On Fair Division of Indivisible Goods

joint work with Yun Kuen (Marco) Cheung, Bhaskar Chaudhury, Jugal Garg, Naveen Garg, Martin Hoefer

arXiv 2018

Kurt Mehlhorn

max planck institut informatik

SIC Saarland Informatics Campus

Outline

- Fair Division Problems
- Problem Definition: Allocation of Indivisible Items
- State of the Art
- Divisible Items
- An Approximation Algorithm for Indivisible Items via Envy-Freeness by Barman et al.
- Our Generalization
- Open Problems

Fair Division Problems

- Share rent.
- Assign credit to the authors of a paper.
- Distribute tasks, e.g., household chores.
- Split goods among kids at Xmas.
- Split an estate among heirs.

spliddit

Allocation of Items to Agents

- Set G of m indivisible items or goods
- Set A of n agents or users
- u_{ij} = value (utility) of good j for agent i
- Each item assigned to some agent.
- x_i = set of items assigned to agent i.
- Value (utility) of x_i for agent i: $u_i(x_i) = \sum_{j \in x_i} u_{ij}$

spliddit

Main Questions

- What is a good allocation ?
- Algorithms to find (approximately) optimal allocations?
- Computational complexity of finding good allocations?

What is a Good Allocation? Objectives

- Utilitarian Social Welfare

$$\text{maximize} \sum_{i \in A} u_i(x_i)$$

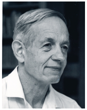
- Max-Min-Fairness, Egalitarian Welfare

$$\text{maximize} \min_{i \in A} u_i(x_i)$$

- Proportional Fairness, Nash Social Welfare (NSW)

$$\text{maximize} \left(\prod_{i \in A} u_i(x_i) \right)^{1/n}$$

- NSW is invariant under scaling.



Algorithms for Approximating Nash Social Welfare

ALG computes a ρ-approximation if for every instance *I*

$$\frac{\text{NSW}(x^*)}{\text{NSW}(ALG(I))} \leq \rho.$$

- APX-hard, no 1.00008-approximation unless P = NP [Lee, IPL'17]
- 2.889-approximation via markets [Cole, Gkatzelis, STOC'15]
- *e*-approximation via stable polynomials [Anari, Gharan, Singh, Saberi, ITCS'17]
- 2-approximation via markets [Cole, Devanur, Gkatzelis, Jain, Mai, Vazirani, Yazdanbod, EC'17]
- 1.45-approximation via limited envy [Barman, Krishnamurthy, Vaish, EC 2018] [Caragiannis, Kurokawa, Moulin, Procaccia, Shah, Wang, EC'16]
- Algorithm by Barman et al. is the simplest to state and to analyse. Took me four hours to implement.

Algorithms for Approximating Nash Social Welfare

ALG computes a ρ-approximation if for every instance *I*

$$\frac{\text{NSW}(x^*)}{\text{NSW}(ALG(I))} \leq \rho.$$

- APX-hard, no 1.00008-approximation unless $P = NP$ [Lee, IPL'17]
- 2.889-approximation via markets [Cole, Gkatzelis, STOC'15]
- *e*-approximation via stable polynomials [Anari, Gharan, Singh, Saberi, ITCS'17]
- 2-approximation via markets [Cole, Devanur, Gkatzelis, Jain, Mai, Vazirani, Yazdanbod, EC'17]
- 1.45-approximation via limited envy [Barman, Krishnamurthy, Vaish, EC 2018] [Caragiannis, Kurokawa, Moulin, Procaccia, Shah, Wang, EC'16]
- Algorithm by Barman et al. is the simplest to state and to analyse. Took me four hours to implement.

