Fair Allocations for Smoothed Utilities

Yushi Bai, Uriel Feige, Paul Gölz & Ariel Procaccia

Fair Item Allocation

5	5	0	4
2	1	2	2
2	1	1	3

Setting:
- indivisible goods
- additive utilities

Envy Freeness (EF)

	5	5	0	4
	2	1	2	2
	2	1	1	3

Setting:
- indivisible goods
- additive utilities

Envy Freeness (EF)

	5	5	0	4
	2	1	2	2
	2	1	1	3

Setting:
- indivisible goods
- additive utilities

values their own bundle at 2,

Envy Freeness (EF)

5	5	0	4
2	1	2	2
2	1	1	3

Setting:
- indivisible goods
- additive utilities

values their own bundle at 2, but values ’s bundle at 1+3=4

Envy Freeness (EF)

	5	5	0	4
	2	1	2	2
	2	1	1	3

Setting:
- indivisible goods
- additive utilities

values their own bundle at 2, but values ’s bundle at $1+3=4$ $\Rightarrow$ envies .

Envy Freeness (EF)

5	5	0	4
2	1	2	2
2	1	1	3

Setting:
- indivisible goods
- additive utilities

When Do Envy Free Allocations Exist?

When Do Envy Free Allocations Exist?

In worst case, envy-free allocations needn't exist:

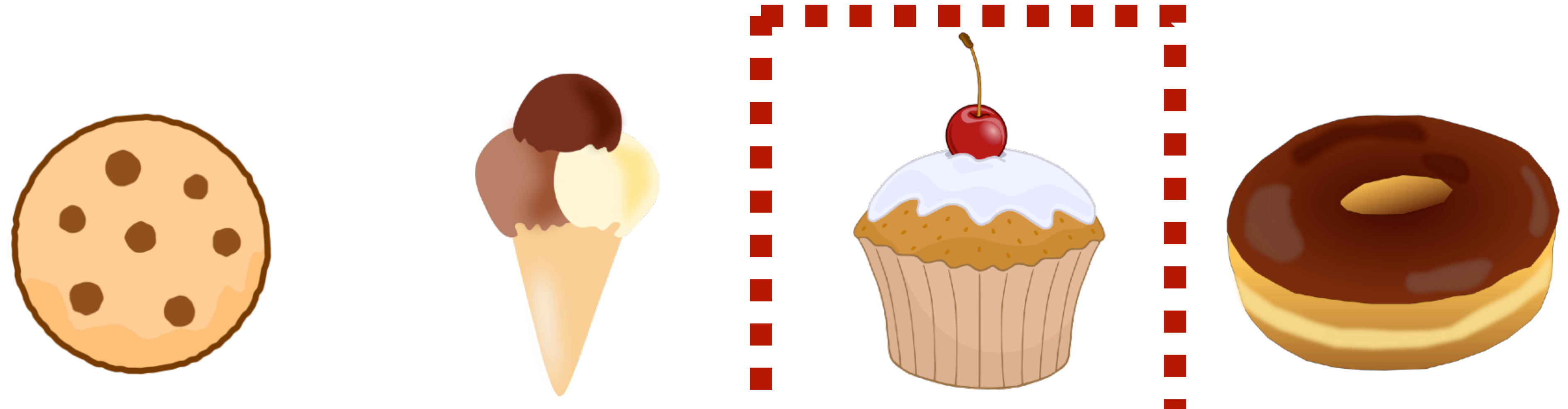
When Do Envy Free Allocations Exist?

In worst case, envy-free allocations needn't exist:

				
	0	0	10	0
	1	0	9	0

When Do Envy Free Allocations Exist?

In worst case, envy-free allocations needn't exist:

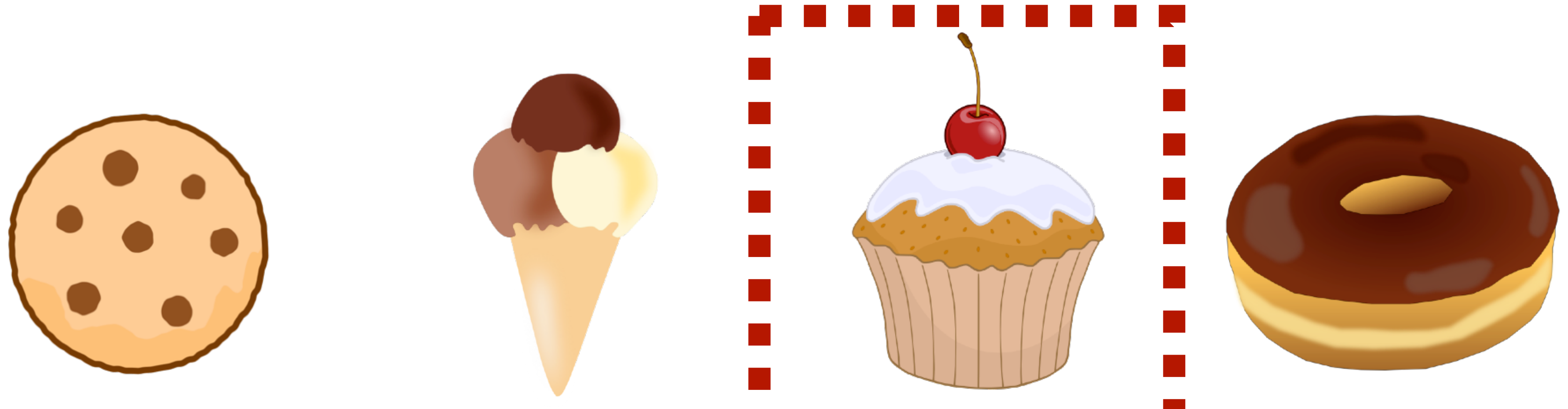


	0	0	10	0
	1	0	9	0

But they seem to exist "most of the time."

When Do Envy Free Allocations Exist?

In worst case, envy-free allocations needn't exist:

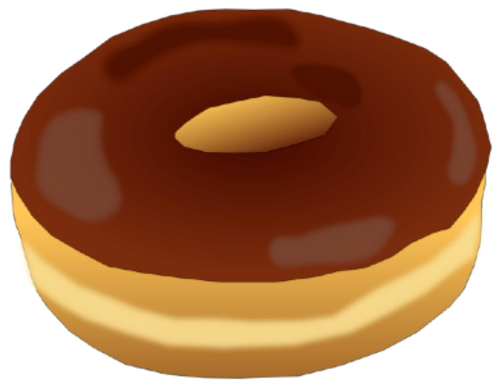


	0	0	10	0
	1	0	9	0

But they seem to exist "most of the time."

How to capture this mathematically?

Previous Work: i.i.d. Utilities

				
	$\sim\mathcal{D}$	$\sim\mathcal{D}$	$\sim\mathcal{D}$	$\sim\mathcal{D}$
	$\sim\mathcal{D}$	$\sim\mathcal{D}$	$\sim\mathcal{D}$	$\sim\mathcal{D}$
	$\sim\mathcal{D}$	$\sim\mathcal{D}$	$\sim\mathcal{D}$	$\sim\mathcal{D}$

Previous Work: i.i.d. Utilities

$\sim \mathscr{D}$	$\sim \mathscr{D}$	$\sim \mathscr{D}$	$\sim \mathscr{D}$
$\sim \mathscr{D}$	$\sim \mathscr{D}$	$\sim \mathscr{D}$	$\sim \mathscr{D}$
$\sim \mathscr{D}$	$\sim \mathscr{D}$	$\sim \mathscr{D}$	$\sim \mathscr{D}$

Theorem [MS21]. If $m \in \Omega(n \log n / \log \log n)^*$, envy-free allocations exist in i.i.d. model with probability $\to 1$.

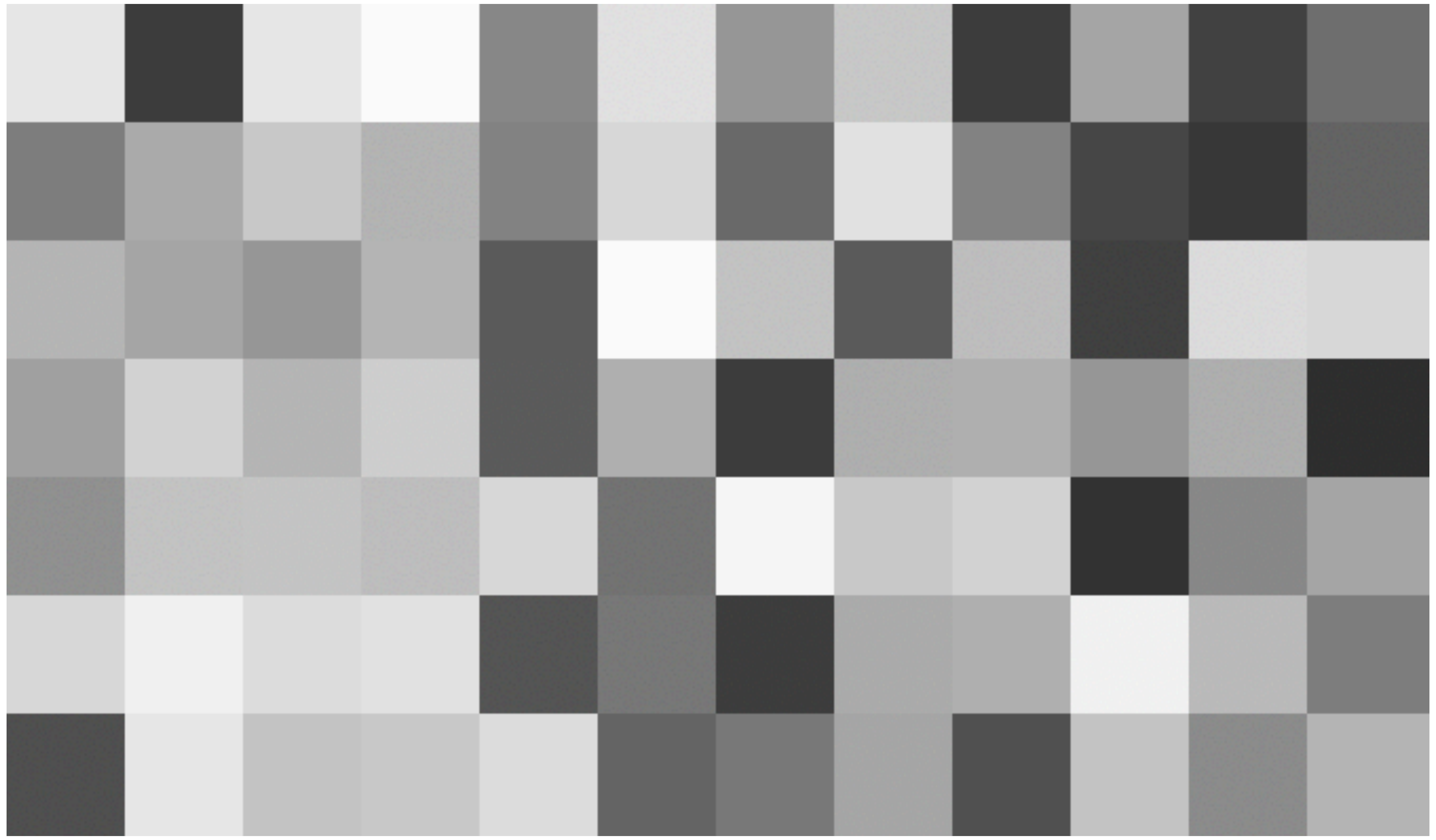
n: number of agents
m: number of items

* under some assumption on $\mathscr{D}$

Manurangsi, P. & Suksompong, W. Closing Gaps in Asymptotic Fair Division. *SIAM J. Discrete Math.* 35, 668–706 (2021).

