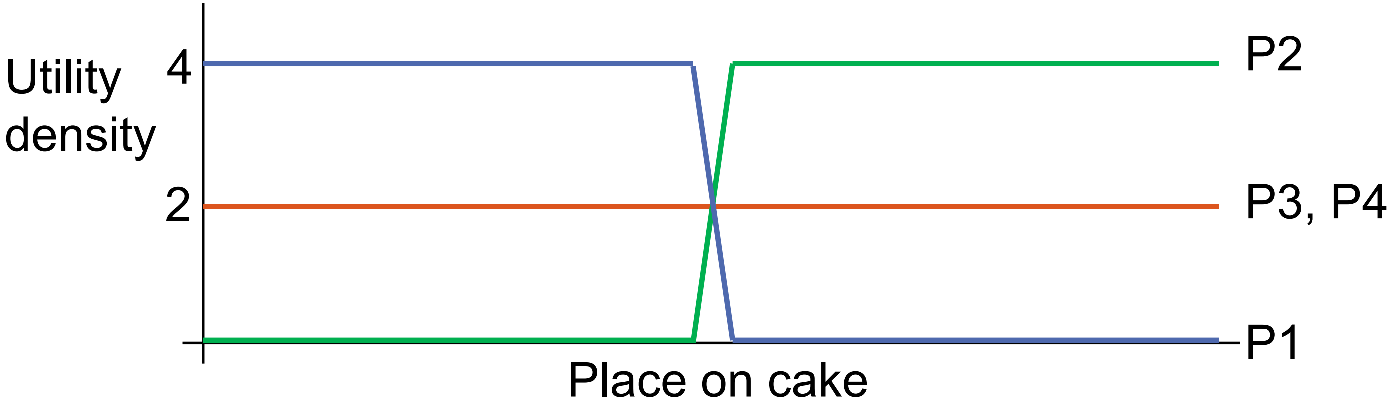
- **Tractability:** how efficient is the algorithm?
- **Truthfulness:** is it strategy-proof?

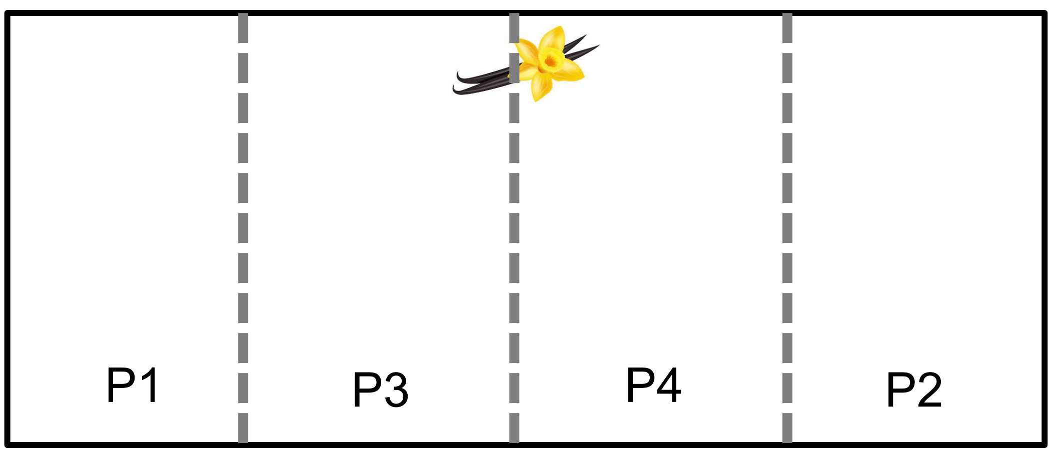


For each player, the cake can be segmented finitely such that its value over a single segment is uniform

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

SOCIAL WELFARE AND FAIRNESS





An envy-free allocation

Allocation social welfare: 3

Best solution social welfare: 4

Fairness ≠ Maximum Utility

THE PRICE OF FAIRNESS IN CAKE CUTTING

Given an instance:

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

utilitarian

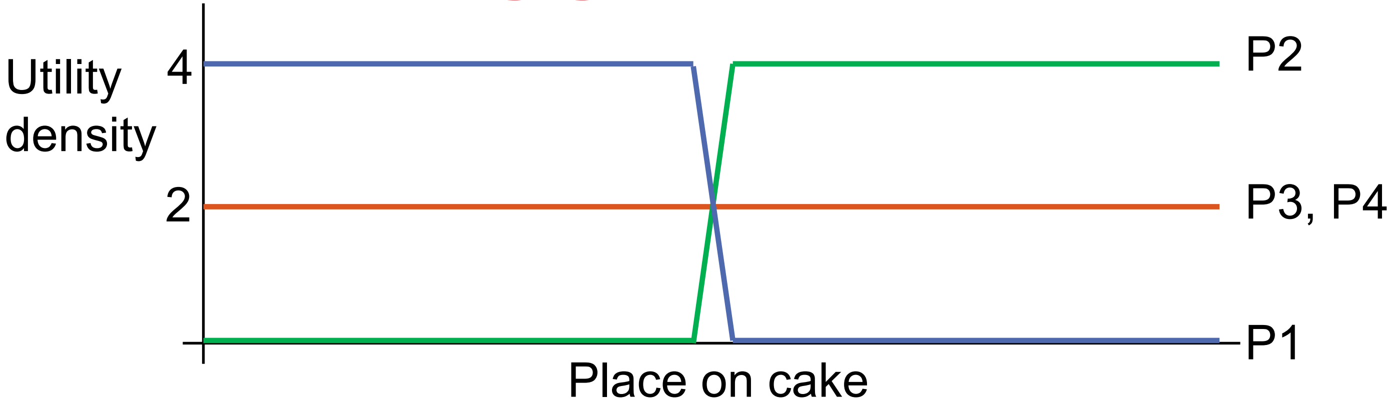
egalitarian (= minimum of players' utilities)