Extensions

- Multiple copies of each item [Bei, Garg, Hoefer, Mehlhorn, SAGT'17]
- Multiple copies, diminishing value [Anari, Mai, Oveis Gharan, Vazirani, SODA 2018]
- Budget-additive utilities, $u_i(x_i) = \min\left(c_i, \sum_{j \in x_i} u_{ij}\right)$ [Garg, Hoefer, M, SODA 2018]
- Multiple copies, diminishing value, budget-additive [Cheung, Chaudhury, Garg, Garg, Hoefer, M., ArXiv 2018]

The latter instance class contains the classes above and the algorithm achieves a better approximation ratio.

The ratio is 1.45, the same as in [Barman, Krishnamurthy, Vaish, EC 2018]

Algorithm combines ideas from Barman et al. and Anari et al. Retains simplicity.

Divisible Goods

$x_{ij} \in [0,1]$: fraction of good j assigned to agent i.

Problem reduces to a Fisher market

- Give every agent the same budget, say 1 Euro
- Find prices p_j for the goods such that the market clears, i.e.,
  - all goods are completely sold, i.e., $\sum_i x_{ij} = 1$ for all j.
  - agents spend all their money, i.e., $\sum_j p_j x_{ij} = 1$.
  - agents behave rationally, i.e., $x_{ij} > 0 \quad \Rightarrow \frac{u_{ij}}{p_j} = \alpha_i = \max_\ell \frac{u_{i\ell}}{p_\ell}$
  - α_i is called the bang-per-buck (MBB) ratio of agent i.

The Algorithm by Barman et al.

computes

- an allocation x; $x_i =$ set of goods assigned to agent i.
- a price vector p; $p_j =$ price of good j.
- a vector α; $\alpha_i =$ MBB-ratio of agent i.

such that

- $\alpha_i = \max_j {}^{u_{ij}}/_{p_j}$ (α_i is maximum-bang-per-buck ratio of i)
- $j \in x_i$ implies $u_{ij}/p_j = \alpha_i$ (only MBB-goods are allocated)
- for all agents h and i, there is a good j such that

$$p(x_h \setminus j) \leq (1 + \varepsilon)p(x_i),$$

  where $p(\text{set } S \text{ of goods}) = \sum_{j \in S} p_j$. (budget equal up to one good)

The first two properties are maintained throughout the algorithm. We work towards the third.

The Algorithm by Barman et al.

computes

- an allocation x; x_i = set of goods assigned to agent i.
- a price vector p; p_j = price of good j.
- a vector α; α_i = MBB-ratio of agent i.

such that

- $\alpha_i = \max_j u_{ij}/p_j$ (α_i is maximum-bang-per-buck ratio of i)
- $j \in x_i$ implies $u_{ij}/p_j = \alpha_i$ (only MBB-goods are allocated)
- for all agents h and i, there is a good j such that

$$p(x_h \setminus j) \leq (1+\varepsilon)p(x_i),$$

  where $p(\text{set } S \text{ of goods}) = \sum_{j \in S} p_j$. (budget equal up to one good)

Note $u_i(x_h \setminus j) \leq \alpha_i \cdot p(x_h \setminus j) \leq (1+\varepsilon)\alpha_i \cdot p(x_i) = (1+\varepsilon)u_i(x_i)$.

The Algorithm by Barman et al.

Initialization: assign every item to the agent that likes it most.

for *good j* **do**
  assign j to $i_0 = \text{argmax}_i\, u_{ij}$, set $p_j \leftarrow u_{i_0,j}$

for *agent i* **do**
  $\alpha_i = 1$

Main Loop: as long as there is envy, reassign goods and adjust prices.

A pair (i, j) of good and agent is tight if $\alpha_i = u_{ij}/p_j$.

Tight Graph: directed bipartite graph, agents on one side, goods on the other side.

- edge (i, j) from agent i to good j: tight and i does not own j.
- edge (j, i) from good j to agent i: tight and i owns j.

The Algorithm by Barman et al.

