Maximizing Nash Social Welfare in Online Settings

Edgar Moreno Martínez

(Work with Max, Suho and Kiarash)

27th April 2023

Overview

Introduction

Set of agents $i \in [N]$ and **indivisible** items $t \in [T]$ that arrive **online**. Each agent has a valuation function $v_i : T \to \mathbb{R}^{\geq 0}$. For groups of items we sum the valuations.

Introduction

Set of agents $i \in [N]$ and **indivisible** items $t \in [T]$ that arrive **online**.
Each agent has a valuation function $v_i : T \to \mathbb{R}^{\geq 0}$. For groups of items we sum the valuations.
Given a partition $T = \bigsqcup_i T_i$ maximize:

$$NSW(\{T_i\}) = \left(\prod_i v_i(T_i) \right)^{1/N}.$$

Introduction

Set of agents $i \in [N]$ and **indivisible** items $t \in [T]$ that arrive **online**.
Each agent has a valuation function $v_i : T \to \mathbb{R}^{\geq 0}$. For groups of items we sum the valuations.
Given a partition $T = \bigsqcup_i T_i$ maximize:

$$NSW(\{T_i\}) = \left(\prod_i v_i(T_i)\right)^{1/N}.$$

The offline problem is NP-complete and $APX-$hard (best factor 1.45).
For the divisible one are some tight online algorithms (with some extra assumptions).

What to expect?

We focus on **Competitive ratio**. Given an algorithm $\textsc{Alg}$,

$$CR = \sup_{\mathcal{I}} \frac{\mathrm{NSW}_{\textsc{Opt}}(\mathcal{I})}{\mathrm{NSW}_{\textsc{Alg}}(\mathcal{I})}$$

where $\mathcal{I}$ is a valid input. Note that $\mathrm{NSW}_{\textsc{Alg}}(\mathcal{I})$ can be the expected value of the NSW if the algorithm is randomized.

What to expect?

We focus on **Competitive ratio**. Given an algorithm $\textsc{Alg}$,

$$CR = \sup_{\mathcal{I}} \frac{\mathrm{NSW}_{\textsc{Opt}}(\mathcal{I})}{\mathrm{NSW}_{\textsc{Alg}}(\mathcal{I})}$$

where $\mathcal{I}$ is a valid input. Note that $\mathrm{NSW}_{\textsc{Alg}}(\mathcal{I})$ can be the expected value of the NSW if the algorithm is randomized.

- Restricted scenarios.
- Lower bounds on the CR and some algorithms (upper bounds).

Strong LB

Definition (Random order)

We take all the permutations of some input and calculate the expected NSW.

Theorem

For any (randomized) algorithm, the competitive ratio is at least $e^{\Theta(N)}$, even if the algorithm knows the total utility of agents beforehand and the items arrive in random order.

Sketch of the proof

For $N = T = 3$ consider:

Items \ Agents	**1**	**2**	**3**
1	1	1	1
2	1	0	0
3	0	1	1

$CR = \frac{1}{0\cdot 1/9 + 1\cdot 8/9} = 9/8$. If we copy for $3n$ agents $\to (9/8)^n$.

Equal valuations

Definition (Equal valuations setting)

We have $\forall i, j \in [N] v_i = v_j \equiv v$.

Note: the offline version is at least NP-hard via `Partition`.

Lower bound

Lemma

Let $M_n = \max \prod a_i$ *s.t.* $\sum_{i \in s} a_i = n$ *for free* s*. If* $n = 3k + r$*:*

1. $M_n = 3^k$ *for* $r = 0$
2. $M_n = 4 \cdot 3^{k-1}$ *for* $r = 1$
3. $M_n = 2 \cdot 3^k$ *for* $r = 2$

Example

For $n = 11$ we have $(3, 3, 3, 2)$.

