$(\phi - 1)$-EFX for n agents with additive valuations

ϕ is the positive solution to the quadratic equation $x^2 - x - 1 = 0$. $\phi = \frac{1+\sqrt{5}}{2} \simeq 1.618$ & $\phi - 1 = \phi^{-1} \approx 0.618$

* $N = \{1, \ldots, n\}$: set of agents
* M : set of m goods
* v_i s are additive
* An allocation A is α-EFX, if $\forall i, j \in N$

$$v_i(A_i) < v_i(A_j) \Rightarrow \forall g \in A_j, \; v_i(A_i) \geq \alpha v_i(A_j \setminus g)$$

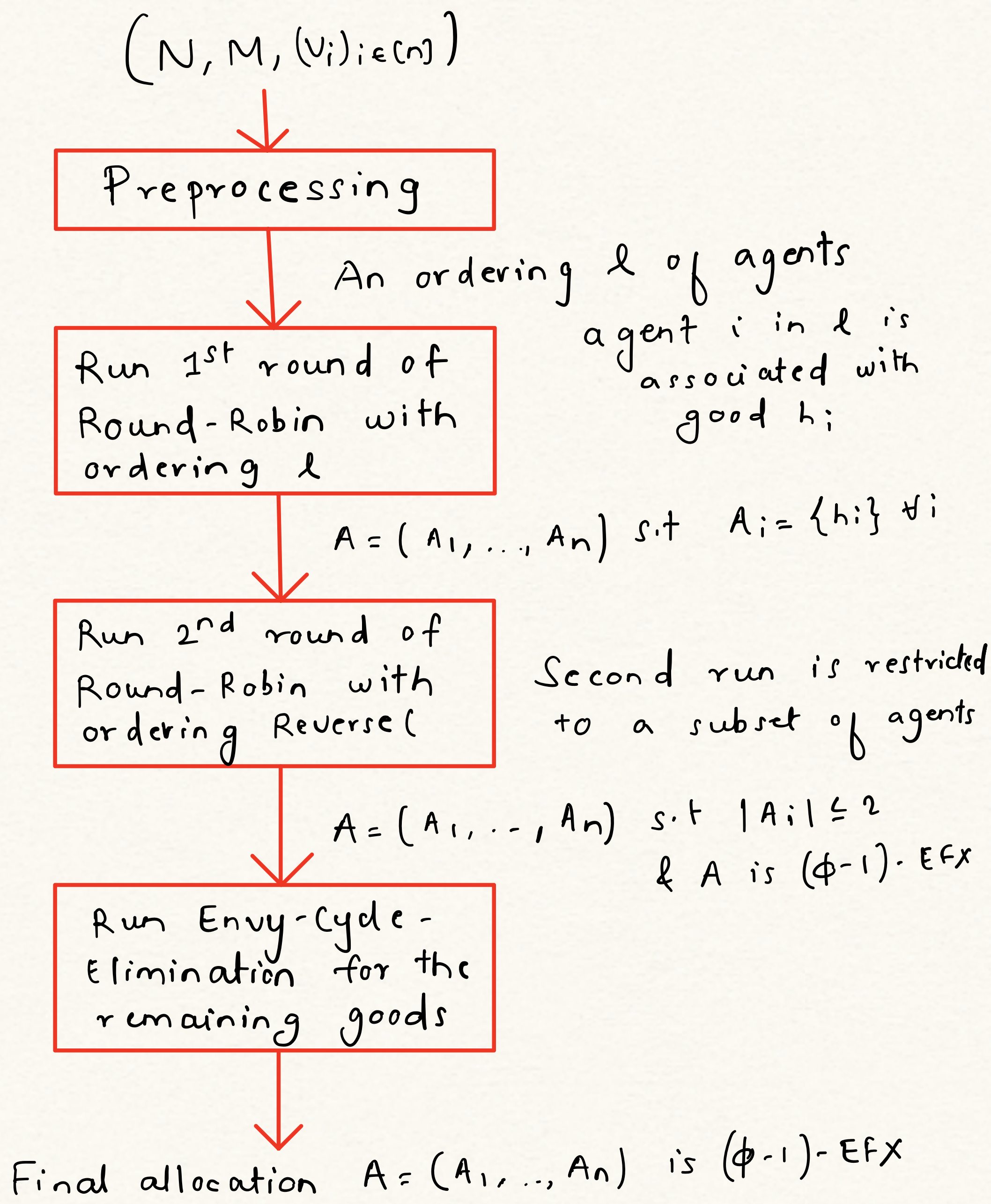
What is Known?