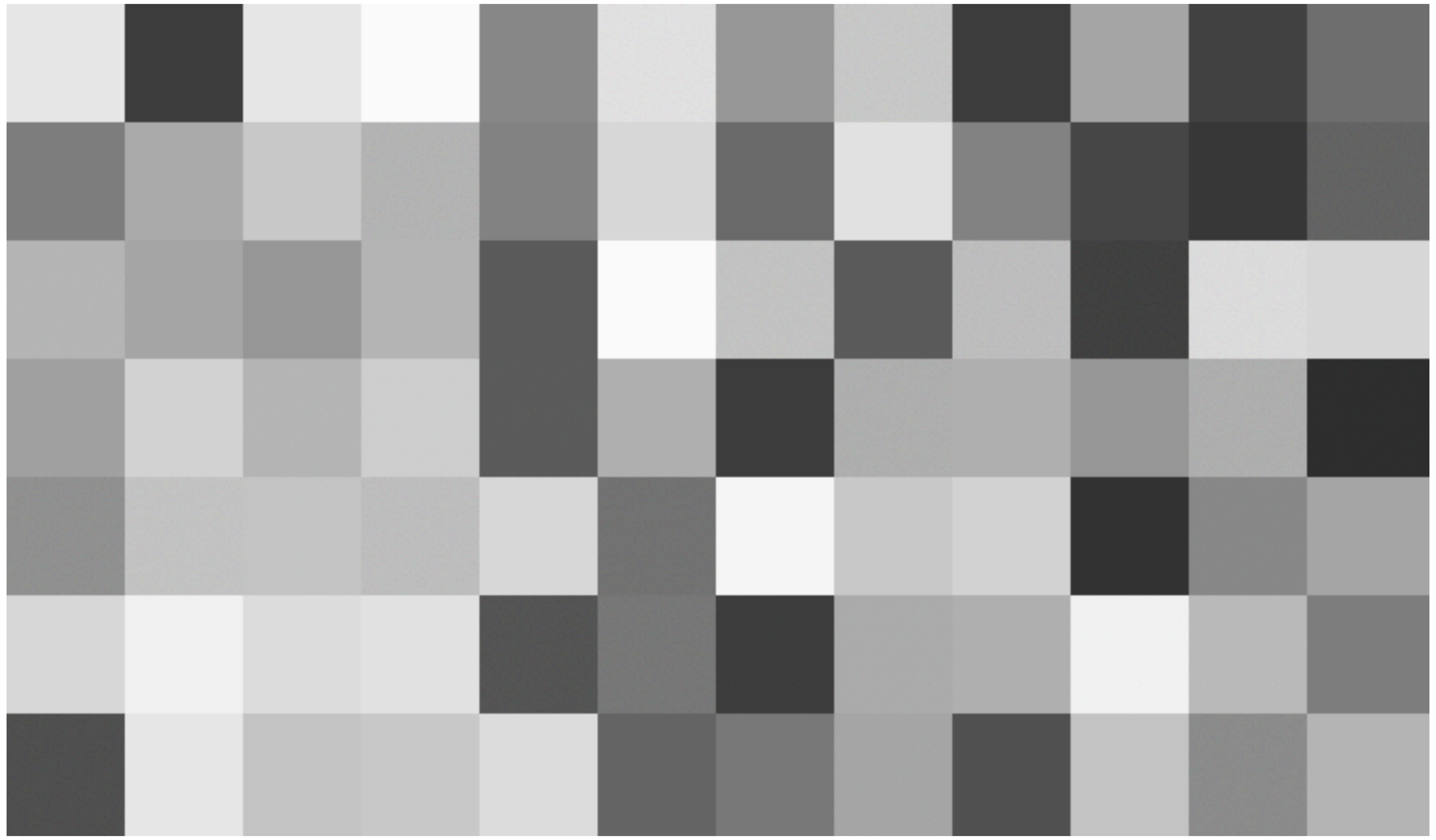
Limitations of Previous Work

i.i.d. values

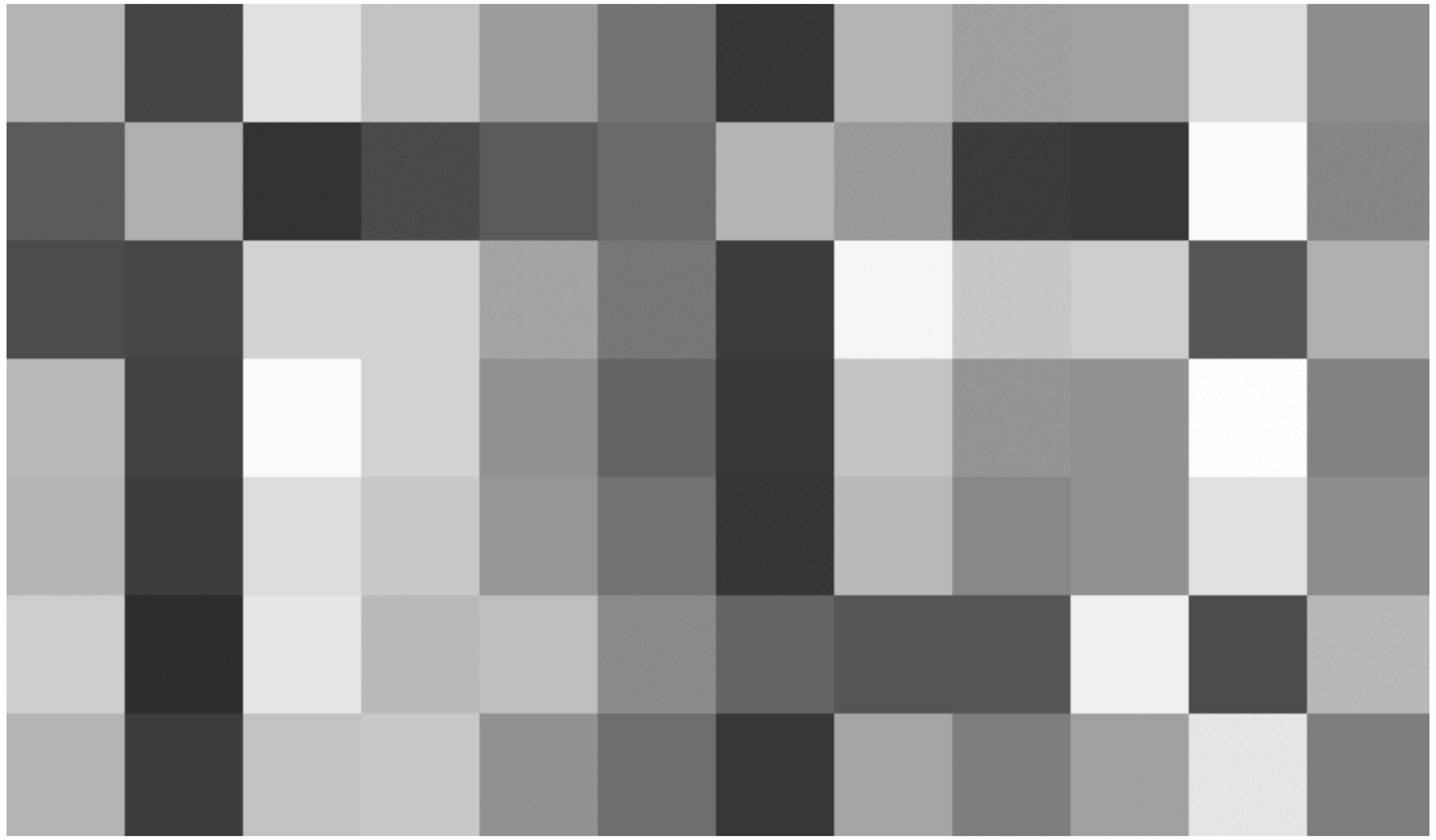


Limitations of Previous Work

i.i.d. values

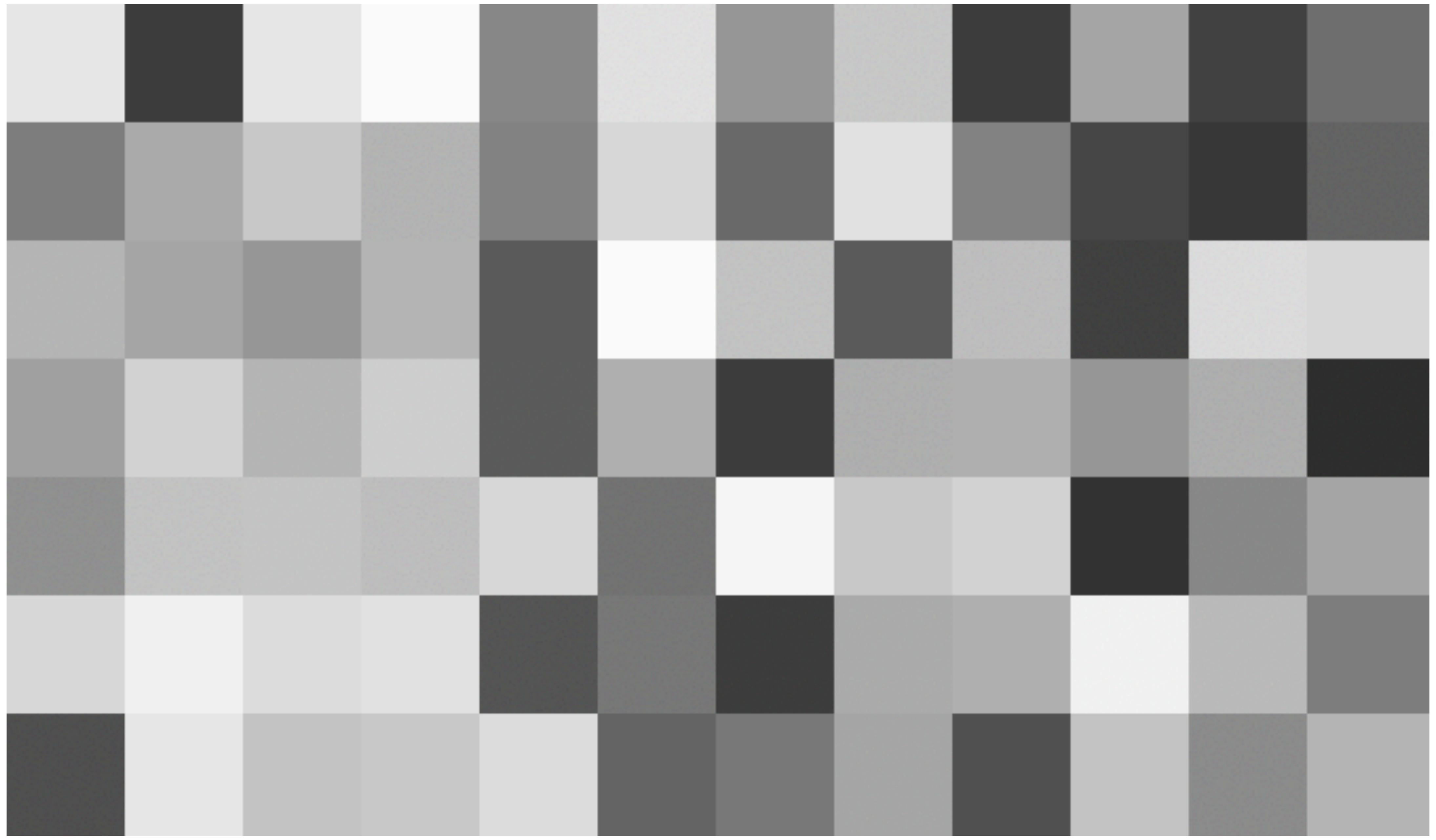


values with structure

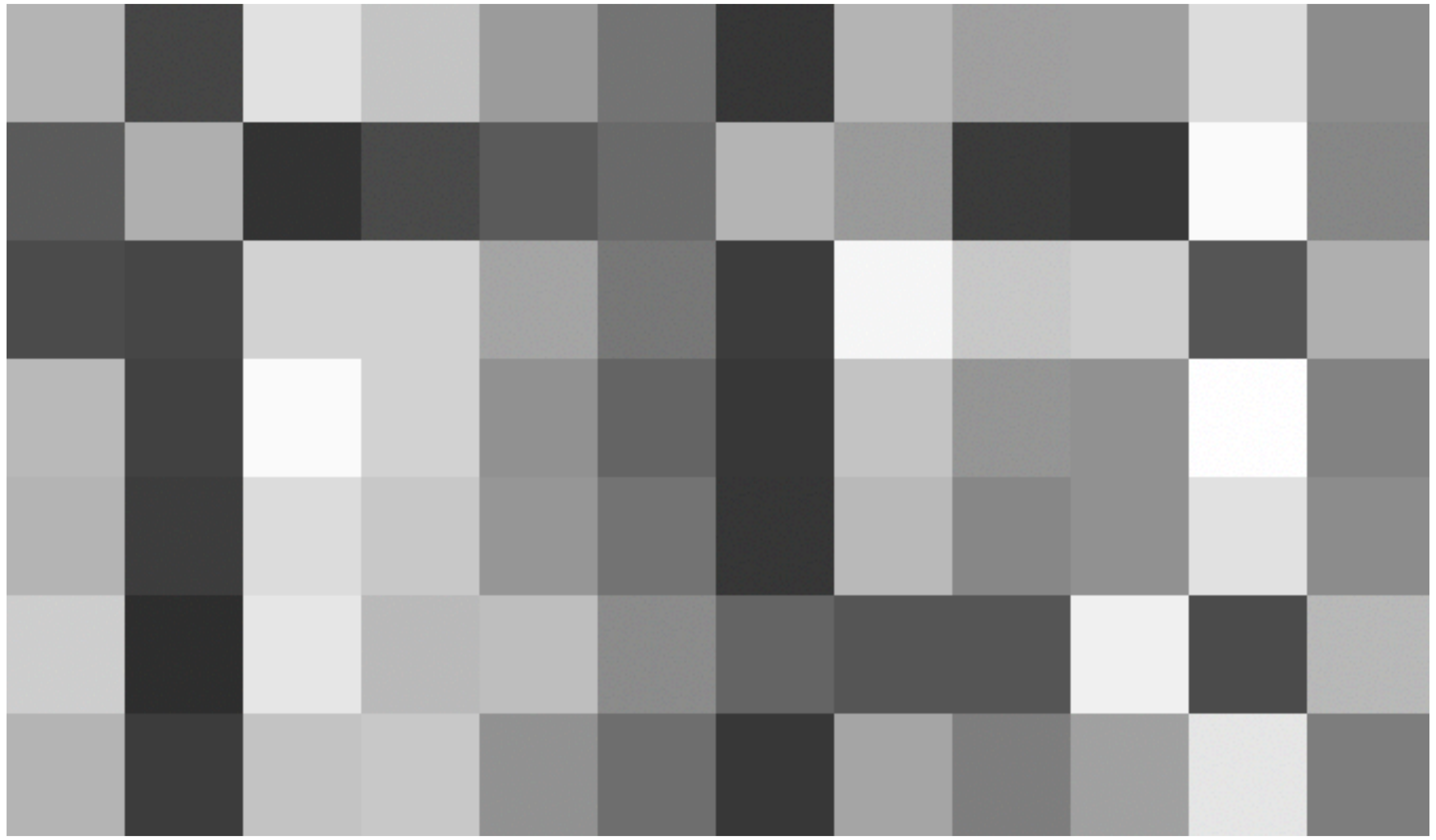


Limitations of Previous Work

i.i.d. values



values with structure



Structure like on the right unlikely in i.i.d. models. If practice looks like right-hand side, limits relevance of i.i.d. results.

