ENS Lyon
2024/2025 **Mathurin Massias, Titouan Vayer, Quentin Bertrand**

Homework 2 : The Auction algorithm

You have 3 weeks to do this homework: it must be returned by December 20th.

- You can do it by group of two.
- Send the homework to titouan.vayer@inria.fr, mathurin.massias@inria.fr and quentin.bertrand@inria.fr with the header "Homework 1 Name 1 Name 2". Please **name your file LASTNAME1_LASTNAME2_hw2.pdf** where your replace LASTNAME by your respective last names.
- For the maths send a scan by mail or give it by hand on December 19th.

- Exercise 1: Theory of the Auction algorithm (10 points). -

The auction algorithm solves the Monge assignment problem. To detail it, we consider N people and N objects.

We assume that for each person $i \in \{1, \cdots, N\}$, there is a certain gain a_{ij} associated with acquiring object $j \in \{1, \cdots, N\}$. This gain could represent an increase in the person's happiness, as when buying a delicious cake, or a mischievous gain tied to an increase in assets.

Since we live in a capitalist world, these people will be put in competition, and each one will have to make a dishonest bid to acquire the objects (instead of simply negotiating). Our goal is to determine the person/object assignment, which we will denote by $\sigma : \{1, \cdots, N\} \to \{1, \cdots, N\}$.

We also want this assignment to be *bijective*, meaning that each person corresponds to a unique object. Since we are not monsters, we will force each person to acquire one object, but we aim to find the pairing that maximizes the total gain (i.e. the sum of the individual gains). To achieve this, we need to solve the following optimization problem:

$$\max_{\sigma \in S_N} \sum_{i=1}^{N} a_{i\sigma(i)} \text{ where } S_N \text{ is the set of all bijections from } [\![N]\!] \to [\![N]\!]\,. \tag{1}$$

(I) Explain why solving (1) is equivalent to solving the Monge assignment problem.

The basic idea of the auction algorithm is that all persons will iteratively try to agree on the right price to pay to acquire object j, denoted p_j. We will try to find the prices such that everyone is as satisfied as possible with having purchased an object. We still need to define what it means to be "the most satisfied".

Definition 1. *Given an assignment σ and prices $\mathbf{p} = (p_1, \cdots, p_N)$, we say that a person $i \in \{1, \cdots, N\}$ is* happy *when the net cost of the object $\sigma(i)$ they purchase is the lowest possible among all objects. Formally, this means that*

$$a_{i\sigma(i)} - p_{\sigma(i)} = \max_{j \in \{1, \cdots, N\}} a_{ij} - p_j\,. \tag{2}$$

If all the people are happy, we say that the pair of assignment/price $(\sigma, \mathbf{p})$ is at equilibrium.

The definition above accurately reflects this idea because $a_{ij} - p_j$ is the net cost of object j for person i (the gain they get from acquiring the object minus the price they pay for it). A fundamental result tells us the following.

Theorem 1. *If we find a pair $(\sigma^\star, \mathbf{p}^\star)$ at equilibrium, then $\sigma^\star$ solves (1) and $\mathbf{p}^\star$ solves the* dual *problem given by*

$$\min_{\mathbf{p} \in \mathbb{R}^N} \sum_{j=1}^{N} p_j + \sum_{i=1}^{N} \max_{j \in \{1, \cdots, N\}} (a_{ij} - p_j)\,. \tag{3}$$

(II) Prove Theorem 1. Do you have any economic interpretation of this result?

The goal of the auction algorithm is to find a pair at equilibrium so as to use Theorem 1 to solve the Monge problem. We will first write a somewhat naive algorithm.

It proceeds in successive rounds. At the start of the algorithm, we choose an initial assignment and prices (often set to zero). If all the people are *happy* (Def. 1), we stop, obviously.

Otherwise, at each round $t \in \mathbb{N}$, a person who is not happy is selected – without requirements: randomly, or the first unhappy, or any other choice. For this person $i \in \{1, \dots, N\}$, we find the object $j_i \in \{1, \dots, N\}$ that makes them the happiest, i.e.

$$j_i \in \underset{j \in \{1,\dots,N\}}{\operatorname{argmax}} \, a_{ij} - p_j \, . \tag{4}$$

Then, we proceed to exchange this object with the person i' who originally had it: i' and i swap their objects.