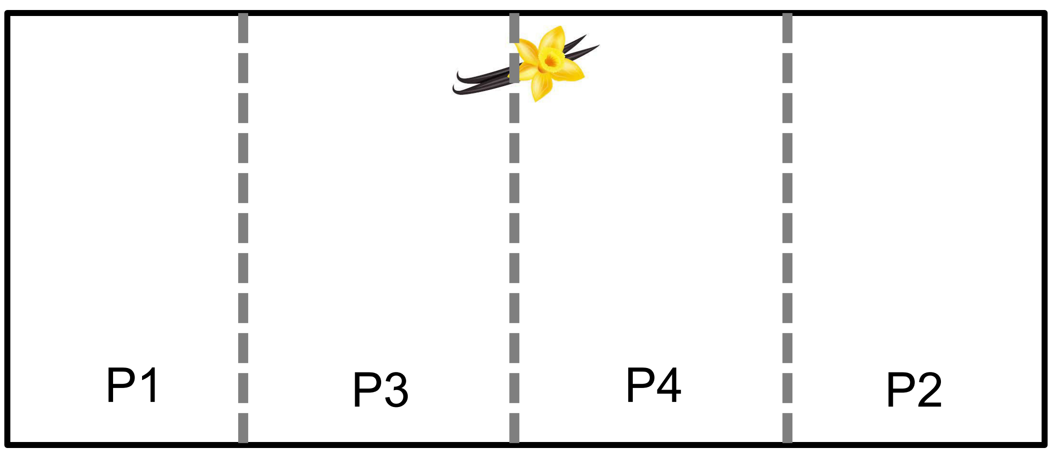
Price of envy-freeness

Price of equitability

Price of proportionality

SOCIAL WELFARE AND FAIRNESS





An envy-free allocation

Allocation social welfare: 3

Best solution social welfare: 4

Utilitarian Price of Envy-Freeness: 4/3

"PRICE OF" BOUNDS

[Aumann & Dombb]

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

APPLICATIONS: ESTATE DIVISION

River is a rich person with two children, Dave and Alex. In their will, River wishes to leave a fair allocation of their assets to their children. Their assets are as follows:

- A large mansion in upstate NY
- A second home in Florida
- A small café in Manhattan
- 2 Teslas
- 13 rare guitars

Note: these items are *indivisible*, whereas cakes are divisible. However, liquidation of one large asset can make many algorithms work to fairly divide these assets.

APPLICATIONS: ESTATE DIVISION

River is a rich person with two children, Dave and Alex. In their will, River wishes to leave a fair allocation of their assets to their children. Their assets are as follows:

- A large mansion in upstate NY - **liquidate**
- A second home in Florida - **A**
- A small café in Manhattan - **D**
- 2 Teslas - **A**
- 13 rare guitars - **D**

Note: these items are *indivisible*, whereas cakes are divisible. However, liquidation of one large asset can make many algorithms work to fairly divide these assets.

APPLICATIONS: LAND REFORM

Land reform: government-initiated property redistribution of (generally agricultural) land either between property holders or from individual ownership to collective ownership (or vice versa).

Relevant instances

- Egypt, 8th century BCE
- Irish Land Acts, 19th century
- Chinese Revolution, 1949
- Zimbabwe, 1980
- Much, much more

“At least 1.5 billion people today have some farmland as a result of land reform, and are less poor, or not poor, as a result.”

APPLICATIONS: LEGISLATION

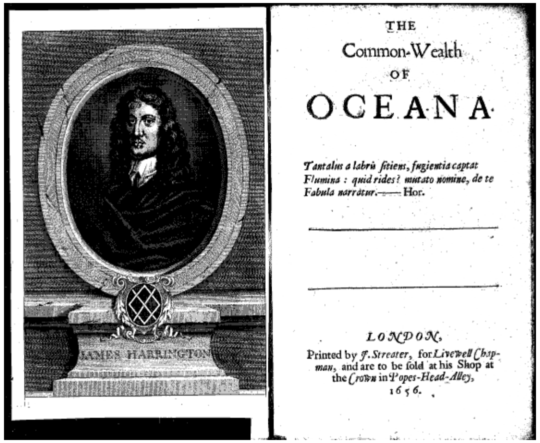
1656: James Harrington wrote the Commonwealth of Oceana in an attempt to argue political philosophy for use in English government.

Oceana government: (1) Senate of Knights, (2) House of deputies.