Our Model: Smoothed Analysis

Our Model: Smoothed Analysis

4	5	0	4
2	4	2	2
2	5	1	3

1. Start from worst-case *base values.*

Our Model: Smoothed Analysis

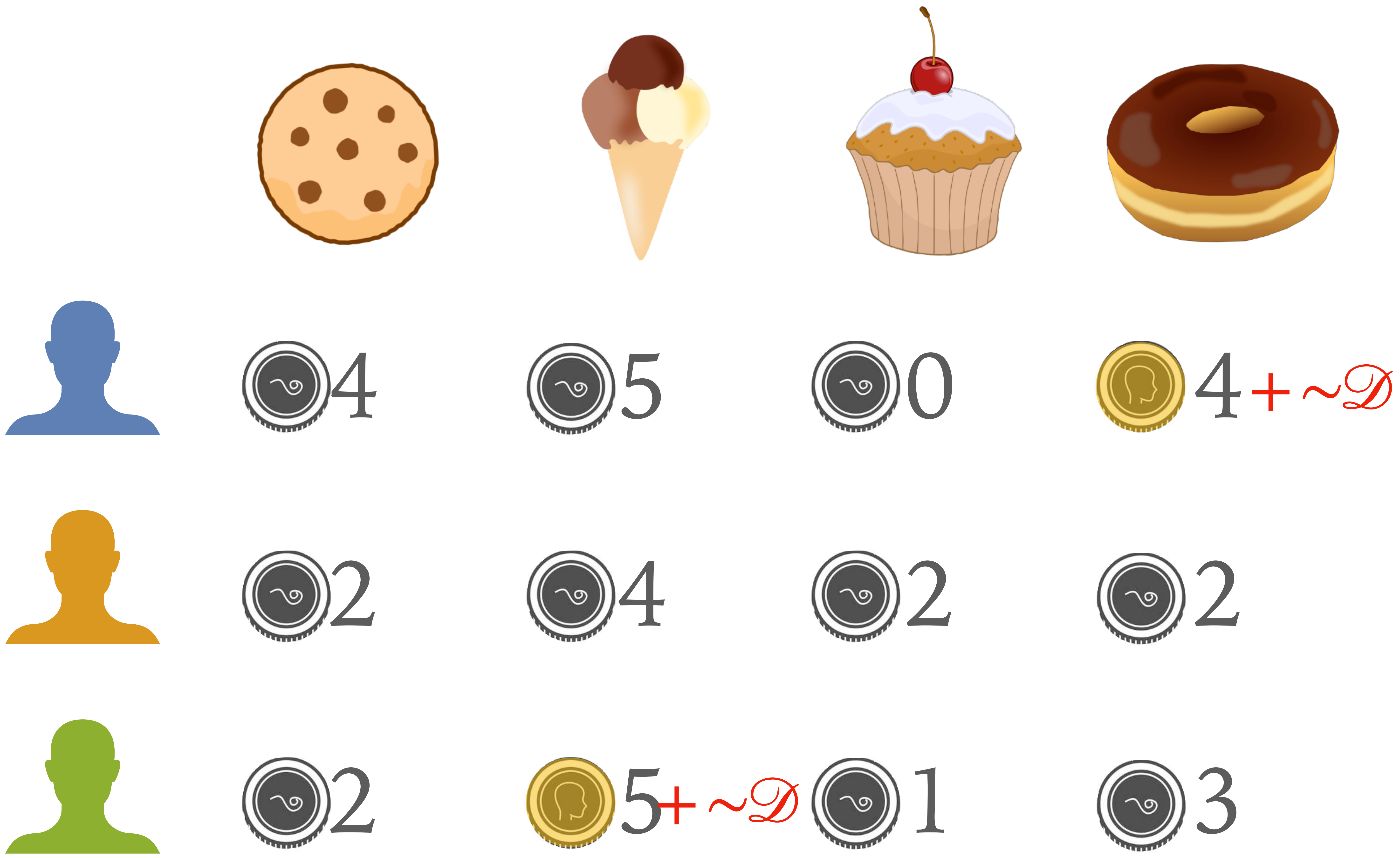
4	5	0	4
2	4	2	2
2	5	1	3

1. Start from worst-case *base values*.
2. For each entry, flip biased coin with probability $p \ll 1$.

Our Model: Smoothed Analysis

1. Start from worst-case *base values.*
2. For each entry, flip biased coin with probability $p \ll 1$.

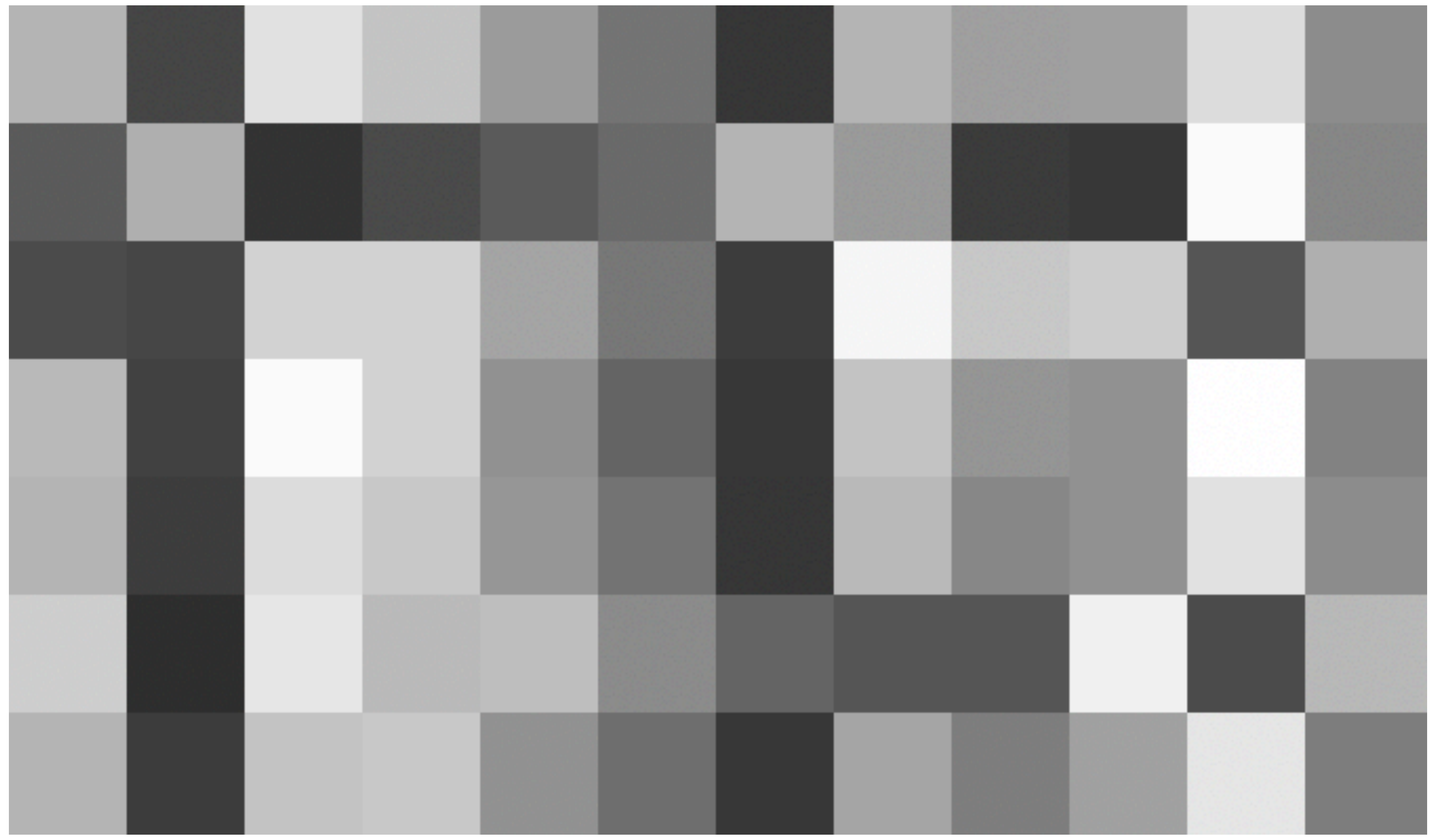
Our Model: Smoothed Analysis



1. Start from worst-case *base values*.
2. For each entry, flip biased coin with probability $p \ll 1$.
3. Where coin comes up heads, “boost” value by amount $\sim \mathscr{D}$.