Finally, and this is the crucial step, we will increase the price of object j_i to make it less attractive. Specifically, we will increase it minimally, under the constraint that person i would still choose it after this increase, i.e. is still *happy* about their choice after the price increase. If we denote by $j_i^2 \in \underset{j \in \{1,\dots,N\}, j \neq j_i}{\operatorname{argmax}} \, a_{ij} - p_j$ the second most attractive object for person i, then the new price of object j_i is given by

$$p_{j_i} \leftarrow p_{j_i} + \underbrace{\Big((a_{ij_i} - p_{j_i}) - (a_{ij_i^2} - p_{j_i^2}) \Big)}_{\geq 0} \, . \tag{5}$$

The important remark is that by doing this, we ensure that the prices always increase, so all the happy persons at step t remain happy in the next round $t+1$ if they keep their object. We then iterate the process.

Does this always work? Does the algorithm always terminate? The answer is no: it is possible that the price does not change in a given round (when the second most attractive object has the same net cost as the first). In this case, it can happen that multiple people fight indefinitely for the same small number of objects without the gains changing.

The true Auction algorithm The main idea is to relax the problem a bit with an $\varepsilon > 0$, giving ourselves a small margin. We will try to satisfy everyone to some extent. We adopt the following definition.

Definition 2 (ε-good pair)**.** *Let $\varepsilon > 0$. We say that a person i is* approximately happy *if*

$$a_{i\sigma(i)} - p_{\sigma(i)} \geq \max_{j \in \{1,\dots,N\}} \{a_{ij} - p_j\} - \varepsilon \, . \tag{6}$$

A pair $(\sigma, \mathbf{p})$ is ε-good if everyone is approximately happy.

A direct remark is that if we have such a pair, then everyone is happy within ε and we recover the previous definition with $\varepsilon = 0$. Moreover, we have the following result:

Proposition 1. *Let $(\sigma^\star, \mathbf{p}^\star)$ be an ε-good pair. Then the permutation $\sigma^\star$ solves the Monge problem within $N\varepsilon$, i.e.,*

$$\max_{\sigma \in S_N} \sum_{i=1}^N a_{i\sigma(i)} - N\varepsilon \leq \sum_{i=1}^N a_{i\sigma^\star(i)} \leq \max_{\sigma \in S_N} \sum_{i=1}^N a_{i\sigma(i)} \, . \tag{7}$$

Similarly, $\mathbf{p}^\star$ solves the dual problem within $N\varepsilon$. Moreover if $\forall (i,j) \in \{1, \dots, N\}^2, a_{ij} \in \mathbb{N}$ and $\varepsilon < \frac{1}{N}$, then $\sigma^\star$ solves the Monge problem.

(III) Prove Proposition 1.

The key now is to construct an ε-good pair. The auction algorithm is almost identical to the previous one, except that the prices of the objects will be increased by the same quantity as before *plus* ε. Specifically, the algorithm starts with an initialization of $(\sigma, \mathbf{p})$. If everyone is approximately happy, we stop. Otherwise, as long as there is a person who is not approximately happy:

- We select a person i who is not approximately happy. Let's say they are initially assigned to an object o at the beginning of the round.
- We find the object that makes the person the happiest:

$$j_i \in \underset{j\in\{1,\dots,N\}}{\operatorname{argmax}} \; a_{ij} - p_j \,. \tag{8}$$

  We say that the object receives a bid.
- We proceed to exchange this object with the person who originally had it. More precisely, if this person is i', i.e., if $\sigma^{-1}(j_i) = I$ at the beginning of the round, then we update $\sigma(i) \leftarrow j_i$ and $\sigma(i') \leftarrow o$.
- We denote $j_i^2 \in \underset{j\in\{1,\dots,N\}, j\neq j_i}{\operatorname{argmax}} a_{ij} - p_j$ the second most attractive object for person i. We increase the price of the object via

$$p_{j_i} \leftarrow p_{j_i} + \underbrace{\Big((a_{ij_i} - p_{j_i}) - (a_{ij_i^2} - p_{j_i^2}) + \varepsilon\Big)}_{\geq \varepsilon > 0} \,. \tag{9}$$

- We update the list of people who are not approximately happy by checking among all the people who were not approximately happy if they still are after the change in gains.

We will now formally prove that we indeed find an ε-good solution. The goal is to prove the following result.