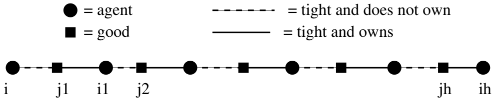
Initialization
while *true* **do**
- let i be a least spending agent ($p(x_i)$ is minimum)
- **if** *i does not envy any other agent* **then**
  - **break** from the loop and halt
- do a BFS in tight graph starting at i;
- **if** *BFS finds an envy-reducing path starting in i* **then**
  - use the shortest such path to improve the assignment
- **else**
  - Let S be the set of agents that can be reached from i in tight graph
  - multiply all prices of goods owned by S and divide all MBB-values of agents in S by an increasing factor $t > 1$ until
    - (a) a new tight edge from an agent in S to a good outside S
    - (b) i is not envious anymore
    - (c) new least spender

Envy-Reducing Path

Invariant: $\alpha_i \geq u_{ij}/p_j$ for all j and $\alpha_i = u_{ij}/p_j$ if $j \in x_i$.

A pair (i, j) of agent and good is tight if $\alpha_i = u_{ij}/p_j$.

Tight path:



Tight path is envy-reducing if

$p(i_h \setminus j_{h-1}) > (1+\varepsilon)p(x_i)$ and $p(i_\ell \setminus j_{\ell-1}) \leq (1+\varepsilon)p(x_i)$ for $\ell < h$.

Use of envy-reducing $P = (i = i_0, j_1, i_1, \ldots, j_h, i_h)$:

Set $\ell \leftarrow h$

while $\ell > 0$ *and* $p_{i_\ell}(x_{i_\ell} \setminus j_\ell) > (1+\varepsilon)p_i(x_i)$ **do**

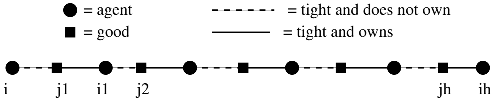
remove j_ℓ from x_{i_ℓ} and assign it to $i_{\ell-1}$; $\ell \leftarrow \ell - 1$

Envy-Reducing Path

Invariant: $\alpha_i \geq u_{ij}/p_j$ for all j and $\alpha_i = u_{ij}/p_j$ if $j \in x_i$.

A pair (i, j) of agent and good is tight if $\alpha_i = u_{ij}/p_j$.

Tight path:



Tight path is envy-reducing if

$$p(i_h \setminus j_{h-1}) > (1+\varepsilon)p(x_i) \text{ and } p(i_\ell \setminus j_{\ell-1}) \leq (1+\varepsilon)p(x_i) \text{ for } \ell < h.$$

Use of envy-reducing $P = (i = i_0, j_1, i_1, \ldots, j_h, i_h)$:

Set $\ell \leftarrow h$

while $\ell > 0$ *and* $p_{i_\ell}(x_{i_\ell} \setminus j_\ell) > (1+\varepsilon)p_i(x_i)$ **do**

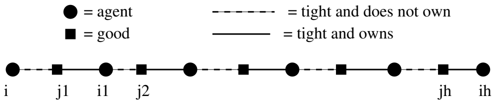
remove j_ℓ from x_{i_ℓ} and assign it to $i_{\ell-1}$; $\ell \leftarrow \ell - 1$

Envy-Reducing Path

Invariant: $\alpha_i \geq u_{ij}/p_j$ for all j and $\alpha_i = u_{ij}/p_j$ if $j \in x_i$.

A pair (i, j) of agent and good is tight if $\alpha_i = u_{ij}/p_j$.

Tight path:



Tight path is envy-reducing if

$$p(i_h \setminus j_{h-1}) > (1+\varepsilon)p(x_i) \text{ and } p(i_\ell \setminus j_{\ell-1}) \leq (1+\varepsilon)p(x_i) \text{ for } \ell < h.$$

Use of envy-reducing $P = (i = i_0, j_1, i_1, \ldots, j_h, i_h)$:

Set $\ell \leftarrow h$

while $\ell > 0$ *and* $p_{i_\ell}(x_{i_\ell} \setminus j_\ell) > (1+\varepsilon)p_i(x_i)$ **do**

remove j_ℓ from x_{i_ℓ} and assign it to $i_{\ell-1}$; $\ell \leftarrow \ell - 1$

Polynomial Time

Assume all utilities are powers of $r = 1 + \delta$.