Our Model: Smoothed Analysis

base utilities



Our Model: Smoothed Analysis

base utilities

smoothed
utilities

Our Model: Smoothed Analysis

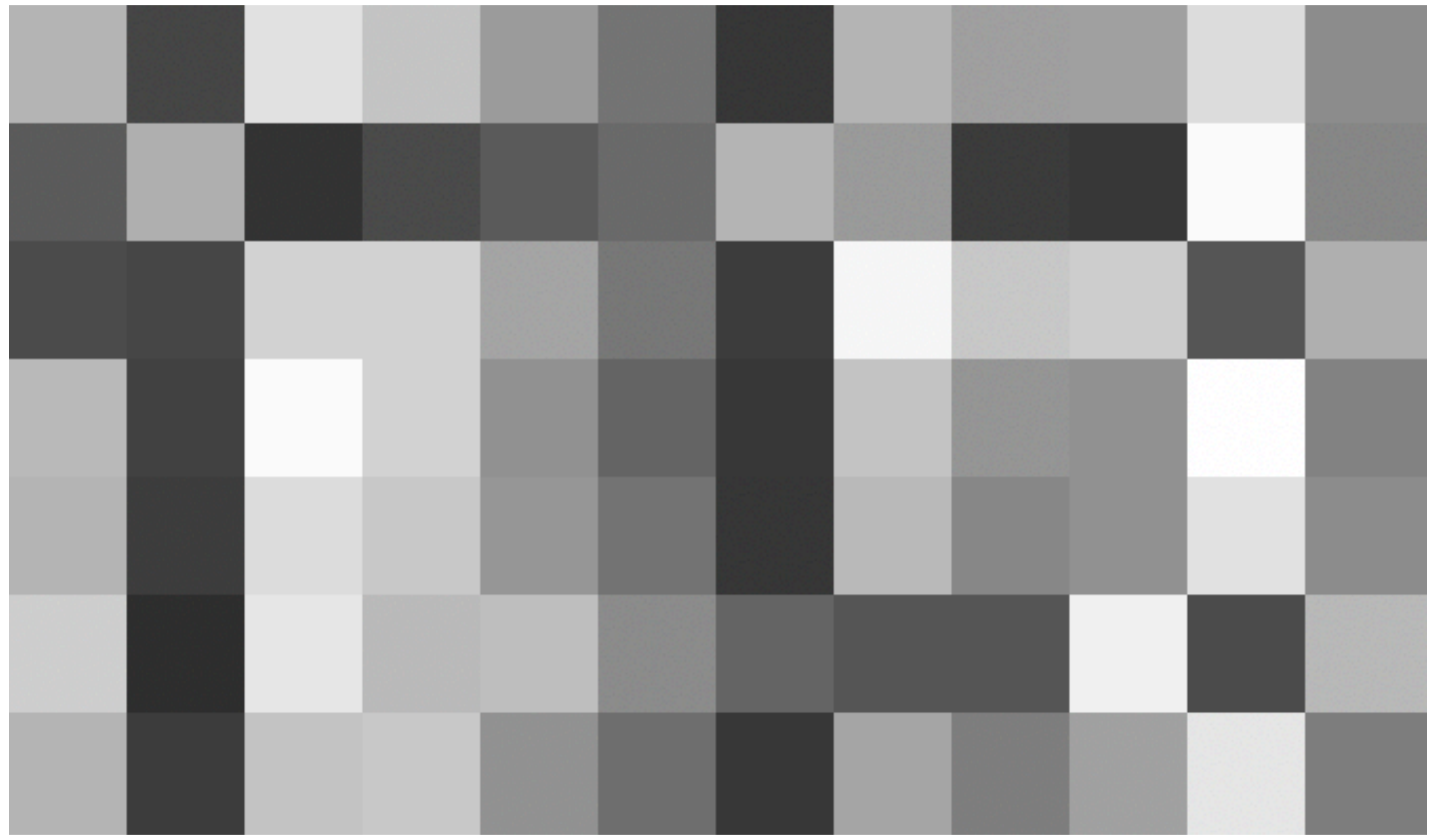
base utilities

smoothed
utilities

with high probability,
envy-free allocation exists

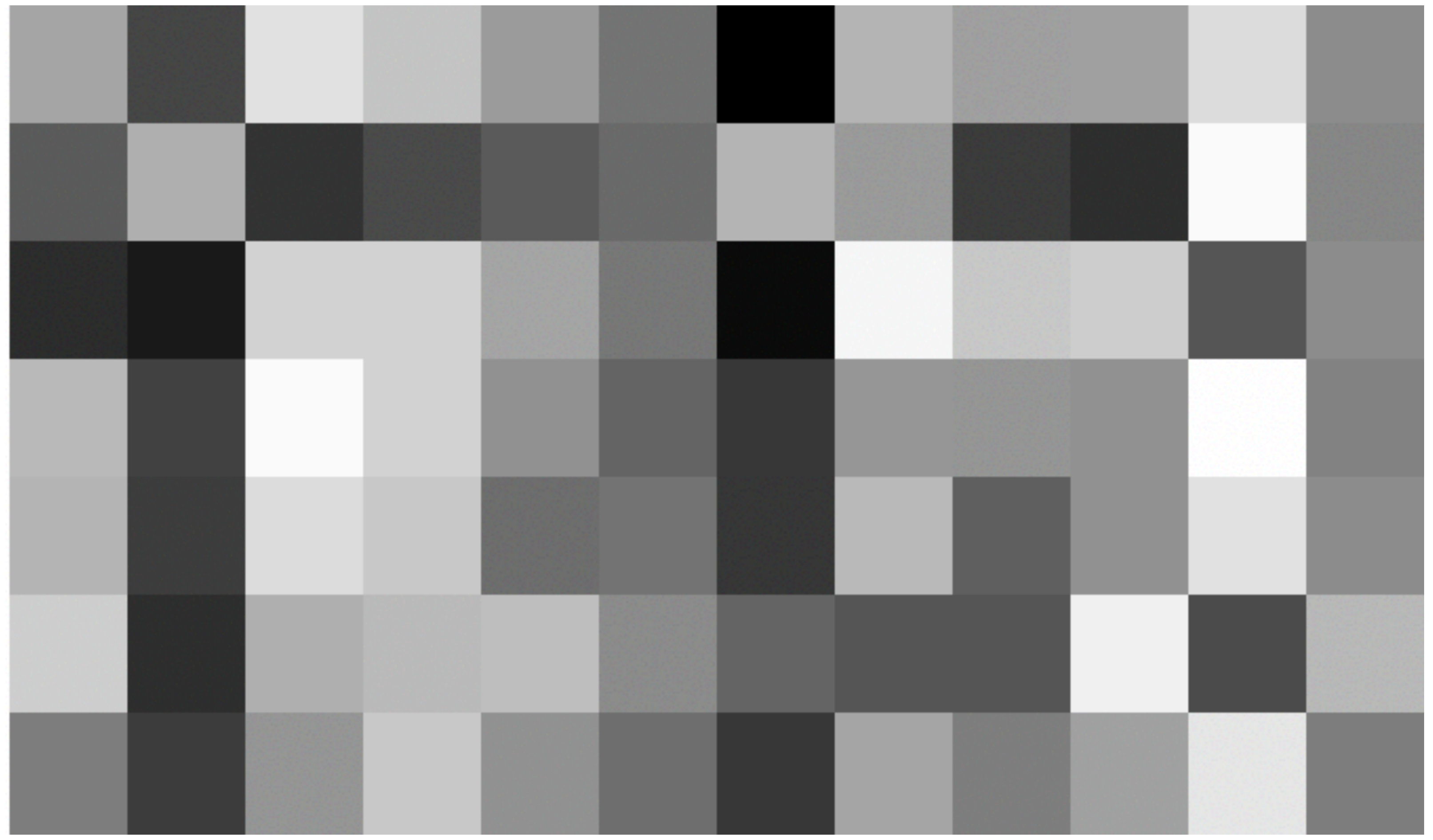
Our Model: Smoothed Analysis

base utilities



random
boosts

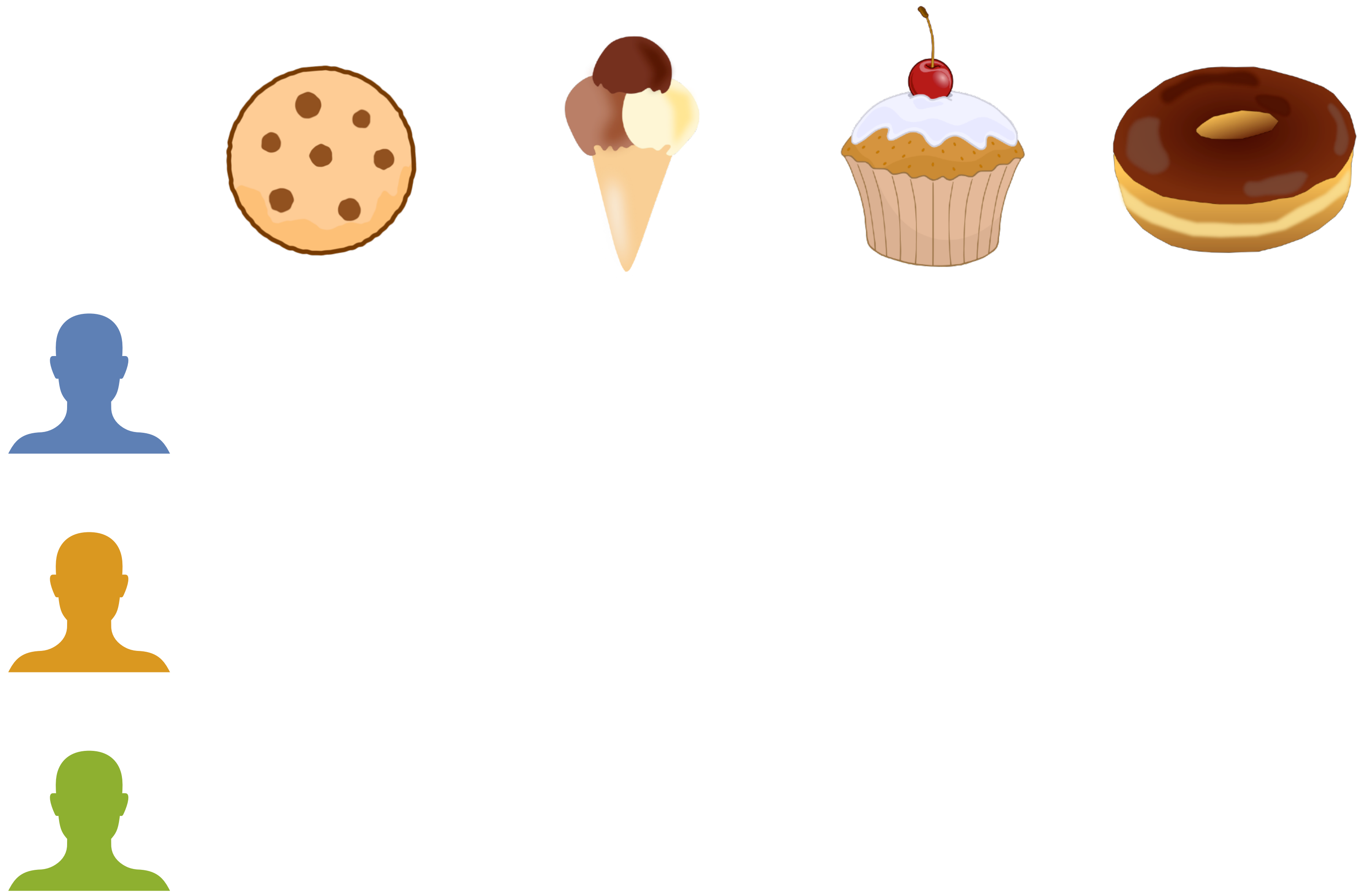
smoothed
utilities



Theorem. If $m \gg n \log n$ and $p \gg \log n/m$, envy-free allocations exist in smoothed model with probability $\to 1$.

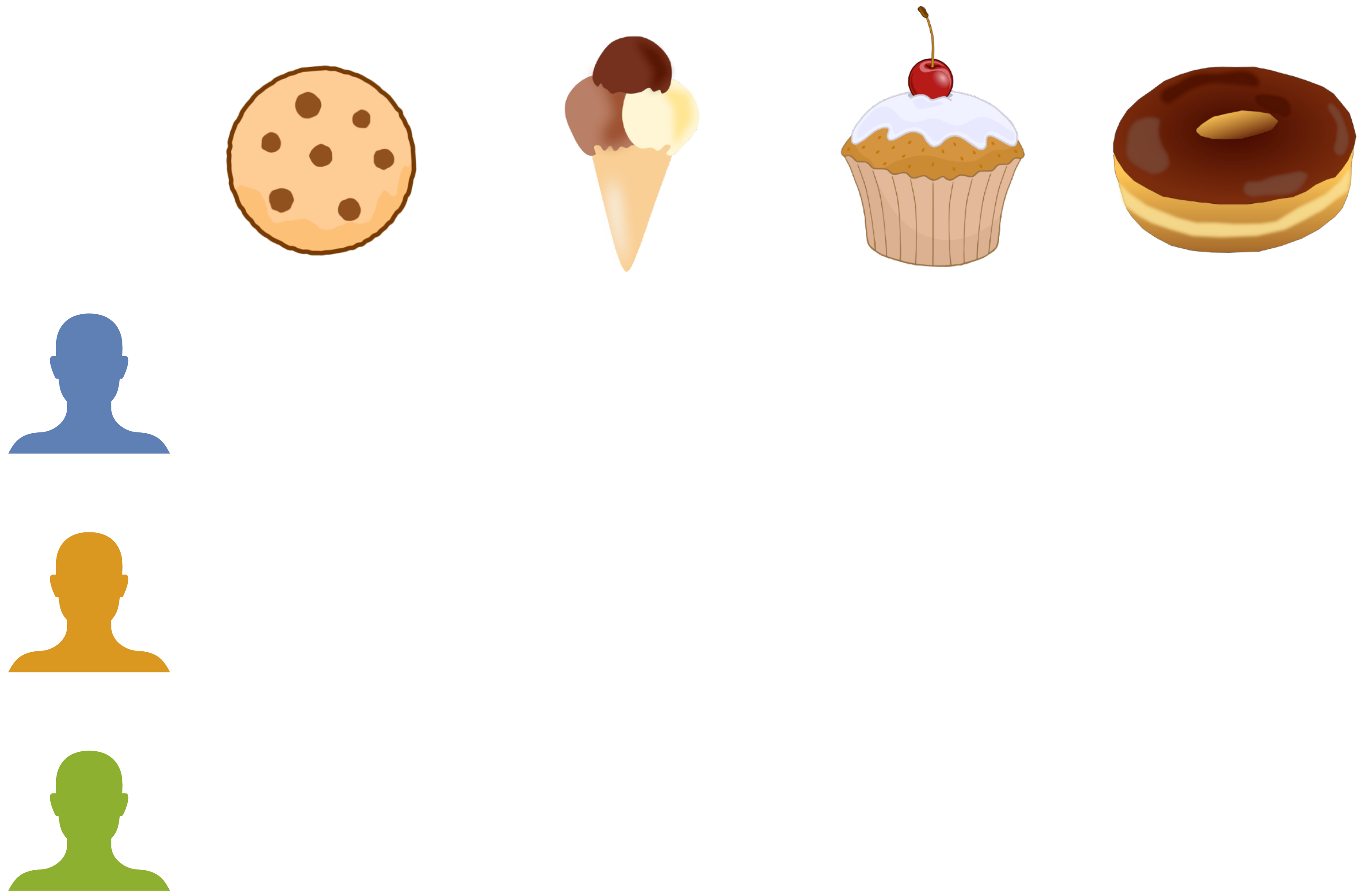
n: number of agents
m: number of items
p: boost probability

Algorithm for Finding EF Allocations w.h.p.



