Resolving Envy by Adding Goods with Bounded Supply: Two-Agent Hardness and Single-Type Tractability

Speaker: Chuang-Chieh Lin (Joseph)

NTOU, TW

a joint work with

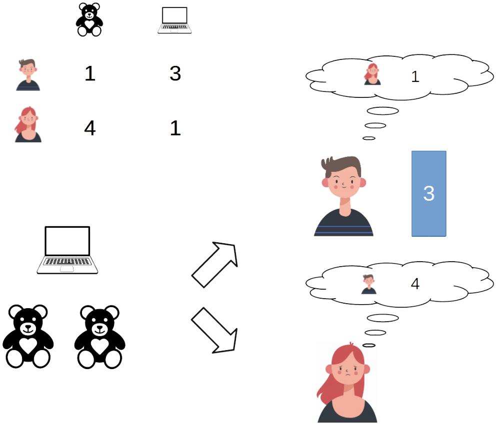
Fertin Guillaume, Géraldine Jean, Emile Benoist
NU, FR

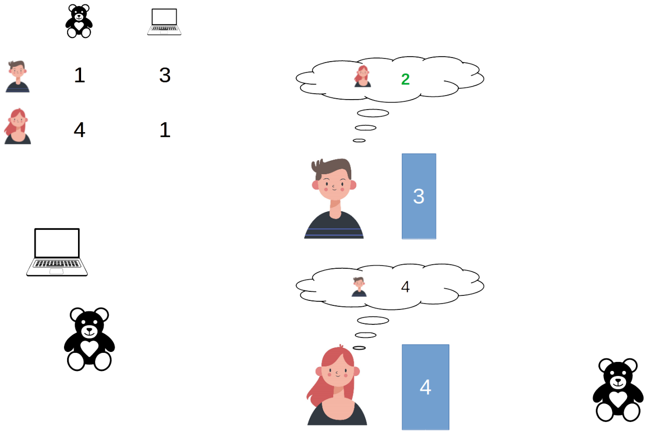
Po-An Chen
NYCU, TW

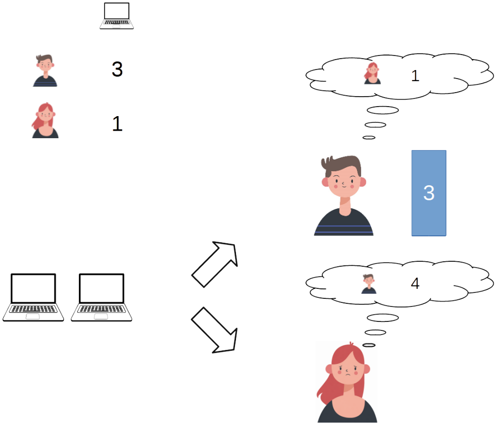
Stéphane Vialette
UGE, FR

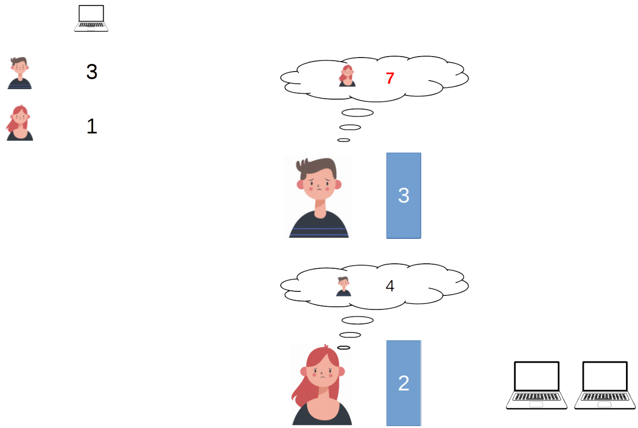
Colin Cleveland
KCL, UK

CFD Workshop, IJCAI-ECAI 2026


1
3
4
1
1
3
4


1
3
4
1
2
3
4
4


3
1
1
3
4


3
1
7
3
4
2

Outline

EEAG repairs a fixed allocation

Initial allocation

fixed; cannot be reallocated

EEAG repairs a fixed allocation

Initial allocation → Additional goods

fixed; cannot be reallocated

bounded supplies; optional leftovers

EEAG repairs a fixed allocation

Initial allocation → Additional goods → Envy-free extension?

fixed; cannot be reallocated

bounded supplies; optional leftovers

no agent prefers another's final bundle

EEAG repairs a fixed allocation

Initial allocation → Additional goods → Envy-free extension?

fixed; cannot be reallocated

bounded supplies; optional leftovers

no agent prefers another's final bundle

Envy Elimination by Adding Goods (EEAG)

- Start from a fixed allocation σ and assign copies of additional item types.
- Decide an extension $\rho(a, r)$ which records how many copies of type r agent a receives.
- Respect every finite supply and make the extended allocation envy-free.