* $\frac{1}{2}$-EFX for n agents with subadditive valuations (Plaut & Roughgarden, 2018)
* $(\phi-1)$-EFX for n agents with additive valuations (Amanatidis, Markakis & Ntokos, 2020) (BEST KNOWN)
* $\frac{2}{3}$-EFX under restricted setting ($n \leq 7$ with additive vals, n agents with $v_i(g) \in \{a, b, c\}$ where $a > b > c \geq 0$) (Amanatidis, Ratsikas & Sgouritsa)

We will prove the following result.

Theorem: Given an instance $(N, M, (v_i)_{i\in[n]})$ with additive valuations, a $(\phi - 1)$-EFX allocation can be computed in polynomial time.

Schematic overview



ALGORITHM 1 : PREPROCESSING

```
L = ϕ, S = N, k = 1
while S ≠ ϕ do
{
    Let i be the lexicographically first
    agent of S
    h_i = arg max_{g ∈ M} v_i(g)
    t_i = m - |M| + 1            (/* i's timestamp */)
    Let R = (N \ S ∪ L) ∪ {i}
    j = arg max_{t ∈ R} v_i(h_t)
    if ϕ v_i(h_i) < v_i(h_j) then
    {
        h_i = h_j
        L = L ∪ {i}
        l[k] = i
        k = k + 1
        S = (S \ {i}) ∪ {j}
    }
    else
    {  S = S \ {i}, M = M \ {h_i}
    }
}
for every i ∈ N \ L in order of increasing timestamp t_i
{
    l[k] = i, k = k + 1
}
return (l, |L|)
```

L = set of high priority agents / locked agents (Their associated goods cannot be claimed)	R = set of processed agents (whose goods can be claimed) union the current agent
S = set of agents that still need to be processed or reprocessed	M = set of unassociated goods
h_i = good associated with agent i	ℓ = desired ordering of agents

Example:

	g_1	g_2	g_3	g_4	g_5	g_6
1	10	9	5	4	3	1
2	2	10	8	4	3	1
3	10	2	5	4	3	1
4	1	2	3	10	5	4

i	S	L	ℓ	h_i	t_i	R	$\phi v_i(h_i) < v_i(h_j)$
1	{1,2,3,4}	ϕ	( )	g_1	1	{1}	1.618 (10) < 10 (F)
2	{2,3,4}	ϕ	( )	g_2	2	{1,2}	1.618 (10) < 10 (F)
3	{3,4}	ϕ	( )	g_3	3	{1,2,3}	1.618 (5) < 10 (T) (j=1)
1	{1,4}	{3}	(3)	g_3	3	{1,2}	1.618 (5) < 9 (T) (j=2)
2	{2,4}	{3,1}	(3,1)	g_3	3	{2}	1.618 (8) < 8 (F)
4	{4}	{2,1}	(3,1)	g_4	4	{2,4}	1.618 (10) < 10 (F)

After the while loop ends

$N \setminus L = \{2,4\}$ $\quad t_2 = 3 < t_4 = 4$

Append these agents in increasing order of their timestamps

$(3,1) \longrightarrow (3,1,2) \longrightarrow (3,1,2,4)$

$\ell = (3,1,2,4)$ $\quad$ & $|L| = 2$

$h_3 = g_1 \quad h_1 = g_2 \quad h_2 = g_3 \quad h_4 = g_4$

Let us establish some useful properties of the goods associated with the agents.