Passing bills: Senate proposes a legislation, and the house passes it. Divide and choose!

Note: this isn't division in the same sense,

but it is a highly related problem.



PART 2 – INDIVISIBLE GOODS

	Alex	Tasha	Jim
	100	500	5,000
	50,000	5,000	2,000
	1,000	3,000	2,000
	0	2,000	1,000

PART 2 – INDIVISIBLE GOODS

	Alex	Tasha	Jim
	100	500	5,000
	50,000	5,000	2,000
	1,000	3,000	2,000
	0	2,000	1,000

FAIRNESS WITH INDIVISIBILITY

Is there always a fair (proportional, envy-free) solution?

	Alex	Tasha	Jim
	50,000	5,000	2,000

FAIRNESS WITH INDIVISIBILITY

Is there always a fair (proportional, envy-free) solution?

	Alex	Tasha	Jim
	50,000	5,000	2,000

Slightly more generally, one *very* highly valued item can make our fairness notions impossible to achieve.

ENVY FREENESS UP TO ONE GOOD

Idea: We can't guarantee no one will envy anyone else. BUT, we can ensure envy if we remove a high-valued item.

	Alex	Tasha	Jim
	100	500	5,000
	50,000	50,000	50,000
	1,000	3,000	2,000
	0	2,000	1,000

Now everyone loves the car

ENVY FREENESS UP TO ONE GOOD

In the previous allocation, Tasha and Jim envied Alex because Alex got the expensive car. Whoever gets the car will be envied.

Not envy free ☹

Is it envy free up to one good (EF1)?

Removing the car, neither Jim nor Tasha envy Alex.

No one else envies anyone else.

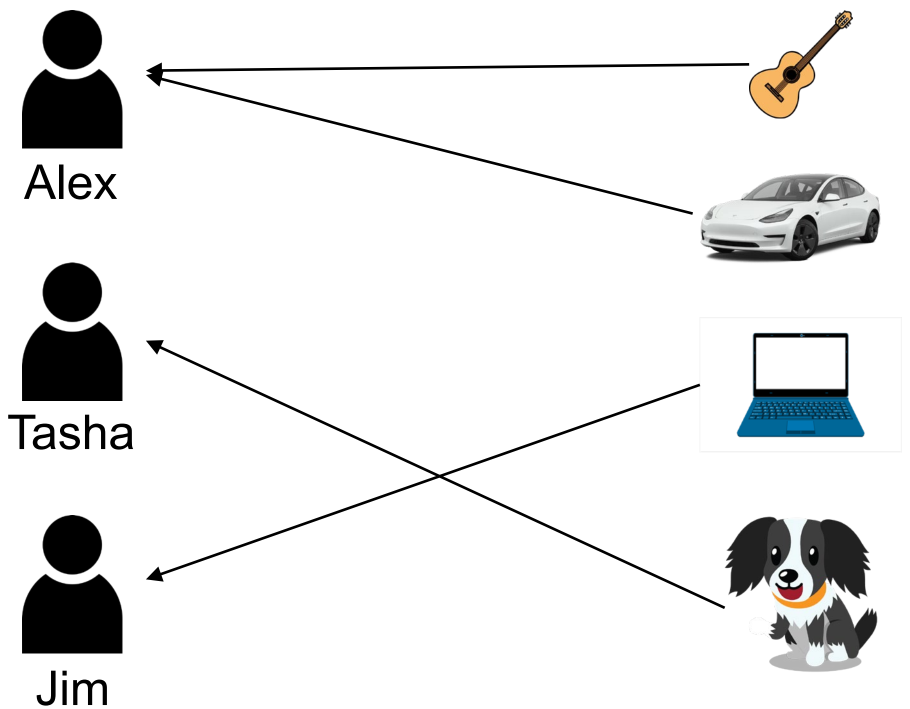
So yes! ☺

What is EF1 formally?

An allocation is EF1 if for any two agents *a* and *b*, there is some item you can take away from *a* such that *b* will not envy *a*.

ROUND ROBIN ALGORITHM FOR EF1

1. List out your players
2. In order, let the players pick their favorite item
3. Repeat until there are no items



ROUND ROBIN ALGORITHM FOR EF1

Is this EF1?

Clearly, Alex does not even envy Tasha.

Say we want to see if Tasha envies Alex.

Remove the first item Alex took.