Algorithm for Finding EF Allocations w.h.p.

for $\rho = 1, 2, 4, \ldots, m$:



Algorithm for Finding EF Allocations w.h.p.

for $\rho = 1, 2, 4, \ldots, m$:

- with LP, allocate ρ fractional items such that $t = \min_{\text{agents } i \neq j} u_i(A_i) - u_i(A_j)$ is maximized.

i's utility for own bundle i's utility for j's bundle

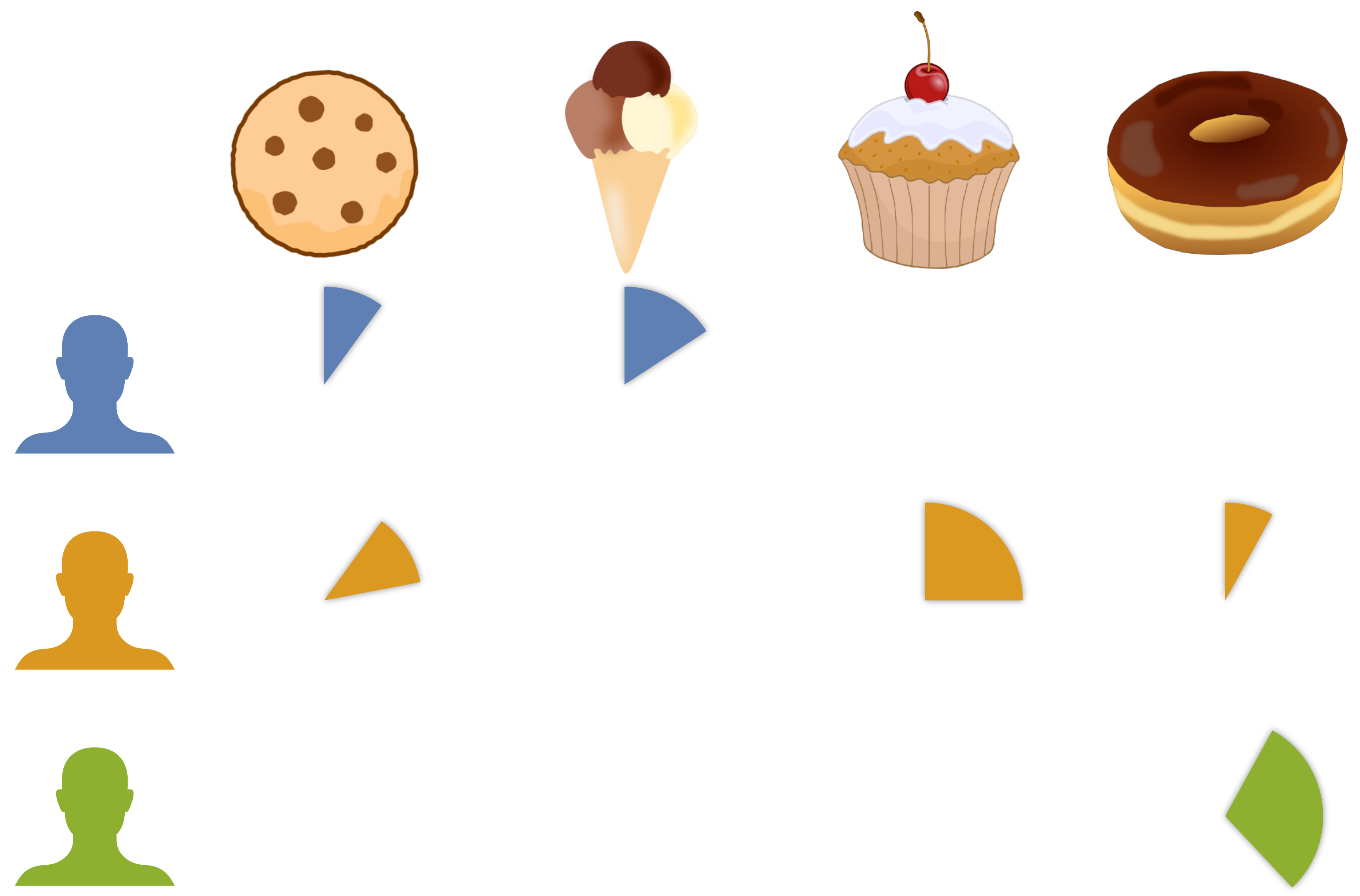
Algorithm for Finding EF Allocations w.h.p.

for $\rho = 1, 2, 4, \ \ldots, m$:

- with LP, allocate ρ fractional items such that $t = \min_{\text{agents } i \neq j} u_i(A_i) - u_i(A_j)$ is maximized.

i's utility for own bundle

i's utility for *j*'s bundle

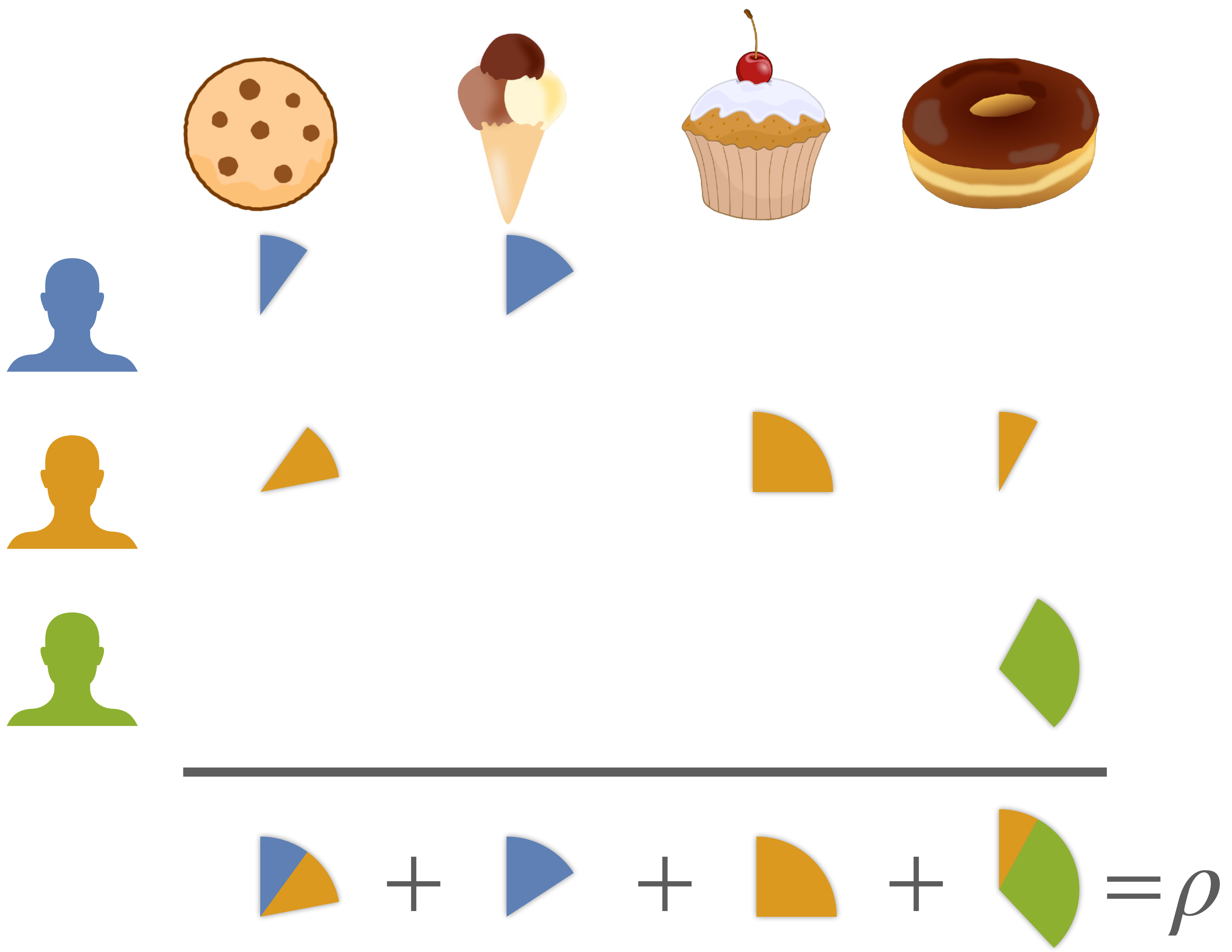


Algorithm for Finding EF Allocations w.h.p.

for $\rho = 1, 2, 4, \ldots, m$:

- with LP, allocate ρ fractional items such that $t = \min_{\text{agents } i \neq j} u_i(A_i) - u_i(A_j)$ is maximized.

i's utility for own bundle   *i*'s utility for *j*'s bundle

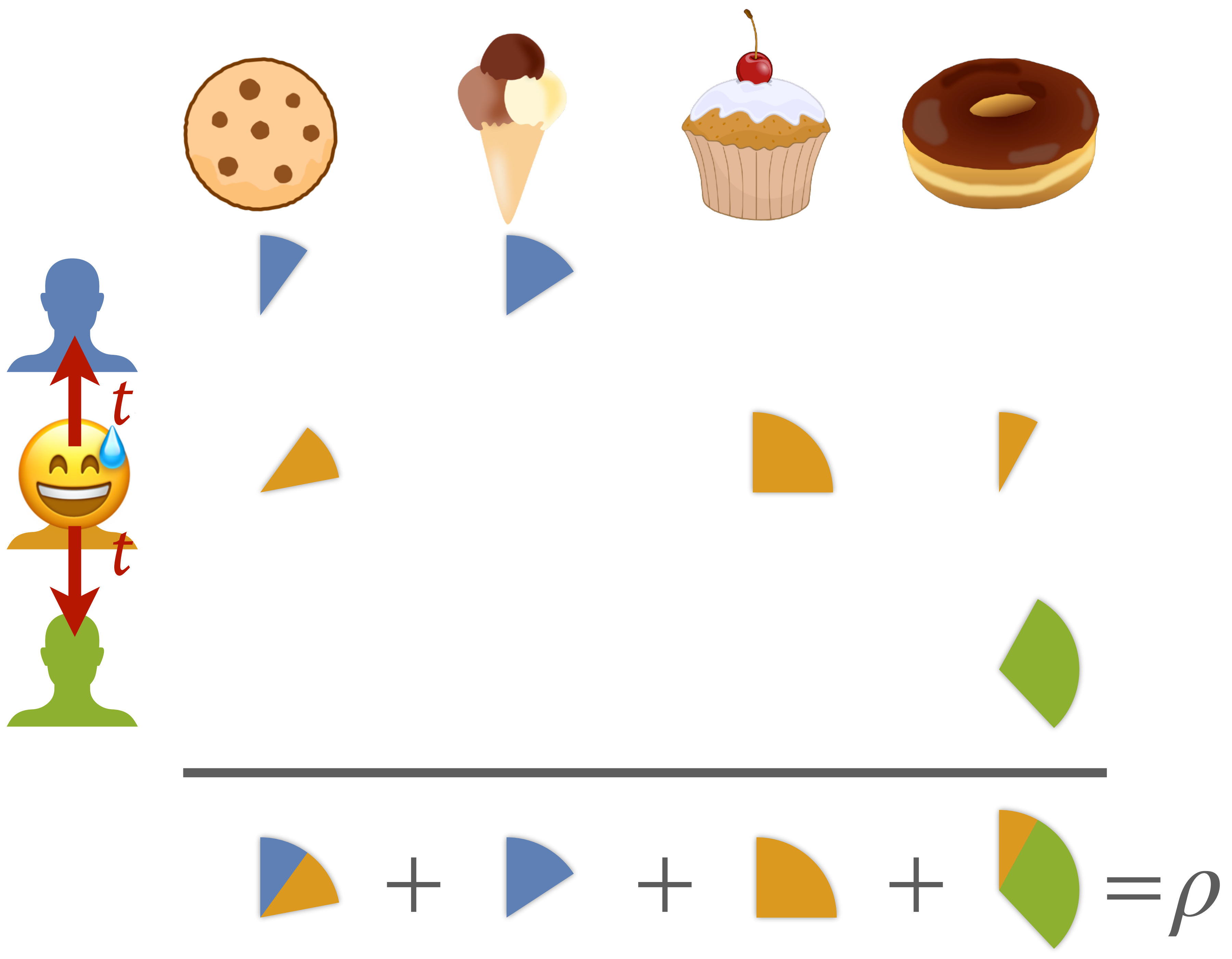


Algorithm for Finding EF Allocations w.h.p.

for $\rho = 1, 2, 4, \ldots, m$:

- with LP, allocate ρ fractional items such that $t = \min_{\text{agents } i \neq j} u_i(A_i) - u_i(A_j)$ is maximized.