The prices of goods that are owned by agents that are envied by some other agent are not increased. Agents that are envied by another agent do not gain additional goods.

Total spending of least spending agent never decreases. Is increased by factor r in price increases.

Therefore, number of price increases = $O(\log_r \max u_{ij} / \min u_{ij})$.

Time between price increases is polynomial: Similar to analysis of matching algs.

Analysis of the Approximation Factor

We have computed

- an allocation x; $x_i =$ set of goods assigned to agent i.
- a price vector p; $p_j =$ price of good j.
- a vector α; $\alpha_i =$ MBB-ratio of agent i.

such that

- $\alpha_i = \max_j {}^{u_{ij}}/_{p_j}$ (α_i is maximum-bang-per-buck ratio of i)
- $j \in x_i$ implies $u_{ij}/p_j = \alpha_i$ (only MBB-goods are allocated)
- for all agents h and i, there is a good j such that

$$p(x_h \setminus j) \le (1 + \varepsilon) p(x_i),$$

where $p(\text{set } S \text{ of goods}) = \sum_{j \in S} p_j$. (no envy up to one good)

The Approximation Factor: Rescaling

Let x^{alg} be the allocation computed by the algorithm.

$$j \in x_i^{alg} \to u_{ij}/p_j = \alpha_i = \max_k {}^{u_{ik}}/_{p_k} \quad \forall h, i \, \exists j \text{ s.t. } p(x_h \setminus j) \le (1{+}\varepsilon) p(x_i).$$

Rescale: Replace u_{ij} by u_{ij}/α_i. This multiplies the NSW of *every* allocation by $\left(\prod_i \alpha_i^{-1}\right)^{1/n}$ and hence does not change the optimal allocation. The above becomes

$$j \in x_i^{alg} \to u_{ij}/p_j = 1 = \max_k {}^{u_{ik}}/_{p_k} \quad \forall h, i \, \exists j \text{ s.t. } p(x_h \setminus j) \le (1{+}\varepsilon) p(x_i)$$

and hence $u_{ij} = p_j$ whenever good j is allocated to i. If j is not allocated to i, $u_{ij} \le p_j$.

$$j \in x_i^{alg} \to u_{ij} = p_j, \quad u_{hj} \le p_j, \quad \forall h, i \, \exists j \text{ s.t. } p(x_h \setminus j) \le (1{+}\varepsilon) p(x_i)$$

The Approximation Factor: Rescaling

Let x^{alg} be the allocation computed by the algorithm.

$$j \in x_i^{alg} \rightarrow u_{ij}/p_j = \alpha_i = \max_k u_{ik}/p_k \quad \forall h, i\, \exists j \text{ s.t. } p(x_h \setminus j) \leq (1+\varepsilon)p(x_i).$$

Rescale: Replace u_{ij} by u_{ij}/α_i. This multiplies the NSW of *every* allocation by $\left(\prod_i \alpha_i^{-1}\right)^{1/n}$ and hence does not change the optimal allocation. The above becomes

$$j \in x_i^{alg} \rightarrow u_{ij}/p_j = 1 = \max_k u_{ik}/p_k \quad \forall h, i\, \exists j \text{ s.t. } p(x_h \setminus j) \leq (1+\varepsilon)p(x_i)$$

and hence $u_{ij} = p_j$ whenever good j is allocated to i. If j is not allocated to i, $u_{ij} \leq p_j$.

$$j \in x_i^{alg} \rightarrow u_{ij} = p_j, \quad u_{hj} \leq p_j, \quad \forall h, i\, \exists j \text{ s.t. } p(x_h \setminus j) \leq (1+\varepsilon)p(x_i)$$