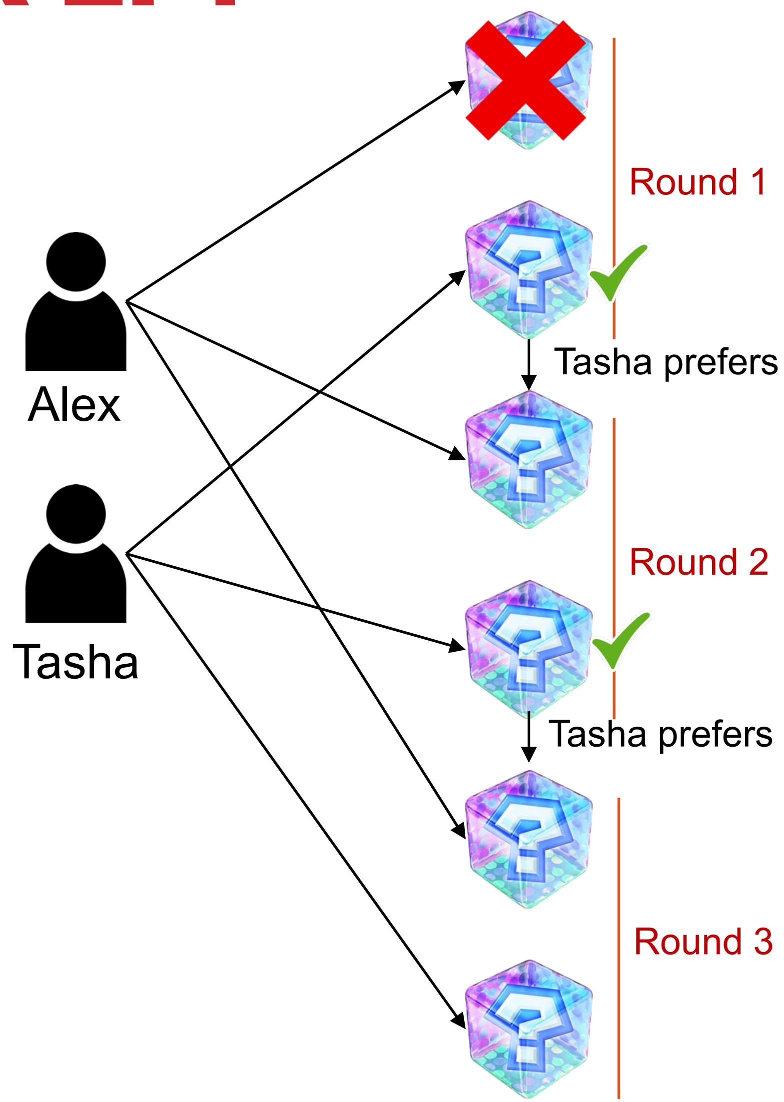
Tasha prefers her first item to Alex's second.

Tasha prefers her second item to Alex's first.

So on….

Clearly, Tasha does not envy Alex when this item is removed.

QED

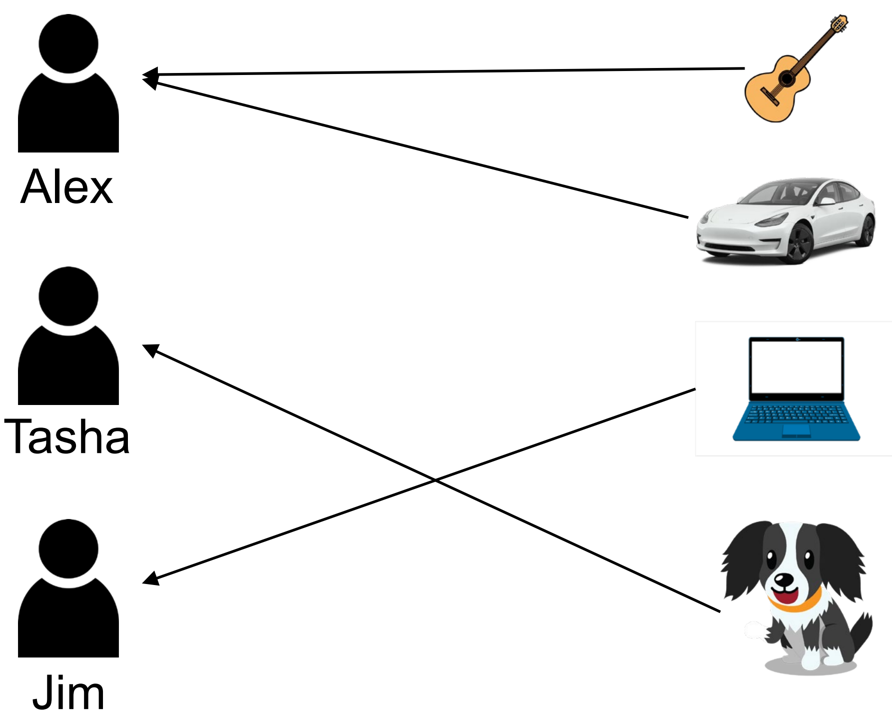


EF1, EFX ON INDIVISIBLE GOODS

Round robin always finds an EF1 solution.

This means there always exists an EF1 allocation ☺

But that was too easy… and maybe too weak.