i's utility for own bundle   *i*'s utility for *j*'s bundle



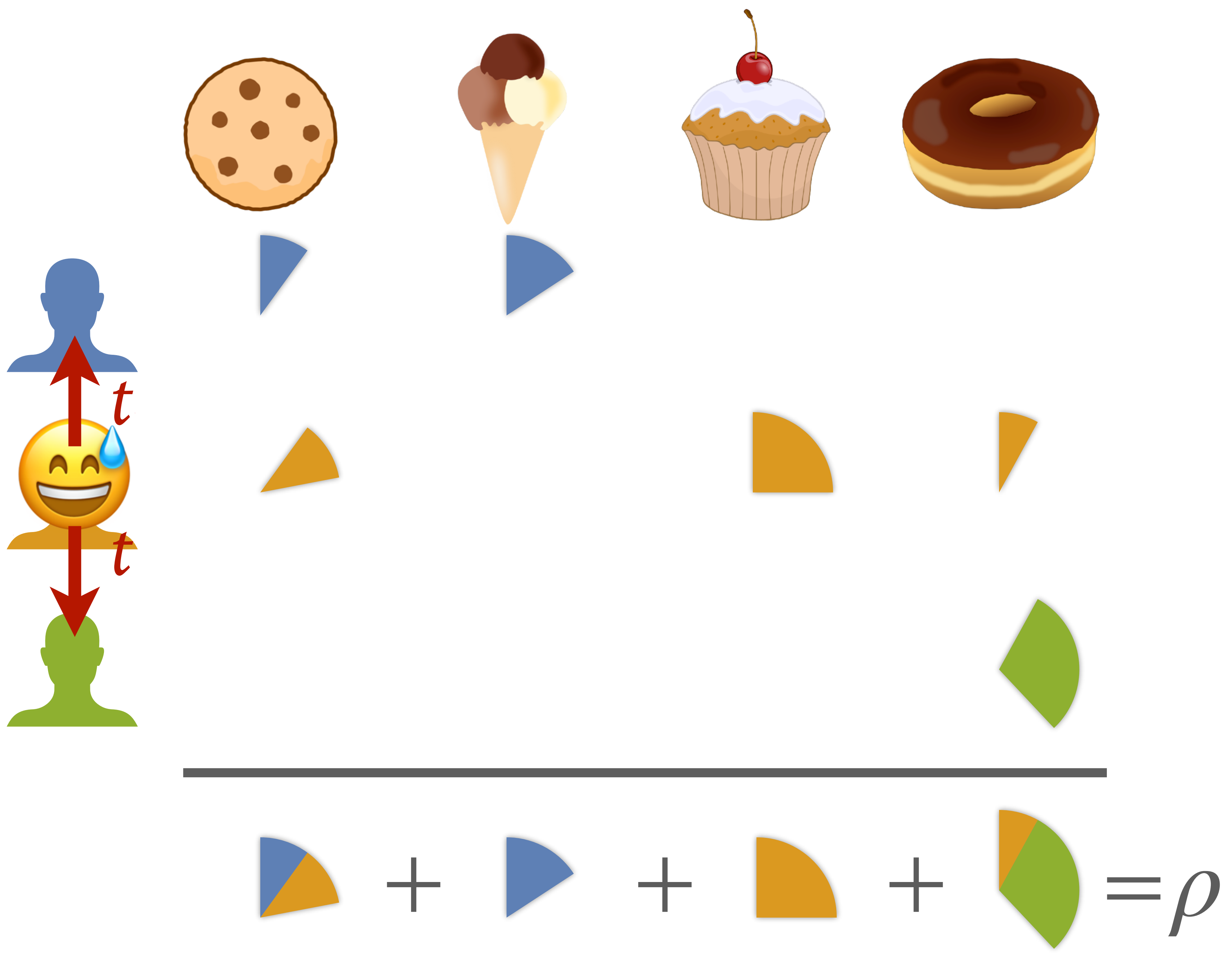
Algorithm for Finding EF Allocations w.h.p.

for $\rho = 1, 2, 4, \ldots, m$:

- with LP, allocate ρ fractional items such that $t = \min_{\text{agents } i \neq j} u_i(A_i) - u_i(A_j)$ is maximized.

  i's utility for own bundle *i*'s utility for *j*'s bundle

- randomly round this fractional allocation.



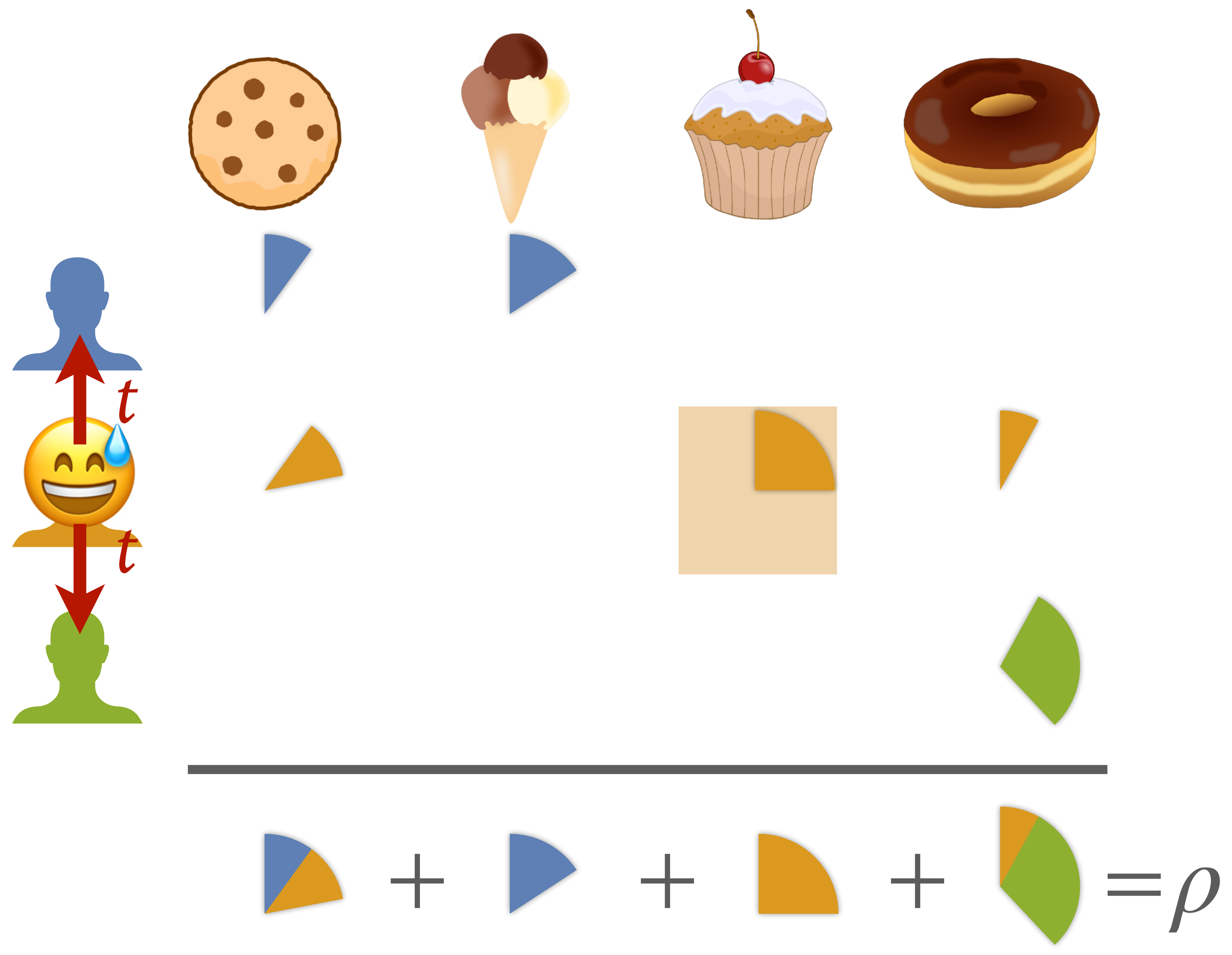
Algorithm for Finding EF Allocations w.h.p.

for $\rho = 1, 2, 4, \ldots, m$:

- with LP, allocate ρ fractional items such that $t = \min\limits_{\text{agents } i \neq j} u_i(A_i) - u_i(A_j)$ is maximized.

  i's utility for own bundle   *i*'s utility for *j*'s bundle

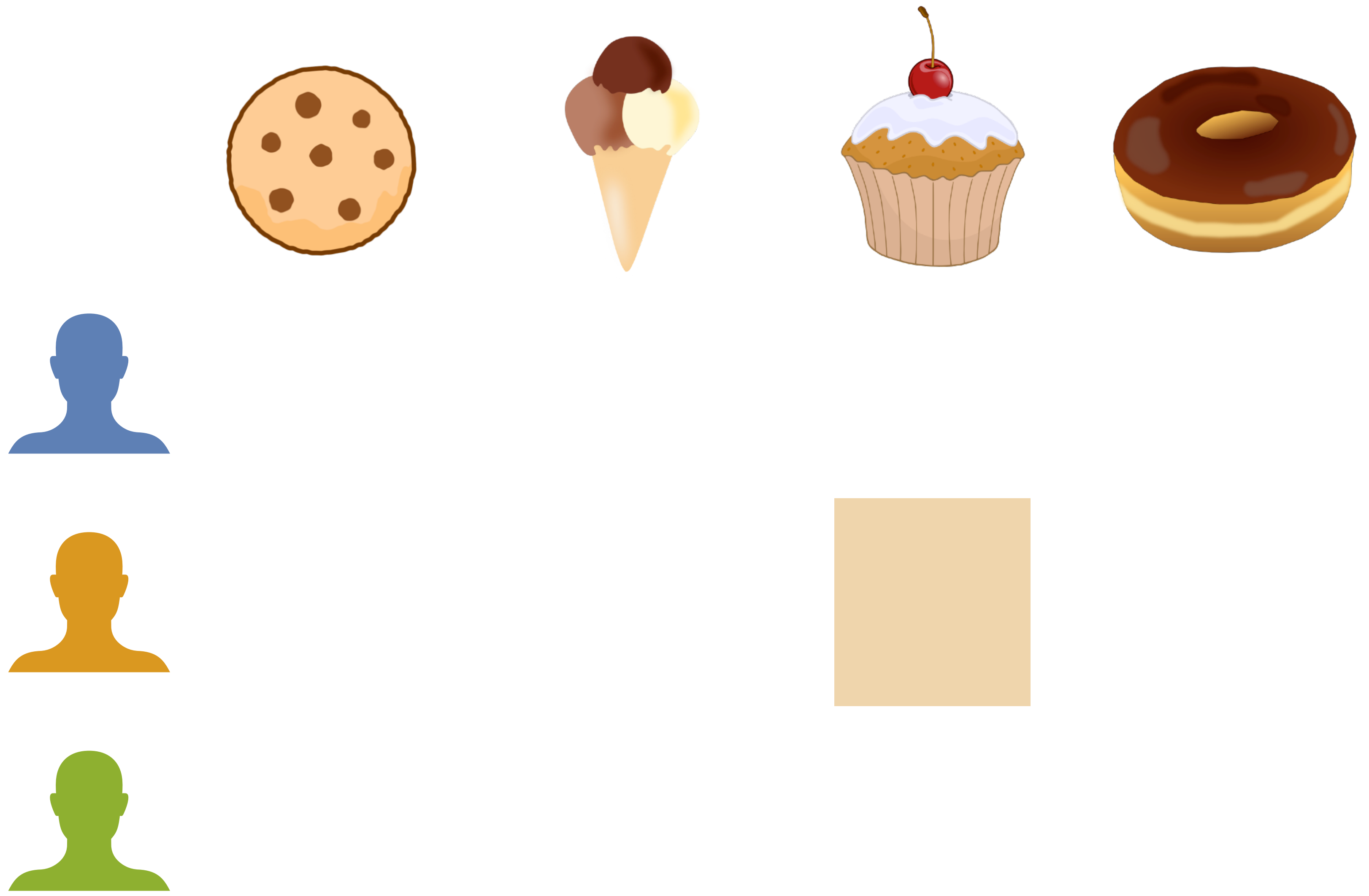
- randomly round this fractional allocation.



Algorithm for Finding EF Allocations w.h.p.

for $\rho = 1, 2, 4, \ldots, m$:

- with LP, allocate ρ fractional items such that
  $$t = \min_{\text{agents } i \neq j} u_i(A_i) - u_i(A_j) \text{ is maximized.}$$
  i's utility for own bundle *i*'s utility for *j*'s bundle
- randomly round this fractional allocation.



Algorithm for Finding EF Allocations w.h.p.