The Approximation Factor: Rescaling

Let x^{alg} be the allocation computed by the algorithm.

$$j \in x_i^{alg} \rightarrow u_{ij}/p_j = \alpha_i = \max_k u_{ik}/p_k \quad \forall h, i\, \exists j \text{ s.t. } p(x_h \setminus j) \leq (1+\varepsilon) p(x_i).$$

Rescale: Replace u_{ij} by u_{ij}/α_i. This multiplies the NSW of *every* allocation by $\left(\prod_i \alpha_i^{-1}\right)^{1/n}$ and hence does not change the optimal allocation. The above becomes

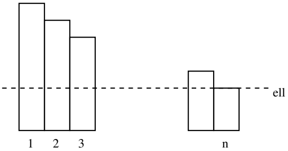
$$j \in x_i^{alg} \rightarrow u_{ij}/p_j = 1 = \max_k u_{ik}/p_k \quad \forall h, i\, \exists j \text{ s.t. } p(x_h \setminus j) \leq (1+\varepsilon) p(x_i)$$

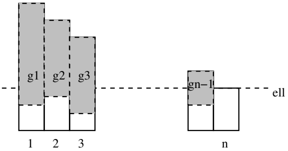
and hence $u_{ij} = p_j$ whenever good j is allocated to i. If j is not allocated to i, $u_{ij} \leq p_j$.

$$j \in x_i^{alg} \rightarrow u_{ij} = p_j, \quad u_{hj} \leq p_j, \quad \forall h, i\, \exists j \text{ s.t. } p(x_h \setminus j) \leq (1+\varepsilon) p(x_i)$$

The Approximation Factor

Rename the agents s.t. $p(x_1^{alg}) \geq p(x_2^{alg}) \geq \ldots \geq p(x_n^{alg}) =: \ell$.



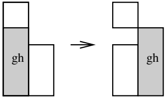


Each x_i, $1 \leq i \leq n$, contains a g_i such that $p(x_i \setminus g_i) \leq \ell$.

Give additional freedom to OPT. It must allocate g_1 to g_{n-1} integrally, can allocate the other goods fractionally. Contribution of a good is its price.

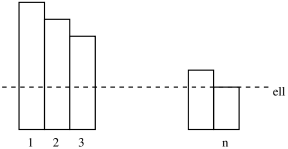
Claim: OPT does not have to allocate g_i and g_h to same agent.

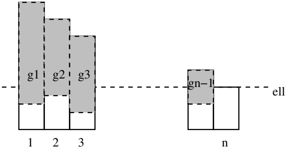
Assume otherwise. Then there is an agent a with only fractional goods. Move g_h to this agent and move $\min(p(g_h), p(x_a))$ in return. This does not decrease NSW.



The Approximation Factor

Rename the agents s.t. $p(x_1^{alg}) \geq p(x_2^{alg}) \geq \ldots \geq p(x_n^{alg}) =: \ell$.



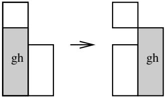


Each x_i, $1 \leq i \leq n$, contains a g_i such that $p(x_i \setminus g_i) \leq \ell$.

Give additional freedom to OPT. It must allocate g_1 to g_{n-1} integrally, can allocate the other goods fractionally. Contribution of a good is its price.

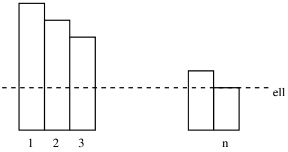
Claim: OPT does not have to allocate g_i and g_h to same agent.

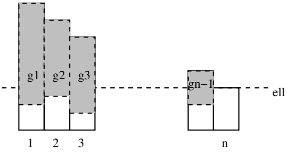
Assume otherwise. Then there is an agent a with only fractional goods. Move g_h to this agent and move $\min(p(g_h), p(x_a))$ in return. This does not decrease NSW.