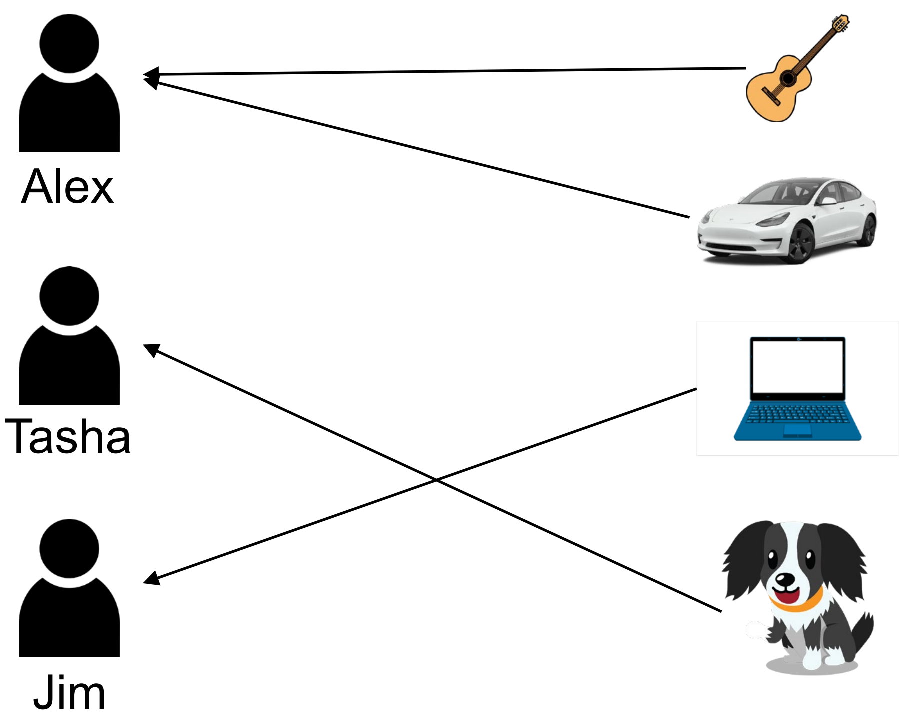


Alex got the “highly desirable” car and still got to pick first the second round! ☹

ENVY FREENESS UP ANY ONE GOOD

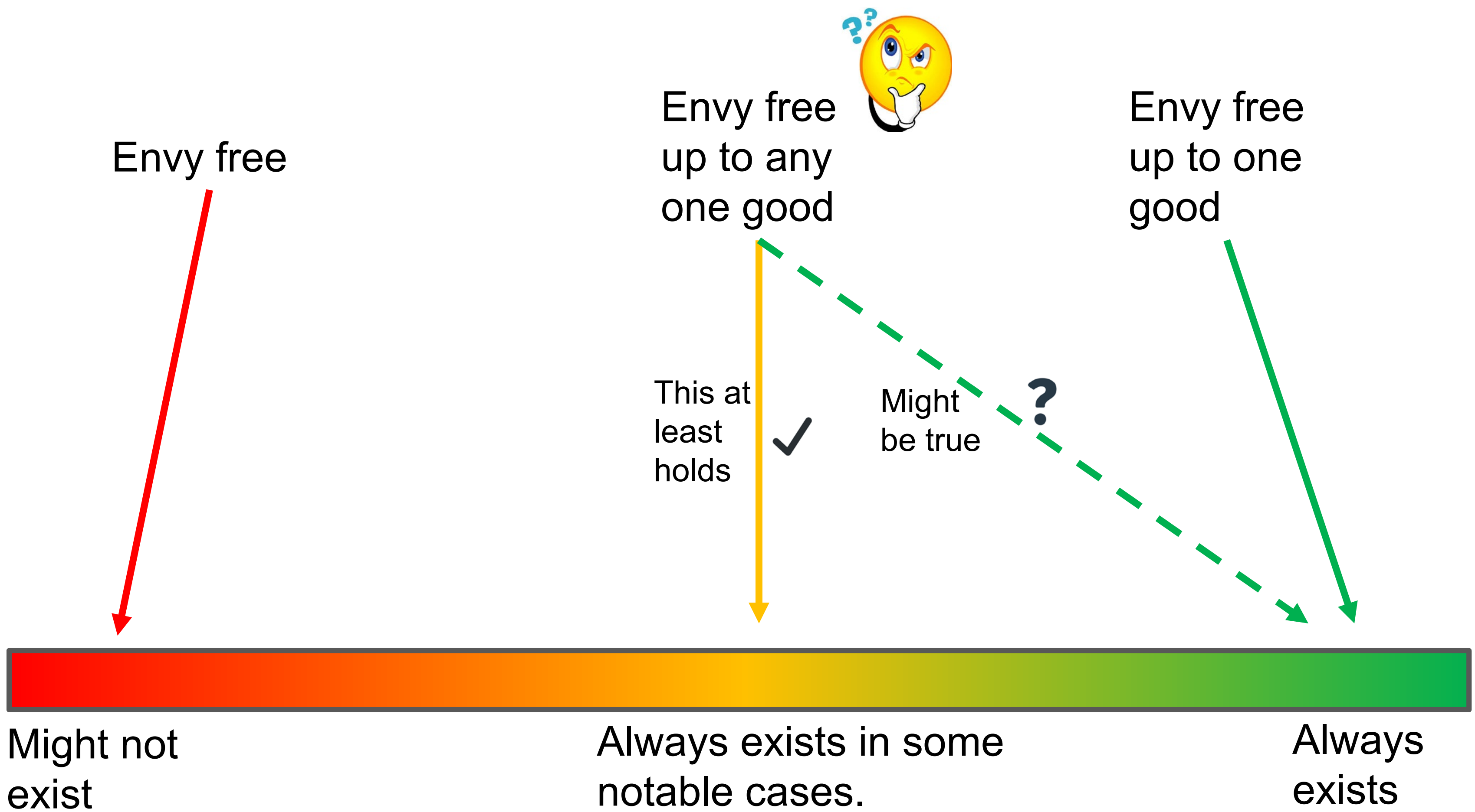
Envy freeness up to *any* 1 good (EFX):

An allocation is EFX if for any two agents *a* and *b*, **for any** item you can take away from *a*, *b* will not envy *a*.