Claim 1 : At the end of the execution of Algorithm 1 each agent i is associated with a single good h_i, so that

(i) $v_i(h_i) > \phi\, v_i(g)$, for any $i \in L$ & $g \in M \setminus \bigcup_{k=1}^{n} \{h_k\}$,

(ii) $\phi\, v_i(h_i) \geq v_i(h_j)$, for any $i,j \in N \setminus L$

Proof :

(i) Fix some $i \in L$ and consider the last iteration of the while loop where i was the lexicographically first agent of S, during which i was added to L

M' = set of unassociated goods

h_i^{old} = i's favorite good in M'

h_i = new item associated with i after she was added to L in the if block

We know $M' \supseteq M \setminus \bigcup_{k=1}^{n} \{h_k\}$ & $g \in M \setminus \bigcup_{k=1}^{n} \{h_k\}$

Since h_i is the new item associated with i, we have

$$v_i(h_i) > \phi\, v_i(h_i^{old}) \geq \phi\, v_i(g)$$

(ii)

REMARK
* Multiple agents in L may have the same timestamp
* Every agent in $N \setminus L$ has a distinct timestamp
* $i \in L$ & $j \in N \setminus L$ may have the same timestamp

Consider the last iteration in which some agent k claimed i's good h_i
" " " " " " " " k' " j's " h_j

Then, $t_k \neq t_{k'}$ $\quad t_i = t_k \neq t_{k'} = t_j \Rightarrow t_i \neq t_j$

Suppose for contradiction that $t_k = t_{k'}$. This means either k claimed k''s associated good or k' claimed k's associated good (Note: if agent x claims agent y's good then in the next iteration y gets the same timestamp)

Suppose k claimed k's good. The other case is symmetric. We know an agent can claim other agent's associated good at most once (an agent can enter L at most once). Therefore, we must have $i = k'$.

We also know k' claimed j's associated good h_j which means $i = k' \in L$, a contradiction (because $i \in N \setminus L$).

$\therefore t_i \neq t_j$

① If $t_i < t_j$ then

$$\boxed{v_i(h_i) \geq v_i(h_j)} \Rightarrow \phi\, v_i(h_i) \geq \phi\, v_i(h_j) > v_i(h_j)$$

h_j was available when i chose h_i

② If $t_i > t_j$ then

$$\boxed{\phi\, v_i(h_i) \geq v_i(h_j)}$$

otherwise ($\phi\, v_i(h_i) < v_i(h_j)$) i would be added to L irrevocably which is a contradiction

$\square$

Consider the following parameterized version of Round-Robin

ALGORITHM 2 : ROUND-ROBIN(N, A, M', ℓ, τ)

N: set of agents
A: partial allocation
M': set of unallocated goods
ℓ: ordering of N
τ: number of steps

$k = 1$
while $M' \neq \phi$ and $\tau > 0$ do
{
$$g = \arg\max_{h \in M} v_{\ell[k]}(h)$$
$$A_{\ell[k]} = A_{\ell[k]} \cup g$$
$$M' = M' \setminus g$$
$k = k+1 \bmod n$
$\tau = \tau - 1$
}
return (A, M')

ALGORITHM 3 : DRAFT-AND-ELIMINATE

1. $(\ell, n') =$ PREPROCESSING $(N, M, (v_i)_{i\in[n]})$

2. $A = (\phi, \ldots, \phi)$

3. $(A, M') =$ ROUND-ROBIN (N, A, M, ℓ, n) (1^{st} round)

4. $\ell^r = (\ell[n], \ldots, \ell[1])$

5. $(A, M') =$ ROUND-ROBIN $(N, A, M', \ell^r, n-n')$ (2^{nd} round)

6. $A =$ ENVY-CYCLE-ELIMINATION (N, A, M')

7. return A

Claim 2 : The partial allocation produced in Line 3 of Algorithm 3 is $A = (\{h_1\}, \{h_2\}, \ldots, \{h_n\})$ where h_i s are the goods associated with the agents after the end of Algorithm 1.