The Approximation Factor

Rename the agents s.t. $p(x_1^{alg}) \ge p(x_2^{alg}) \ge \ldots \ge p(x_n^{alg}) =: \ell$.



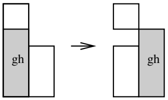


Each x_i, $1 \le i \le n$, contains a g_i such that $p(x_i \setminus g_i) \le \ell$.

Give additional freedom to OPT. It must allocate g_1 to g_{n-1} integrally, can allocate the other goods fractionally. Contribution of a good is its price.

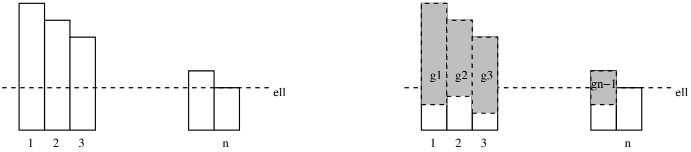
Claim: OPT does not have to allocate g_i and g_h to same agent.

Assume otherwise. Then there is an agent a with only fractional goods. Move g_h to this agent and move $\min(p(g_h), p(x_a))$ in return. This does not decrease NSW.



The Approximation Factor

Rename the agents s.t. $p(x_1^{alg}) \geq p(x_2^{alg}) \geq \ldots \geq p(x_n^{alg}) =: \ell$.

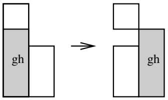


Each x_i, $1 \leq i \leq n$, contains a g_i such that $p(x_i \setminus g_i) \leq \ell$.

Give additional freedom to OPT. It must allocate g_1 to g_{n-1} integrally, can allocate the other goods fractionally. Contribution of a good is its price.

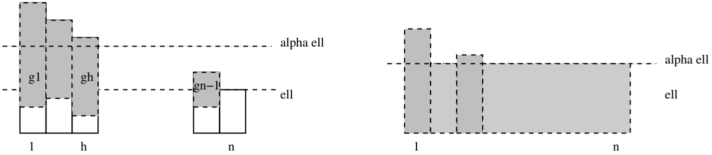
Claim: OPT does not have to allocate g_i and g_h to same agent.

Assume otherwise. Then there is an agent a with only fractional goods. Move g_h to this agent and move $\min(p(g_h), p(x_a))$ in return. This does not decrease NSW.



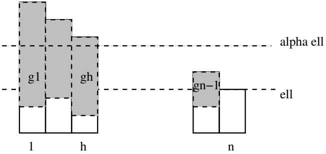
More on OPT and ALG

OPT assigns the g_i's injectively. Wlog., OPT assigns g_i to i. Let $\alpha\ell = \min_i p(x_i^{opt})$, let h be maximum such that $p(x_h^{alg}) > \alpha\ell$.



More on OPT and ALG

OPT assigns the g_i's injectively. Wlog., OPT assigns g_i to i. Let $\alpha\ell = \min_i p(x_i^{opt})$, let h be maximum such that $p(x_h^{alg}) > \alpha\ell$.

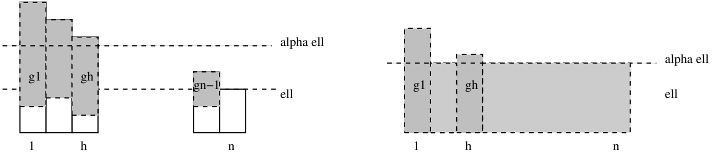


For $i \leq h$: $p(x_i^{opt}) \leq p(x_i^{alg})$.

This is clear if $p(x_i^{opt}) = \alpha\ell$. Otherwise, $x_i^{opt} = \{ g_i \}$.

More on OPT and ALG

OPT assigns the g_i's injectively. Wlog., OPT assigns g_i to i. Let $\alpha\ell = \min_i p(x_i^{opt})$, let h be maximum such that $p(x_h^{alg}) > \alpha\ell$.