Original goods stay fixed; unused additional goods are allowed.
No separate budget is imposed.

Our Contributions: Two complementary endpoints

We sharpen the bounded-supply landscape of Bentert et al. (IJCAI'25).

Many unit-supply types

Weakly NP-complete, even with

- two agents with identical additive valuations,
- one initially endowed good, and
- no explicit budget bound.

One bounded item type

Polynomial time for

- one bounded item type, any number of agents, and
- arbitrary nonnegative integer per-copy values.

Difference constraints yield a componentwise least extension.

The Hardness Result

ESSE: Equal Sum Subsets with an Enforced Element

Restricted EEAG instance

Two identical agents; agent 1 owns one good of value h. Additional goods have unit supplies and values $w_1, w_2, \ldots, w_\ell$. For disjoint assigned sets I_1, I_2, envy-freeness is exactly

$$h + \sum_{t \in I_1} w_t = \sum_{t \in I_2} w_t.$$

The remaining items may be unused.

Reduction from ESSE

[Cieliebak et al. 2008]

Given m positive integers $q_1, q_2, \ldots, q_m$, ESSE asks for disjoint $X, Y \subseteq [m]$ with

$$m \in X, \quad \sum_{t \in X} q_t = \sum_{t \in Y} q_t.$$

Set $h := q_m$ and $w_t := q_t$ for $t < m$; then

$$I_1 = X \setminus \{m\}, \quad I_2 = Y,$$

in both directions.

Theorem

The restricted EEAG is weakly NP-complete.

The Hardness Result

ESSE: Equal Sum Subsets with an Enforced Element

Restricted EEAG instance

Two identical agents; agent 1 owns one good of value h. Additional goods have unit supplies and values $w_1, w_2, \ldots, w_\ell$. For disjoint assigned sets I_1, I_2, envy-freeness is exactly

$$h + \sum_{t \in I_1} w_t = \sum_{t \in I_2} w_t.$$

The remaining items may be unused.

Reduction from ESSE

[Cieliebak et al. 2008]

Given m positive integers $q_1, q_2, \ldots, q_m$, ESSE asks for disjoint $X, Y \subseteq [m]$ with

$$m \in X, \quad \sum_{t \in X} q_t = \sum_{t \in Y} q_t.$$

Set $h := q_m$ and $w_t := q_t$ for $t < m$; then

$$I_1 = X \setminus \{m\}, \quad I_2 = Y,$$

in both directions.

Theorem

The restricted EEAG is weakly NP-complete.

Note: A direct reduction from PARTITION is insufficient.

One item type turns envy into difference constraints

There are K copies of one type g; agent i values each copy at $v_i \in \mathbb{Z}_{\geq 0}$ and receives x_i copies. Let the initial bundles be

$$U_i^{\mathrm{self}} := u_i(B_i), \quad U_i^{(j)} := u_i(B_j), \quad \Delta_{ij} := U_i^{(j)} - U_i^{\mathrm{self}}.$$

For every ordered pair (i, j), the envy constraint becomes

$$v_i(x_i - x_j) \geq \Delta_{ij}.$$

$v_i = 0$**.** If $\Delta_{ij} > 0$, no added copy can help agent i: reject.

$v_i > 0$**.** The constraint becomes

$$x_i - x_j \geq c_{ij}, \quad c_{ij} := \left\lceil \frac{\Delta_{ij}}{v_i} \right\rceil.$$

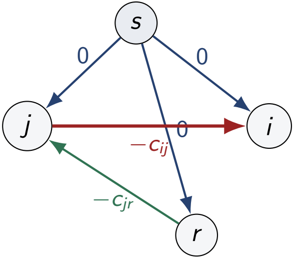
Shortest paths construct a least extension

Set $y_i := -x_i$. Each lower-bound constraint becomes

$$x_i - x_j \geq c_{ij} \iff y_i \leq y_j - c_{ij}.$$

Construct a directed graph H:

- vertices: $\{s, 1, 2, \ldots, n\}$;
- for each i, add (s, i) with weight 0;
- for each constraint, add (j, i) with weight $-c_{ij}$.