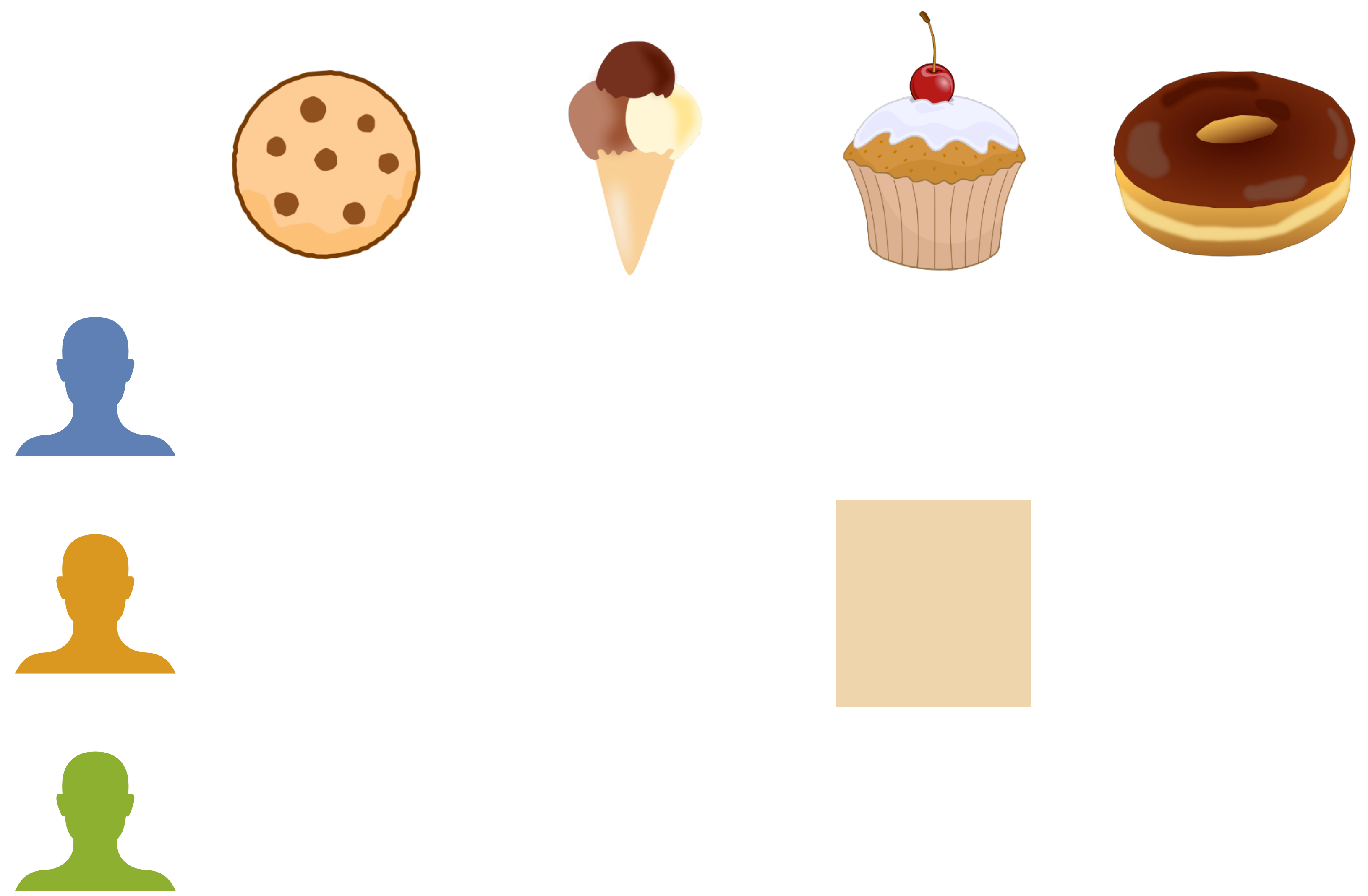
for $\rho = 1, 2, 4, \ldots, m$:

- with LP, allocate ρ fractional items such that $t = \min_{\text{agents } i \neq j} u_i(A_i) - u_i(A_j)$ is maximized.

  i's utility for own bundle   *i*'s utility for *j*'s bundle

- randomly round this fractional allocation.

- allocate remaining items using round robin.



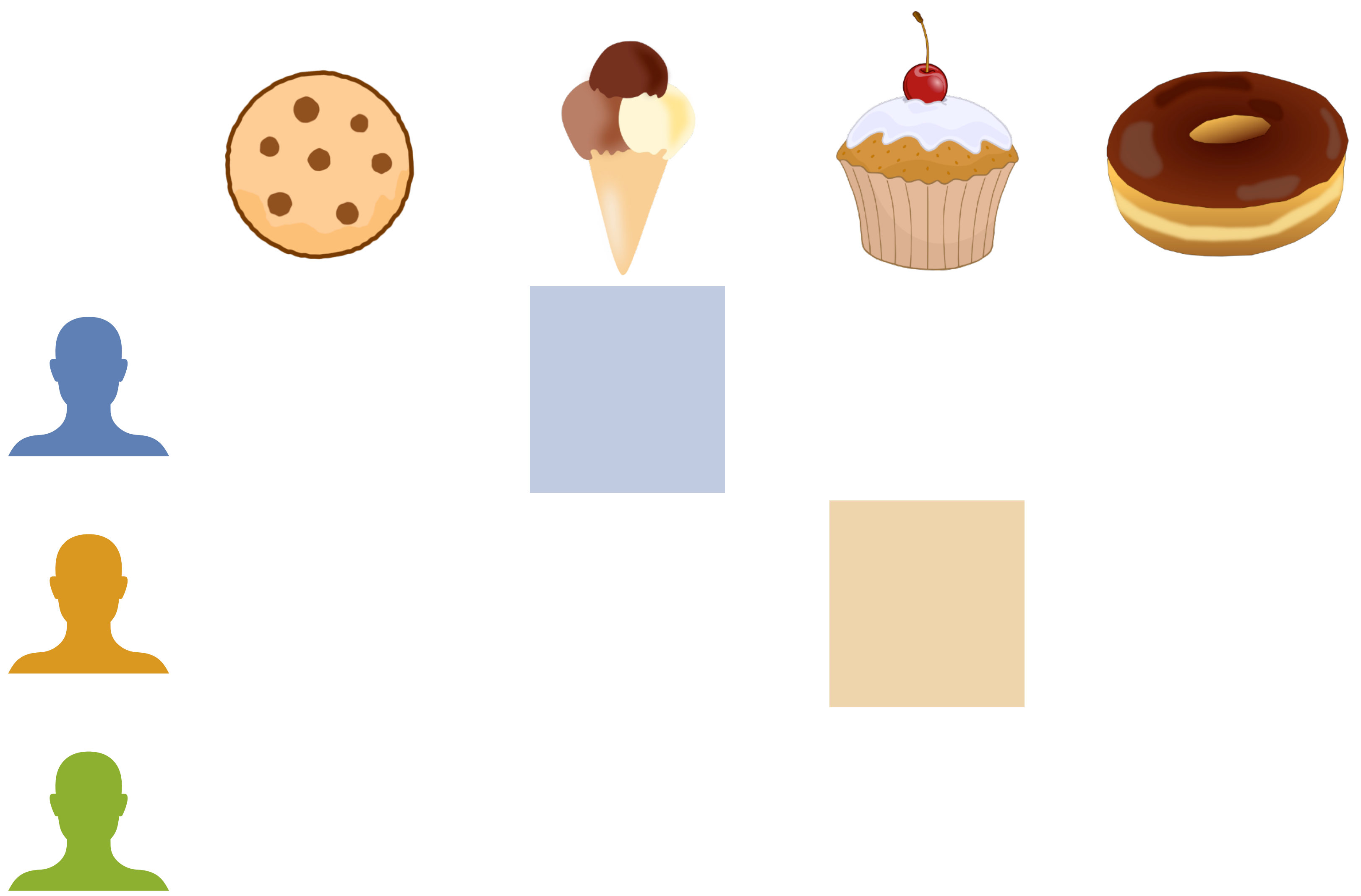
Algorithm for Finding EF Allocations w.h.p.

for $\rho = 1, 2, 4, \ldots, m$:

- with LP, allocate ρ fractional items such that $t = \min_{\text{agents } i \neq j} u_i(A_i) - u_i(A_j)$ is maximized.

  i's utility for own bundle   *i*'s utility for *j*'s bundle

- randomly round this fractional allocation.
- allocate remaining items using round robin.



Algorithm for Finding EF Allocations w.h.p.

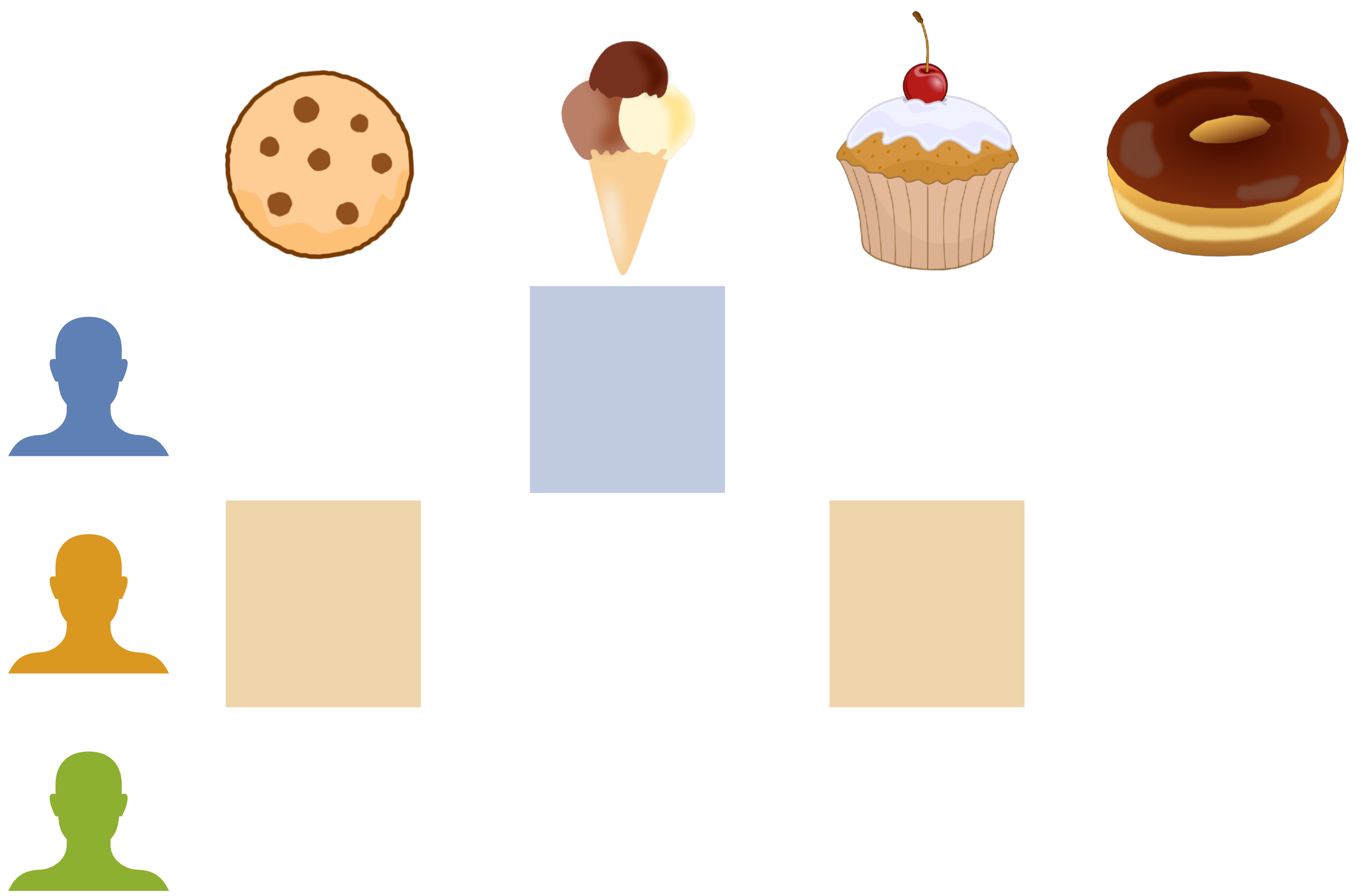
for $\rho = 1, 2, 4, \ldots, m$:

- with LP, allocate ρ fractional items such that $t = \min_{\text{agents } i \neq j} u_i(A_i) - u_i(A_j)$ is maximized.

  i's utility for own bundle   *i*'s utility for *j*'s bundle

- randomly round this fractional allocation.