For $i \le h$: $p(x_i^{opt}) \le p(x_i^{alg})$.

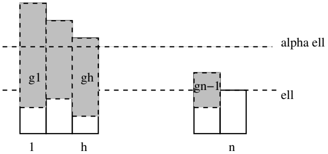
This is clear if $p(x_i^{opt}) = \alpha\ell$. Otherwise, $x_i^{opt} = \{ g_i \}$.

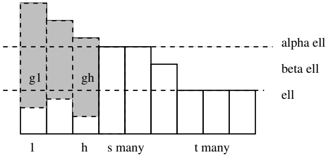
Thus

$$\text{NSW}(x^{opt}) \le \left(\prod_{i \le h} p(x_i^{alg}) \cdot (\alpha\ell)^{n-h} \right)^{1/n}.$$

More on OPT and ALG

OPT assigns the g_i's injectively. Wlog., OPT assigns g_i to i. Let $\alpha\ell = \min_i p(x_i^{opt})$, let h be maximum such that $p(x_h^{alg}) > \alpha\ell$.



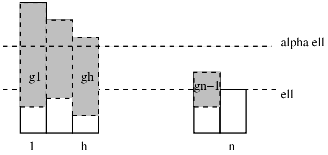


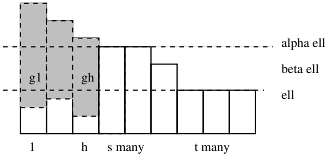
$$\mathrm{NSW}(x^{opt}) \le \left(\prod_{i \le h} p(x_i^{alg}) \cdot (\alpha\ell)^{n-h} \right)^{1/n}.$$

We now make x^{alg} worse. For agents $i > h$, we move the heights towards the bounds ℓ and $\alpha\ell$. Thus

$$\mathrm{NSW}(x^{alg}) \ge \left(\prod_{i \le h} p(x_i^{alg}) \cdot (\alpha\ell)^{s} \cdot \beta\ell \cdot \ell^{t} \right)^{1/n}.$$

More on OPT and ALG





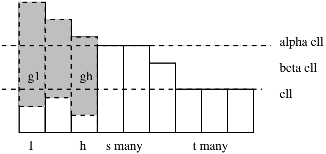
$$\mathrm{NSW}(x^{opt}) \le \left(\prod_{i \le h} p(x_i^{alg}) \cdot (\alpha \ell)^{n-h} \right)^{1/n}.$$

$$\mathrm{NSW}(x^{alg}) \ge \left(\prod_{i \le h} p(x_i^{alg}) \cdot (\alpha \ell)^{s} \cdot \beta\ell \cdot \ell^t \right)^{1/n}.$$

$$\frac{\mathrm{NSW}(x^{opt})}{\mathrm{NSW}(x^{alg})} \le \left(\frac{(\alpha\ell)^{s+1+t}}{(\alpha\ell)^s \cdot \beta\ell \cdot \ell^t} \right)^{1/n} = \left(\alpha^t \cdot \frac{\alpha}{\beta} \right)^{1/n}.$$

More on OPT and ALG

$$\frac{\mathrm{NSW}(x^{opt})}{\mathrm{NSW}(x^{alg})} \le \left(\alpha^t \cdot \frac{\alpha}{\beta}\right)^{1/n} \le \left(\frac{t\alpha + \alpha/\beta}{t+1}\right)^{(t+1)/n}$$



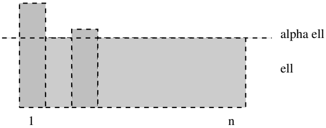
$$\alpha\ell(s+t+1) \le s\alpha\ell + \beta\ell + t\ell + h\ell$$