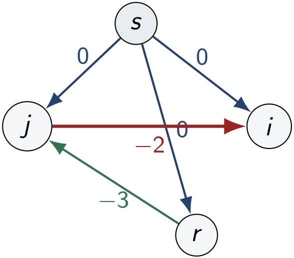
Shortest paths construct a least extension

Set $y_i := -x_i$. Each lower-bound constraint becomes

$$x_i - x_j \geq c_{ij} \iff y_i \leq y_j - c_{ij}.$$

Construct a directed graph H:

- vertices: $\{s, 1, 2, \ldots, n\}$;
- for each i, add (s, i) with weight 0;
- for each constraint, add (j, i) with weight $-c_{ij}$.



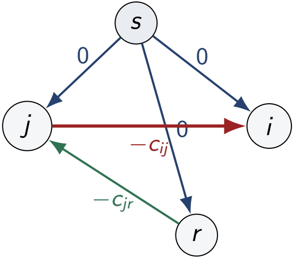
Shortest paths construct a least extension

Set $y_i := -x_i$. Each lower-bound constraint becomes

$$x_i - x_j \geq c_{ij} \iff y_i \leq y_j - c_{ij}.$$

Construct a directed graph H:

- vertices: $\{s, 1, 2, \ldots, n\}$;
- for each i, add (s, i) with weight 0;
- for each constraint, add (j, i) with weight $-c_{ij}$.



Bellman–Ford outcome

A negative-weight cycle means infeasibility. Otherwise, let d_i be the shortest-path distance from s to i and set $x_i^* := -d_i$. The running time is $O(n^3)$ arithmetic operations.

Why the supply check is correct

1. The distance vector is feasible

The direct edge (s, i) has weight 0, so $d_i \leq 0$ and $x_i^* = -d_i \geq 0$. Moreover, $d_i \leq d_j - c_{ij}$ implies

$$x_i^* - x_j^* = -d_i + d_j \geq c_{ij}.$$

2. It is componentwise least

Let $\mathbf{x}$ be any feasible vector and put $\mathbf{y} = -\mathbf{x}$. Chaining the edge constraints along any $s \to i$ path gives $y_i \leq$ its path weight. Hence $y_i \leq d_i$, and therefore

$$x_i = -y_i \geq -d_i = x_i^* \qquad \text{for every } i.$$

Thus every feasible vector uses at least $\sum_i x_i^*$ copies:

$$\text{an extension within supply exists} \iff \sum_i x_i^* \leq K.$$

Concluding Remarks

Two agents, many unit-supply types

Weakly NP-complete with two identical agents, one initially endowed good, and no explicit budget.

One bounded type

Polynomial time for any number of agents and arbitrary $v_i \in \mathbb{Z}_{\geq 0}$, via shortest paths.

Future Work

- What is the complexity for **exactly two** additional item types?
- Which broader valuation classes preserve a polynomial tractability?

Concluding Remarks

Two agents, many unit-supply types

Weakly NP-complete with two identical agents, one initially endowed good, and no explicit budget.

One bounded type

Polynomial time for any number of agents and arbitrary $v_i \in \mathbb{Z}_{\geq 0}$, via shortest paths.

Future Work

- What is the complexity for **exactly two** additional item types?
  - **NP-Completeness** [arXiv:2608.10326]
- Which broader valuation classes preserve a polynomial tractability?

Thanks for your attention!

Q & A

Backup Slides

Equal Sum Subsets with an Enforced Element (ESSE)

Definition (ESSE)

Given positive integers $q_1, q_2, \dots, q_m$, with q_m distinguished, decide whether there are disjoint index sets $X, Y \subseteq [m]$ such that

$$\sum_{t \in X} q_t = \sum_{t \in Y} q_t \quad \textit{and} \quad m \in X.$$

Known fact [Cieliebak et al., 2008]

ESSE is NP-complete and admits an $O(m^2 Q)$ pseudo-polynomial algorithm, where $Q = \sum_t q_t$; hence it is weakly NP-complete.

Why a direct Partition reduction fails

In PARTITION, every input item must be placed on one side. In our EEAG restriction, **added items may remain unused**, so a proposed correspondence can create false YES-instances.

Counterexample

The PARTITION instance $A = \{1, 1, 4\}$ is NO because no subset sums to $\frac{1}{2}\sum_{a \in A} a = 3$. But setting $h = 3$ gives a YES EEAG instance:

$$I_1 = \{1\}, \qquad I_2 = \{4\}, \qquad 3 + 1 = 4,$$

with one item of value 1 unused.