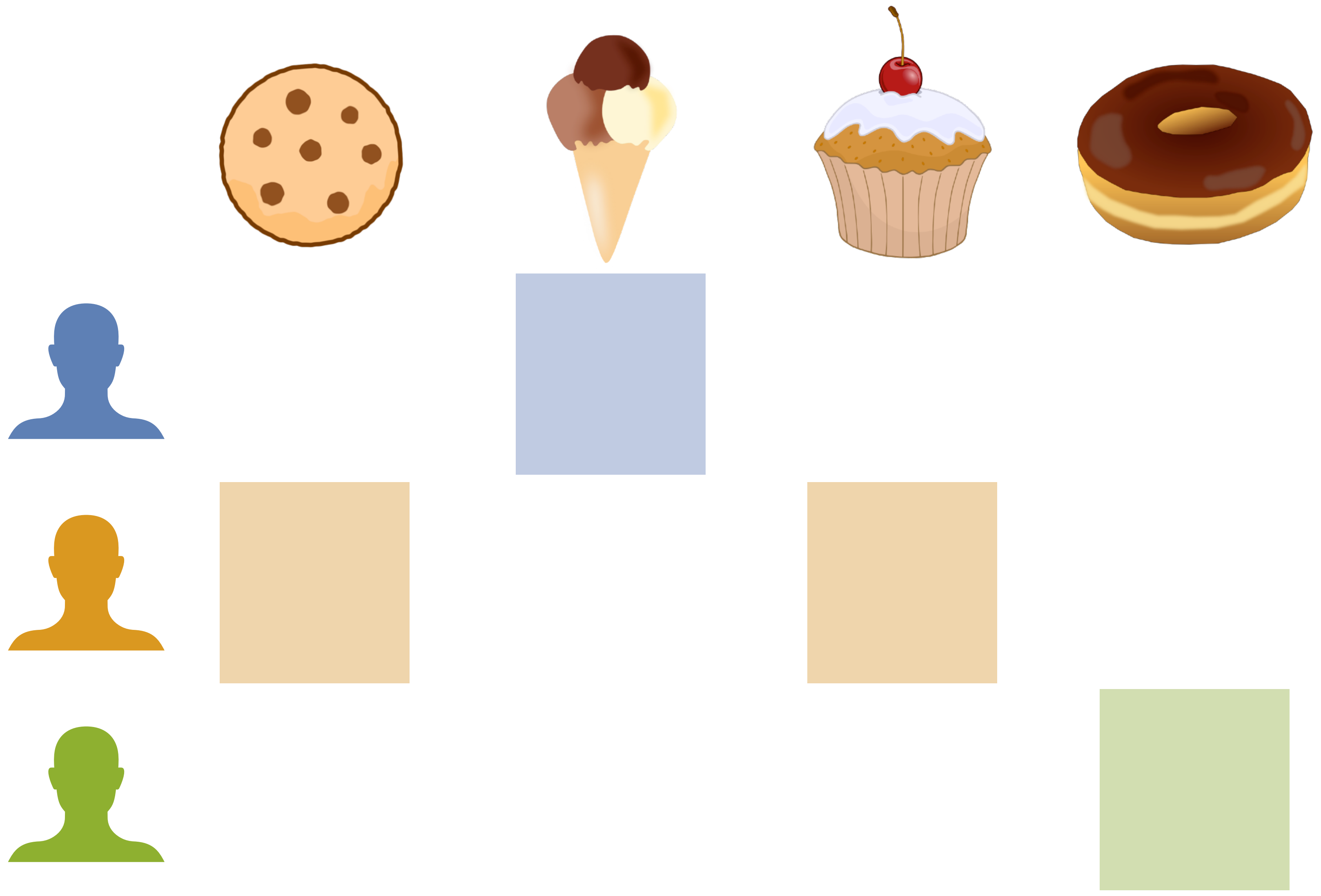
- allocate remaining items using round robin.



Algorithm for Finding EF Allocations w.h.p.

for $\rho = 1, 2, 4, \ldots, m$:

- with LP, allocate ρ fractional items such that
  $$t = \min_{\text{agents } i \neq j} u_i(A_i) - u_i(A_j) \text{ is maximized.}$$
  i's utility for own bundle *i*'s utility for *j*'s bundle
- randomly round this fractional allocation.
- allocate remaining items using round robin.



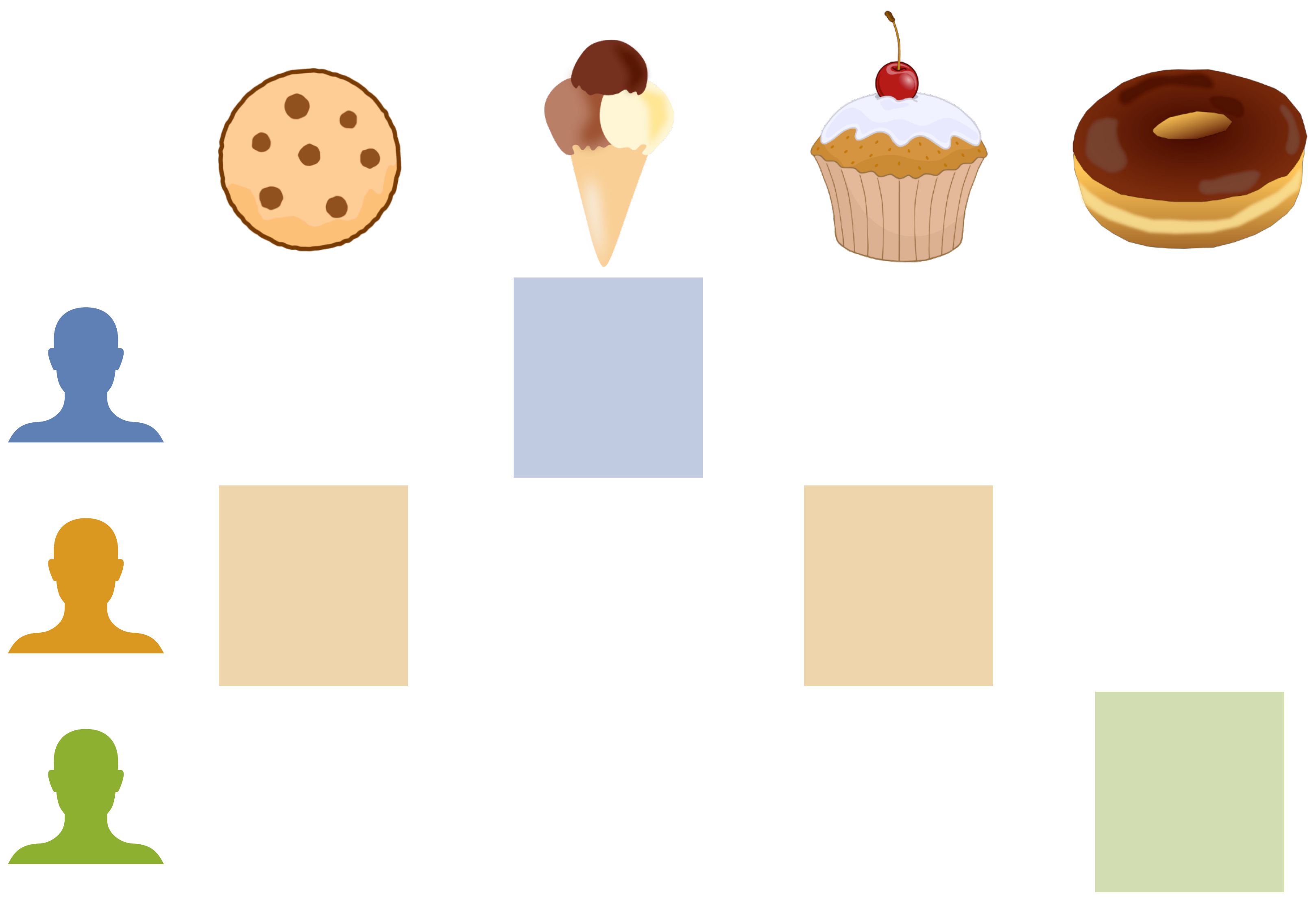
Algorithm for Finding EF Allocations w.h.p.

for $\rho = 1, 2, 4, \ldots, m$:

- with LP, allocate ρ fractional items such that
  $t = \min_{\text{agents } i \neq j} u_i(A_i) - u_i(A_j)$ is maximized.

  i's utility for own bundle　　*i*'s utility for *j*'s bundle

- randomly round this fractional allocation.
- allocate remaining items using round robin.
- if EF, return, else continue loop.



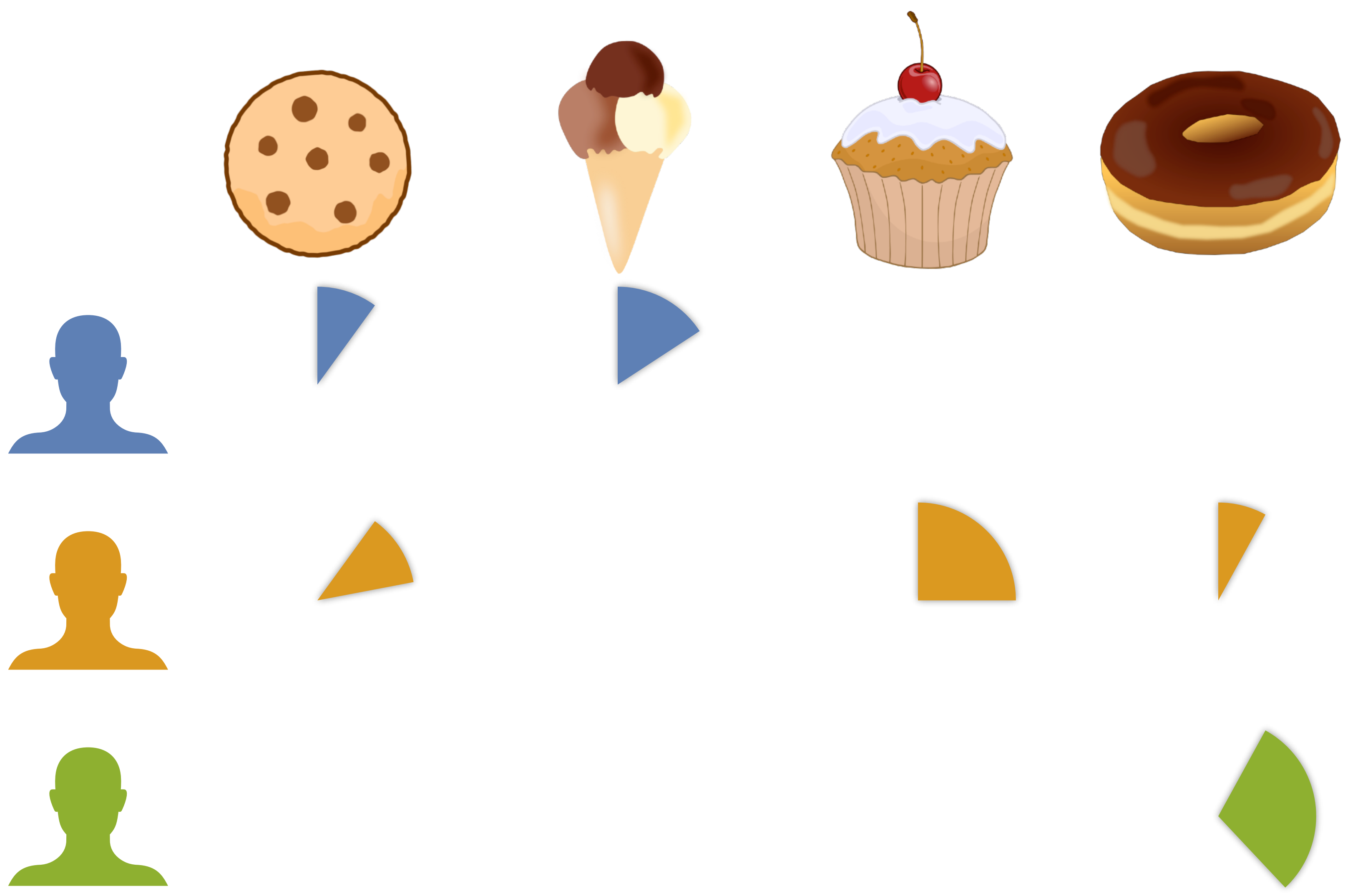
Algorithm for Finding EF Allocations w.h.p.

for $\rho = 1, 2, 4, \ldots, m$:

- with LP, allocate ρ fractional items such that $t = \min_{\text{agents } i \neq j} u_i(A_i) - u_i(A_j)$ is maximized.

  i's utility for own bundle　　*i*'s utility for *j*'s bundle

- randomly round this fractional allocation.
- allocate remaining items using round robin.
- if EF, return, else continue loop.



Overview of Results

Theorem. If $m \gg n \log n$ and $p \gg \log n / m$, our poly-time algorithm finds envy-free allocations in the smoothed model with probability $\to 1$.

n: number of agents
m: number of items
p: boost probability

Overview of Results

Theorem. If $m \gg n \log n$ and $p \gg \log n/m$, our poly-time algorithm finds envy-free allocations in the smoothed model with probability $\to 1$.

n: number of agents
m: number of items
p: boost probability

+ bound on p is tight

Overview of Results

Theorem. If $m \gg n \log n$ and $p \gg \log n / m$, our poly-time algorithm finds envy-free allocations in the smoothed model with probability $\to 1$.

n: number of agents
m: number of items
p: boost probability

+ bound on p is tight
+ even at $p = m^{0.33}/m$, round robin doesn't guarantee envy-freeness

Overview of Results

Theorem. If $m \gg n \log n$ and $p \gg \log n / m$, our poly-time algorithm finds envy-free allocations in the smoothed model with probability $\to 1$.

n: number of agents
m: number of items
p: boost probability

+ bound on p is tight

+ even at $p = m^{0.33}/m$, round robin doesn't guarantee envy-freeness

Theorem. If $m \gg n \log n$ and $p \gg \log n / m$, another algorithm finds Pareto-optimal and proportional allocations with probability $\to 1$.

Overview of Results

Theorem. If $m \gg n \log n$ and $p \gg \log n/m$, our poly-time algorithm finds envy-free allocations in the smoothed model with probability $\to 1$.

n: number of agents
m: number of items
p: boost probability

+ bound on p is tight
+ even at $p = m^{0.33}/m$, round robin doesn't guarantee envy-freeness

Theorem. If $m \gg n \log n$ and $p \gg \log n/m$, another algorithm finds Pareto-optimal and proportional allocations with probability $\to 1$.

Check out our paper!
tinyurl.com/smoothEF