This is no longer fair: if we remove the guitar from Alex’s allocation, Tasha and Jim still envy him.

EXISTENCE OF EF, EF1, EFX



THE LEXIMIN-COMPARE ALGORITHM (2 AGENTS)

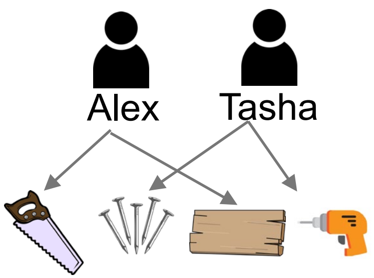
Given two allocations A, B:

1. Order the players by how much they like their bundle (least first).

2. Pop of the first in each list.

3. Find their value of the allocation.

	Saw	Nails	Wood	Drill
Alex	2	0	3	5
Tasha	3	1	2	4



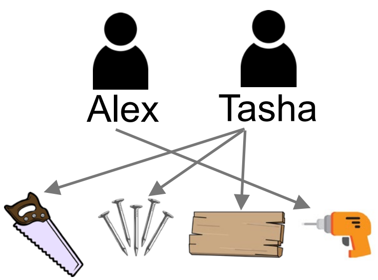
Allocation A

Alex's value: 5

Tasha's value: 5

[Tasha, Alex]

Tasha



Allocation B

Alex's value: 5

Tasha's value: 6

[Alex, Tasha]

Alex

THE LEXIMIN-COMPARE ALGORITHM (2 AGENTS)

Given two allocations A, B:

1. Order the players by how much they like their bundle (least first)

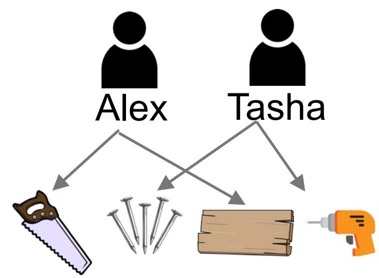
2. Pop off the first in each list.

3. Find their value of the allocation.

4. If one value is better, output that allocation. Else, continue popping and comparing.

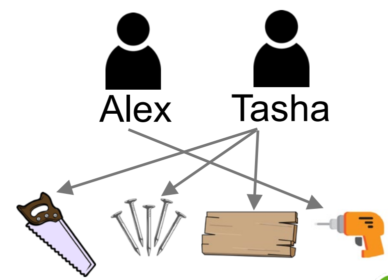
Idea: Find the best of all possible allocations

	Saw	Nails	Wood	Drill
Alex	2	0	3	5
Tasha	3	1	2	4



Allocation A
Alex's value: 5
Tasha's value: 5

[Tasha, Alex]
Tasha -> 5
Alex -> 5



Allocation B
Alex's value: 5
Tasha's value: 6

[Alex, Tasha]
Alex -> 5
Tasha -> 6

=
<

Allocation B is "better than" allocation A

DOES LEXIMIN FIND AN EFX SOLUTION?

	{0}	{Saw}	{Nails}	{Saw, Nails}
Alex	0	0	1	2
Tasha	0	0	1	2

Warning: This is not additive.

All best leximin allocations:

Alex gets {saw, nails}, Tasha gets nothing ✗

Tasha gets {saw, nails}, Alex gets nothing ✗

An EFX solution: Alex gets {saw}, Tasha gets {nails} (or vice versa)

THE LEXIMIN-COMPARE++ ALGORITHM (2 AGENTS)

Given two allocations A, B:

1. Order the players by how much they like their bundle (least first)

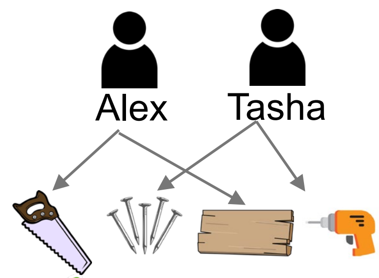
2. Pop of the first in each list.

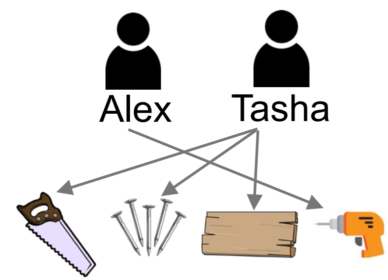
3. Find their value of the allocation.

4. If one value is better, output that allocation. If one allocation is bigger, output that allocation. Else, continue popping and comparing.