Proof :

(i) We will use induction to show that the agents in L receive their associated goods

$$A_{\ell[k]} = \{h_{\ell[k]}\} \text{ for every } k \in [|L|]$$

Base case: Easy to see that $l[1]$ receives $h_{l[1]}$

Induction:

Assume all agents from $l[1]$ to $l[k-1]$ receive their associated goods $h_{l[1]}$ to $h_{l[k-1]}$

Consider the iteration in which $l[k]$ was the lexicographically first agent of S, during which $l[k]$ was added to L.

At that time $L = \{ l[1], \ldots, l[k-1] \}$ which means all the goods $\{ h_{l[1]}, \ldots, h_{l[k-1]} \}$ are locked & unavailable.

$l[k]$ receives $h_{l[k]}$ which is her favorite among the ones assigned outside L & also better than her favorite good initially assigned.

$$\therefore A_{l[k]} = \{ h_{l[k]} \}$$

This proves the induction step

(ii)

L_1 = set of agents in L who claimed i's associated good during the course of Algorithm 1

$L_2 \setminus L_1$ = set of agents in L who never claimed a good from i but i envies their associated goods

L_3 = set of agents in L whom i does not envy

$$\boxed{\begin{array}{c} L_1 \subseteq L_2 \\ L_2 \cup L_3 = L \end{array}}$$

In Algorithm 1, agent i gets associated with her favorite good if we exclude L_2 and T

In Algorithm 2, agent i receives her favorite good if we exclude L and T

i prefers h_i over the goods associated with agents in L_3

$\therefore$ $A_i = \{h_i\}$ after line 3 of Algorithm 3

<u>Note</u>: Goods associated with the agents in L_3 were available for i before getting locked but i didn't choose them.

Now we are ready to prove the fairness guarantees of Algorithm 3

Claim 3 : Algorithm 3 returns an EF1 allocation.

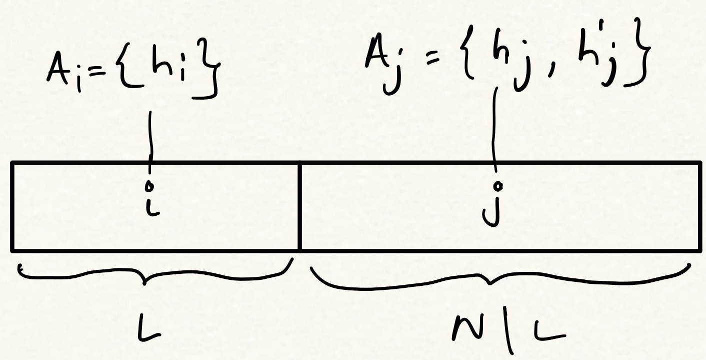
Proof : Easy! (prove yourself)

Claim 4 : The partial allocation obtained in Line 3 of Algorithm 3 is EFX.

Proof : Easy!

Claim 5 : The partial allocation A obtained in Line 5 of Algorithm 3 is $(\phi-1)$-EFX.

Proof:



Let the good received by an agent $j \in N \setminus L$ in the second round be denoted by h_j'

(i) Any envy within L is EFX. Also, any envy towards an agent $i \in L$ is EFX.

(ii) Consider $i \in L$ and $j \in N \setminus L$. By ordering ℓ we know i chose her favorite good before j

$$\therefore v_i(h_i) \geq v_i(h_j) \;\; \& \;\; v_i(h_i) \geq v_i(h'_j)$$

$$v_i(A_i) = v_i(h_i) \geq v_i(A_j \setminus g) \;\; \forall g \in A_j \quad \text{where } A_j = \{h_j, h'_j\}$$

(iii) Let us prove that any envy within $N \setminus L$ is $(\phi - 1)$ - EFX

Consider $i, j \in N \setminus L$ s.t $\ell[i] < \ell[j]$

$$A_i = \{h_i, h'_i\} \quad A_j = \{h_j, h'_j\}$$

$i \rightarrow j$

As earlier, any envy from i to j is EFX because $v_i(h_i) \geq v_i(h_j)$ and $v_i(h_i) \geq v_i(h'_j)$

$j \rightarrow i$

We know $v_j(h'_j) \geq v_j(h'_i)$ because in the second round, j chose her favorite good before i.