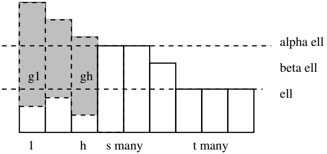
and hence $t\alpha + \alpha/\beta \le \beta + t + h + \alpha/\beta - \alpha \le t + h + 1$. Thus

$$\frac{\mathrm{NSW}(x^{opt})}{\mathrm{NSW}(x^{alg})} \le \left(\frac{t+h+1}{t+1}\right)^{(t+1)/n} \le \left(\frac{n}{t+1}\right)^{(t+1)/n} \le e^{1/e} \approx 1.45.$$

More on OPT and ALG

$$\frac{\mathrm{NSW}(x^{opt})}{\mathrm{NSW}(x^{alg})} \le \left(\alpha^t \cdot \frac{\alpha}{\beta}\right)^{1/n} \le \left(\frac{t\alpha + \alpha/\beta}{t+1}\right)^{(t+1)/n}$$





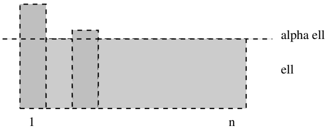
$$\alpha\ell(s+t+1) \le s\alpha\ell + \beta\ell + t\ell + h\ell$$

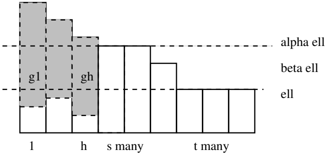
and hence $t\alpha + \alpha/\beta \le \beta + t + h + \alpha/\beta - \alpha \le t + h + 1$. Thus

$$\frac{\mathrm{NSW}(x^{opt})}{\mathrm{NSW}(x^{alg})} \le \left(\frac{t+h+1}{t+1}\right)^{(t+1)/n} \le \left(\frac{n}{t+1}\right)^{(t+1)/n} \le e^{1/e} \approx 1.45.$$

More on OPT and ALG

$$\frac{\text{NSW}(x^{opt})}{\text{NSW}(x^{alg})} \le \left(\alpha^t \cdot \frac{\alpha}{\beta}\right)^{1/n} \le \left(\frac{t\alpha + \alpha/\beta}{t+1}\right)^{(t+1)/n}$$





$$\alpha\ell(s+t+1) \le s\alpha\ell + \beta\ell + t\ell + h\ell$$

and hence $t\alpha + \alpha/\beta \le \beta + t + h + \alpha/\beta - \alpha \le t + h + 1$. Thus

$$\frac{\text{NSW}(x^{opt})}{\text{NSW}(x^{alg})} \le \left(\frac{t+h+1}{t+1}\right)^{(t+1)/n} \le \left(\frac{n}{t+1}\right)^{(t+1)/n} \le e^{1/e} \approx 1.45.$$

Generalization to Multiple Copies, Diminishing value, Budget-Additive

For each agent i and good j (k_j copies of good j)

$$u_{ij1} \geq u_{ij2} \geq \ldots \geq u_{ijk_j}.$$

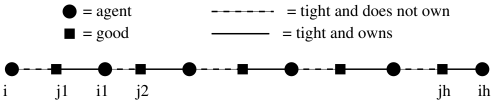
Let $m(x_i, j)$ be the multiplicity of good j in x_i. Then

$$u_i(x_i) = \min(c_i, \sum_j \sum_{1 \leq \ell \leq m(x_i,j)} u_{ij\ell}).$$

An agent is capped if $u_i(x_i) = c_i$. Only uncapped agents envy.

MBB-Invariant: $u_{ijm(x_i,j)+1}/p_j \leq \alpha_i \leq u_{ijm(x_i,j)}/p_j$ for all i and j.

Tight path:



(i, j) is tight: α_i = left endpoint for "does not own" and α_i = right endpoint for "owns".

Open Problems

- Is $e^{1/e} \approx 1.45$ the best approximation factor for this algorithm? I know a lower bound of $1.44 = 3^{1/3}$.
- How does one compute exact solutions?
- How does one compute good upper bounds on NSW(OPT)?
- What is the best approximation factor for this problem? Upper bound is 1.45, lower bound is 1.00008.
- Distributed implementation?