Idea: Find the best of all possible allocations

	Saw	Nails	Wood	Drill
Alex	2	0	3	5
Tasha	3	1	2	4





Allocation A		Allocation B
Alex's value: 5 Tasha's value: 5		Alex's value: 5 Tasha's value: 6
[Tasha, Alex] Tasha -> 5 2 items	= >	[Alex, Tasha] Alex -> 5 1 item

Allocation A is "better than" allocation B

THE LEXIMIN++ SOLUTION IS EFX ON IDENTICAL VALUATIONS

Identical valuations: everyone values items the same
Strongly envies: envies when you remove some item, e.g., breaks EFX

We want to show: Leximin++ implies EFX

Equivalent to contrapositive: Not EFX implies not Leximin++

Assume allocation A is not EFX.

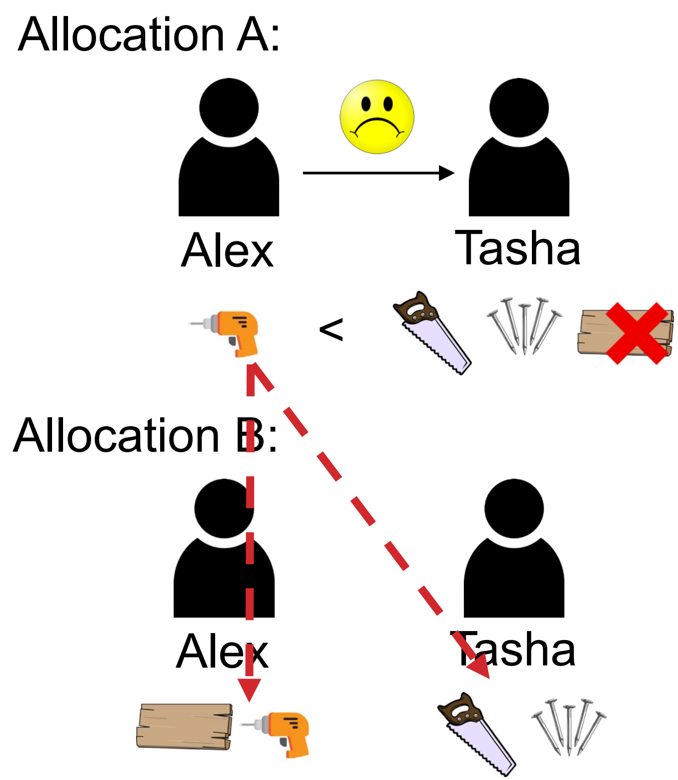
Someone, say Alex, *strongly envies* the other, Tasha. Then, removing, say wood, from Tasha, Tasha's allocation is still better.

- Implies {drill} < {nails, saw}

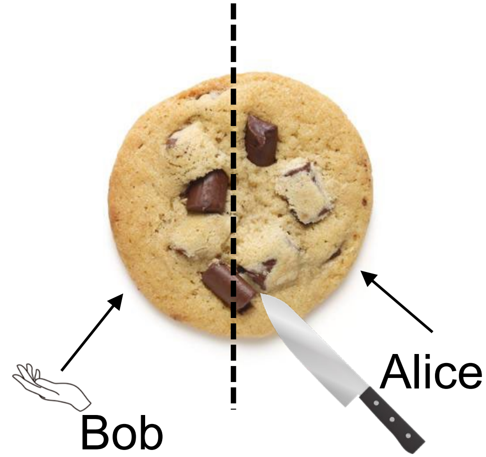
Create allocation B: it's like A, but Alex gets the wood.

A < B under Leximin++:

- Alex has the worst allocation in A.
    - It's worse than Tasha's in B since {nails, saw} > {drill}
    - It's worse than Alex's in A since he gets another item.