Lower bound

Theorem

Any deterministic algorithm is lower bounded by $(M_n)^{1/n}$*, which is at most* $3^{1/3} \approx 1.4422$*, that is also the limit when* $n \to \infty$ *and the exact value for all* $n = 3k$*.*

Proof (sketch)

Consider an input with valuations $(\underbrace{1, ..., 1}_{n}, \underbrace{\infty, ..., \infty}_{\approx 2n/3})$*. The optimal will be* $(M_n \infty^{2n/3})^{1/n}$ *Any deterministic algorithm will allocate first* n *items evenly to avoid the input* $(\underbrace{1, ..., 1}_{n})$*.*

Then it gets a value of $((1+\infty)^{2n/3})^{1/n}$*.*

Greedy and EF1

Algorithm 1: Greedy

```
Given any deterministic tie-breaking rule
Initialize X_1, ..., X_n = ∅
for t = 1, 2, ..., T do
    Item g_t arrives ;
    Find the least satisfied agent j = argmin_{i∈[n]} v_i(X_i);
    X_j = X_j ∪ {g_t} ;
end
```

Definition (EF1)

For all agents i, j if $t = \text{argmax}_{f \in T_i} v(f)$, $v(T_i \backslash \{t\}) \leq v(T_j)$.

Greedy bounds

Lemma

Greedy algorithms maintains an allocation that is at most $e^{1/e}$-approximate. Note $e^{1/e} \approx 1.4446$

Proof

Follows from $EF1$ and a Barman et al. [2018] result.

Greedy bounds

Lemma

Greedy algorithms maintains an allocation that is at most $e^{1/e}$-approximate. Note $e^{1/e} \approx 1.4446$

Proof

Follows from $EF1$ *and a Barman et al. [2018] result.*

Lemma

For all $\epsilon > 0$ *any algorithm that is maintains the* $EF1$ *property returns an allocation that is at least* $(e^{1/e} - \epsilon)$*-approximate.*

Proof (sketch)

Similar to general lower bound. With $p/q \approx e$ *we build input* $n = p$ *and valuations* $(\underbrace{1, ..., 1}_{pq}, \underbrace{\infty, ..., \infty}_{p-q})$.

Optimal is

$$(p^q \infty^{p-q})^{1/p}$$

EF1 gets

$$(q^q (q + \infty)^{p-q})^{1/p}$$

$$CR = \frac{p^{q/p}}{q^{q/p}} \approx e^{1/e}$$

Randomized lower bound

Using similar ideas we can get:

Theorem

Any (possibly ***randomized****) algorithm has a competitive ratio of at least* 1.3692.

Randomized lower bound

Using similar ideas we can get:

Theorem

Any (possibly ***randomized****) algorithm has a competitive ratio of at least* 1.3692*.*

Recap:
Greedy CR 1.4446 $(e^{1/e})$
Deterministic LB 1.4422 $(3^{1/3})$

Small items

We have seen that the lower bounds use "infinite" items.

Theorem

If for all every item I we have $v(I) \leq fT$ where T is the sum of valuations and $f \in (0, \frac{1}{n})$ we have that the competitive ratio of the greedy algorithm is at most $(\sqrt{1 - fn} + nf/2)^{-1}$.

Proof (sketch)