Binary warm-up as an LP

When $v_i \in \{0, 1\}$:

- if $v_i = 0$ and $\Delta_{ij} > 0$, infeasible;
- if $v_i = 1$, the constraint is $x_i - x_j \geq \Delta_{ij}$.

A minimum-sum feasible vector can be obtained from

$$\begin{array}{ll} \min & \sum_i x_i \\ \text{s.t.} & x_i - x_j \geq \Delta_{ij} \quad \text{for all } (i,j) \text{ with } v_i = 1, \\ & x_i \geq 0 \quad \text{for all } i. \end{array}$$

The constraint matrix has rows of the form $e_i - e_j$ and e_i, hence is totally unimodular.

Representing the constraint matrix

- Let A be the matrix whose rows are of the forms $e_i - e_j$ and e_i, where e_i is the i-th standard unit vector.

Auxiliary directed graph

Construct a directed multigraph H with vertex set $V(H) = \{0, 1, \ldots, n\}$, For every row of A:

- a row $e_i - e_j$ is represented by an arc $j \to i$;
- a row e_i is represented by an arc $0 \to i$.

- For every arc $u \to v$, the corresponding incidence row is $e_v - e_u$. That is, it has -1 in column u, $+1$ in column v, and zero in every other column.

Backup: obtaining A from the incidence matrix

Let $\widetilde{A}$ be the edge–vertex incidence matrix of H:

	0	1	$\cdots$	i	$\cdots$	n
$j \to i$	0	$\cdots$	-1 at j	$+1$	$\cdots$	0
$0 \to i$	-1	0	$\cdots$	$+1$	$\cdots$	0

Delete the column corresponding to the auxiliary vertex 0.

- The row for $j \to i$ remains: $e_i - e_j$.
- The row for $0 \to i$ becomes: e_i.

Therefore,

$$A = \widetilde{A} \text{ with the auxiliary column 0 deleted.}$$

It remains to prove that $\widetilde{A}$ is totally unimodular.

Why the incidence matrix is totally unimodular

- Prove that every square submatrix C of $\widetilde{A}$ has $\det(C) \in \{-1, 0, 1\}$ by induction on its order k.

Case 1: some row has at most one nonzero entry

- If the row is zero, then $\det(C) = 0$.

  Otherwise, its unique nonzero entry is $+1$ or -1.
- Cofactor expansion along this row gives $\det(C) = \pm\det(C')$, where C' is a $(k-1) \times (k-1)$ submatrix of $\widetilde{A}$.
- By induction, $\det(C') \in \{-1, 0, 1\}$.

Case 2: every row has at least two nonzero entries

Then, every row of C contains exactly one $+1$ and one -1.
Therefore every row sums to zero $\Rightarrow$ C is singular and $\det(C) = 0$.

Hence $\widetilde{A}$ is totally unimodular.

Bellman–Ford proof details

For each constraint $x_i - x_j \geq c_{ij}$, use $y_i = -x_i$:

$$y_i \leq y_j - c_{ij}.$$

This gives edge $j \to i$ of weight $-c_{ij}$.

- Negative cycle: infeasible system (i.e., $y_i \leq y_i + C$ for some $C \leq 0$).
- No negative cycle: Bellman–Ford gives shortest distances d_i.
- Set $x_i^* = -d_i$.
- $\mathbf{x}^* := (x_i)_{i \in [n]}$ is feasible and componentwise minimum.
- Check $\sum_i x_i^* \leq K$.

Backup: a natural greedy heuristic can fail

Two agents, $K = 2$ copies, $v_1 = 1$, $v_2 = 10$.

$$U_1^{\mathrm{self}} = 5, \quad U_1^{(2)} = 0, \qquad U_2^{\mathrm{self}} = 8, \quad U_2^{(1)} = 20.$$

Greedy gives both items to agent 1 because she remains the lowest-utility agent, producing $(x_1, x_2) = (2, 0)$. But agent 2 still envies agent 1:

$$8 + 10 \cdot 0 < 20 + 10 \cdot 2.$$

By contrast, $(x_1, x_2) = (0, 2)$ is envy-resolving: agent 1 compares 5 with $0 + 1 \cdot 2 = 2$, while agent 2 compares $8 + 10 \cdot 2 = 28$ with 20.