Since $i, j \in N \setminus L$, using claim 1, we get

$$\phi\, v_j(h_j) \geq v_j(h_i)$$

$$v_j(h_j) \geq \phi^{-1} v_j(h_i) = (\phi - 1)\, v_j(h_i)$$

$$\begin{aligned} v_j(A_j) = v_j(\{h_j, h_j'\}) &\geq (\phi-1)\, v_j(h_i) + v_j(h_i') \\ &\geq (\phi-1)\,(v_j(h_i) + v_j(h_i')) \\ &= (\phi-1)\, v_j(\{h_i, h_i'\}) \\ &= (\phi-1)\, v_j(A_i) \end{aligned}$$

▨

Claim 6: The allocation A returned by the Envy-Cycle Elimination algorithm is $(\phi-1)$-EFX.

Proof: We will use induction.

Base case: The allocation obtained in Line 5 of Algorithm 3 is $(\phi-1)$-EFX by Claim 5.

Induction:

It is easy to check that cycle resolution preserves $(\phi-1)$-EFX. So, we will only consider the case where a new good g is assigned to a source s.

Let A & A' be the allocations before & after assigning a good to a source s, respectively.

Let $i \neq s$. Consider two cases: (i) $i \in L$ & (ii) $i \notin L$

$A'_s = A_s \cup g$ $\qquad$ $A'_i = A_i$

• s $\qquad$ • $i \in L$

(i) $i \in L$

Since s is a source, $v_i(A_i) \geq v_i(A_s)$

$$v_i(A_i) \geq v_i(h_i) \quad \text{(Bundle of } i \text{ has only improved)}$$
$$> \phi\, v_i(g) \quad \text{(By Claim 1)}$$

$$v_i(A'_s) = v_i(A_s) + v_i(g) \leq v_i(A_i) + \phi^{-1} v_i(A_i)$$
$$= v_i(A_i) + (\phi - 1)\, v_i(A_i)$$
$$= \phi\, v_i(A_i)$$

$$\therefore \quad v_i(A'_i) = v_i(A_i) \geq \phi^{-1} v_i(A'_s) = (\phi - 1)\, v_i(A'_s)$$

(ii) $i \notin L$ ($\Rightarrow i \in N \setminus L$)

$A'_s = A_s \cup g$ $\quad\quad$ $A'_i = A_i$

s $\quad\quad$ $i \notin L$

$v_i(A_i) \geq v_i(A_s)$ (s is a source)

$v_i(h_i) \geq v_i(h'_i) \geq v_i(g)$ (i preferred h'_i over g in the second round of Round-Robin)

$$\frac{1}{2} v_i(A_i) \geq \frac{1}{2} v_i(\{h_i, h'_i\}) \geq v_i(g)$$

$$\therefore v_i(A'_s) = v_i(A_s) + v_i(g) \leq v_i(A_i) + \frac{1}{2} v_i(A_i) \leq \frac{3}{2} v_i(A_i) \leq \phi\, v_i(A_i)$$

$$\therefore v_i(A'_i) = v_i(A_i) \geq \phi^{-1} v_i(A'_s) = (\phi - 1)\, v_i(A'_s)$$

Since bundle value of s has improved in A' & A' restricted to $N \setminus \{s\}$ is $(\phi-1)$-EFX by induction hypothesis, A' is $(\phi-1)$-EFX. ∎

Theorem: Given an instance $(N, M, (v_i)_{i \in [n]})$ with additive valuations, a $(\phi-1)$-EFX allocation can be computed in polynomial time.

Proof: $(\phi-1)$-EFX follows from Claim 6.

It is easy to check that all algorithms run in polynomial time.

∎