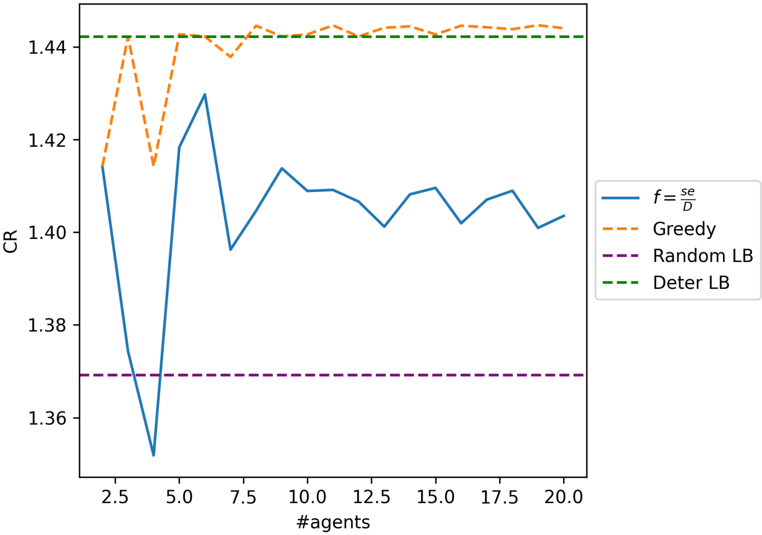
Use the the EF1 property and some approximations.

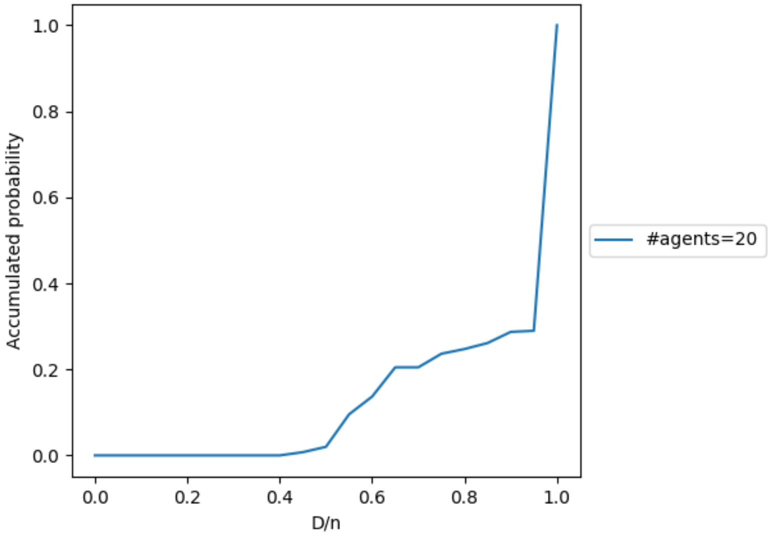
Example

If $f = 1/(2n)$ we get $CR \leq 1.044$.

Beating deterministic algorithms?

Pick $D \in [n]$ following a given distribution $\{p_i\}_{i\in[n]}$;
if $D = n$ **then**
  Run the regular greedy algorithm 1;
else
  Maintain allocations $B_i \in [D]$ and $S_i \in [n]\backslash[D]$ such that $v(B_i) > v(S_j) \forall i, j$.
  for $t = 1, 2, \ldots, T$ **do**
    Item g_t arrives ;
    $s = \sum_{i\in[n]\backslash[D]} v(S_i)$
    if $v(g_t) \geq se/n$ **then**
      Allocate item g_t greedily in S;
    else
      Allocate item g_t greedily in B;


1.44
1.42
1.40
1.38
1.36
CR
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
#agents
$f = \frac{se}{D}$
Greedy
Random LB
Deter LB


1.0
0.8
0.6
0.4
0.2
0.0
Accumulated probability
0.0
0.2
0.4
0.6
0.8
1.0
D/n
#agents=20

Affine utility setting

Generalizing the identical setting of the equal valuations section through *affine valuations*.

Definition (Affine value)

Given a base utility function $u(\cdot)$ and constants $a_i > 0, b_i \geq 0$, we define an ***affine*** *value setting to be the case in which agent i's valuation function v_i of receiving an item t satisfies $v_i(\{t\}) = a_i u(\{t\}) + b_i$.*

Theorems

Theorem

Any problem instance with parameters $(a_i, b_i)_{i\in[n]}$, can be reduced to a problem instance with $(1, b'_i)_{i\in[n]}$.