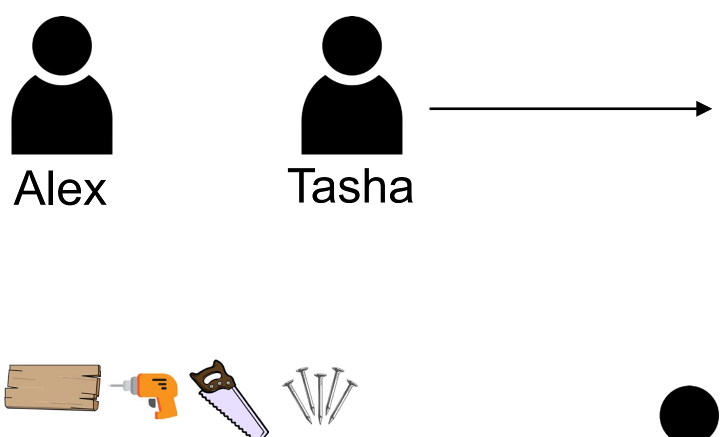


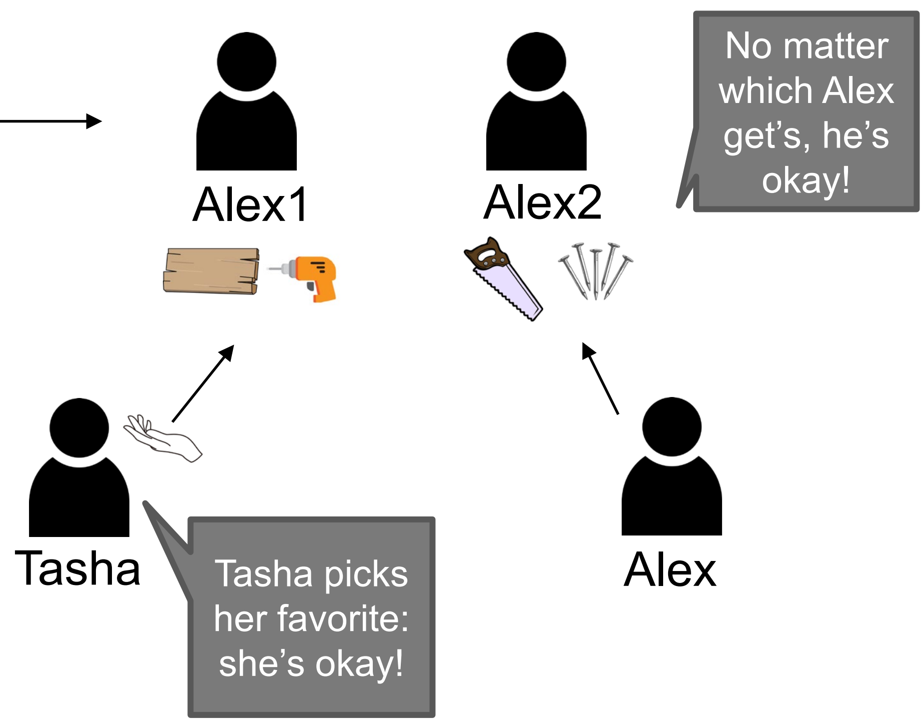
EXTENDING TO 2 PLAYERS, DIFFERENT VALUATIONS



Method: Divide and choose!

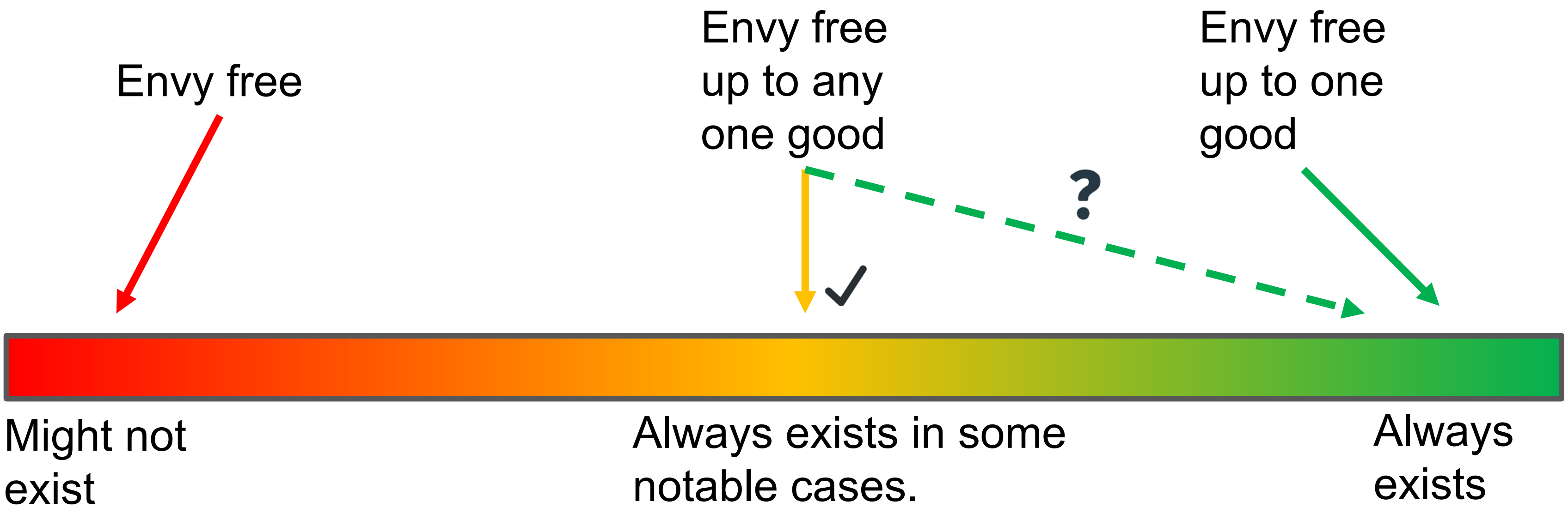
1. Alex finds a Leximin++ solution assuming Tasha had his valuation.
2. Tasha picks her favorite bundle, Alex gets the other.





Tasha doesn't envy Alex.
Alex doesn't strongly envy Tasha.

RECAP: EF, EF1, EFX



We know EFX exists when:

- There are 3 players [Chaudhury, Garg, and Mehlhorn]
    - We saw EFX for 2 players [Plaut & Roughgarden]
- All agents have identical valuations [Plaut & Roughgarden]
- All agents have binary, additive valuations [Halpern, Procaccia, Psomas, & Shah]
- There are 4 players and we can remove one item [Berger, Cohen, Feldman, & Fiat]
- There are n players and we can remove n-1 items [Berger, Cohen, Feldman, & Fiat]