Theorem

Any ***deterministic*** *algorithm lacks from an* ***arbitrarily large*** *CR even if $n = 2$.*

Approximation

If $v = \min_{t\in[T]} u(\{t\})$ and $b = \min_{i\in[n]} b_i$.

$$B = \left(\prod_{i\in[n]} \frac{v+b_i}{v+b}\right)^{1/n} \tag{1}$$

Theorem

There exists an algorithm with a competitive ratio at most $Be^{1/e}$.

Binary and bivalue

Definition

Binary: $v_i(\cdot) \in \{0, 1\} \forall i \in [N]$

m-*Bivalue:* $v_i(\cdot) \in \{1, m\} \forall i \in [N]$ *for* $m > 1$

Binary with many items

Observation

The exponential lower bound of the first section used binary valuations.

Binary with many items

Observation

The exponential lower bound of the first section used binary valuations.

Theorem

In the binary value setting, if every agent values positively at least n items the greedy algorithm has a competitive ratio of at most $\Theta(n)$.

Lemma

If an agent positively values kn items the greedy algorithm allocates him at least k items.

The greedy is asymptotically optimal

Theorem

Any algorithm has a competitive ratio of at least $\Theta(n)$ even if every agent values at least a given number of items (that can grow with n).

Proof (sketch)

Receive n rounds of items. On the first round the items are valued 1 by all agents. In each round an agent pass to value items by 0. In each round the number of received items is $>>$ that in the previous one.

m-Bivalue setting

Theorem

The round robin algorithm has at most a CR of m.

m-Bivalue setting

Theorem

The round robin algorithm has at most a CR of m.

Theorem

The greedy algorithm has at least a CR of $m(1 - 1/n) + 1/n$.

Theorem

The round robin algorithm has at most a CR of m.

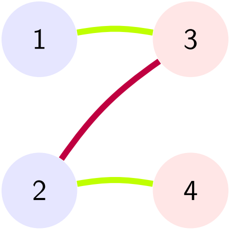
Theorem

The greedy algorithm has at least a CR of $m(1 - 1/n) + 1/n$.

Theorem

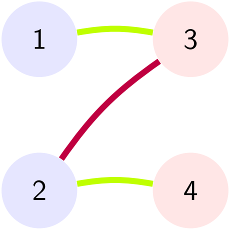
Any online algorithm has a competitive ratio of at least $m^{5/18}$.

Bipartite maximum matching



Left side agents, right side items. Edge if agent values m the item.
Inspired in the known lower bounds of this problem:

Bipartite maximum matching

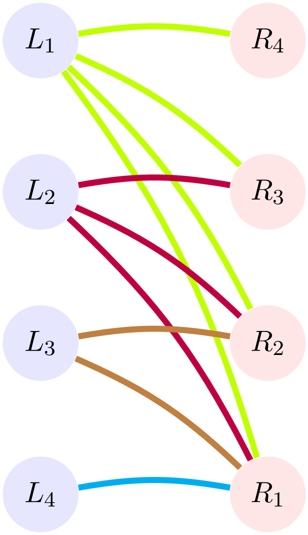


Left side agents, right side items. Edge if agent values m the item.
Inspired in the known lower bounds of this problem:

Theorem

Any random tie-breaker algorithm has a competitive ratio of at least $m^{1/2-\epsilon}$ for all $\epsilon > 0$.

Proof



Conclusions

- Focus on determining lower bounds for competitive ratio.
- Positive results in some restricted scenarios.
- Greedy algorithm performs well in equal valuations scenario.
- Greedy algorithm is asymptotically tight in large number of items with binary valuations.
- No good positive result for m-bivalue setting.

Future work

- Finishing the started ideas: closing the deterministic gap in the equal valuations setting, or proving a good randomized algorithm, or finding a good algorithm for the m-bivalue setting.
- Investigation of bounds and algorithms under the random order arrival model would be particularly interesting.

Questions?

edgarmm19@gmail.com