Proposition 2. *Let $\|A\|_\infty = \max_{(i,j)\in\{1,\dots,N\}^2} |a_{ij}|$. If $\mathbf{p}^{(0)}$ is taken as a constant vector the auction algorithm terminates in at most $2N(\|A\|_\infty/\varepsilon + 1)$ iterations with an ε-good final pair. The algorithm has a time complexity of $\mathcal{O}(N^3\|A\|_\infty/\varepsilon)$.*

Before proving this, we give some intuitions about why this proposition is true. The general idea is that each object that receives a bid sees its price increase by at least ε. Eventually, its price becomes so high that no one has an interest in bidding on this object. Thus, the algorithm terminates after a while because all objects have a high price, and in this case, no one wants to change objects.

Also, it should be noted that a person becomes approximately happy immediately after acquiring an object via a bid and remains so as long as they keep this object, since the prices of the other objects can only increase during the algorithm. Moreover, we can prove by induction that from the moment an object receives a bid *for the first time*, the person assigned to this object in each subsequent round is approximately happy (note that this person may change).

Therefore, at the end of each round, the people associated with objects that have already received at least one bid are approximately happy. Finally, knowing that all prices increase, we have two possibilities: either the algorithm terminates with objects that have never received a bid but everyone is approximately happy, or each object will receive at least one bid at some point, and since the people associated with those objects are all approximately happy, the algorithm terminates. More formally, we have the following results:

Lemma 1. *Let $(\sigma^{(t)})_t$ and $(\mathbf{p}^{(t)})_t$ be the sequences of assignments/gains during the algorithm, $(S^{(t)})_t$ the sequence of persons who are approximately happy, and $(\Delta^{(t)})_t$ the sequence of objects that have received at least one bid during the previous rounds ($\Delta^{(0)} = \emptyset$).*

1) (Prices increase) $\forall t, \forall j \in \{1,\dots,N\},\ p_j^{(t+1)} \geq p_j^{(t)}$.

2) (The selected person becomes approximately happy) *For a fixed round t, if person i is selected by the algorithm, then $i \in S^{(t+1)}$.*

3) (If a person is approximately happy and does not change objects, they remain approximately happy) *For a fixed round t, if $i \in S^{(t)}$ and for all $t' \geq t$, $\sigma^{(t')}(i) = \sigma^{(t)}(i)$, then for all $t' \geq t$, $i \in S^{(t')}$.*

4) (More and more people become approximately happy) *The sequence $(S^{(t)})_t$ is increasing (in the sense of sets: $S^{(t)} \subset S^{(t+1)}$).*

5) (Objects can only receive a limited number of bids as long as there is one that has never received a bid) *Suppose that all initial prices are equal, i.e.,* $\mathbf{p}^{(0)} = \text{cte}$*, that we are at round* $t > 2(\|A\|_\infty/\varepsilon+1)$*, and that there exists* $j_1 \in \{1,\dots,N\}$ *with* $j_1 \notin \Delta^{(t)}$*. Then any object* $j \neq j_1$ *cannot have received more than* $2(\|A\|_\infty/\varepsilon + 1)$ *bids.*

6) (At the end of each round, the people associated with objects that have already received at least one bid are approximately happy) *At the end of round* t*,* $\{i \in \{1,\dots,N\} : \sigma^{(t)}(i) \in \Delta^{(t)}\} \subseteq S^{(t)}$*.*

7) (When all objects have been bid on at least once, the algorithm terminates) *If at the end of round* t *we have* $\Delta^{(t)} = \{1,\dots,N\}$*, then the algorithm terminates and all the people are approximately happy.*

(IV) Prove all the points of Lemma 1. The point 5) is the most difficult: you can prove it by contradiction by checking that if an object j_0 receives more than $n \geq 2(\|A\|_\infty/\varepsilon + 1)$ bids while another object does not have receive one bid then the last person who bid on j_0 is not happy.

(V) Deduce Proposition 2 from Lemma 1.

- Exercise 2: Practice of the Auction algorithm (10 points). -

Implement the Auction algorithm in `Python`. The function should takes as input a matrix $(a_{ij})_{ij}$, an $\varepsilon > 0$, initial prices $\mathbf{p}^{(0)}$ and a certain number of iterations. It should return an ε-good pair. To see if the solution is correct, compare with the `POT` library on synthetic data. You should obtain a matching that has the same OT cost within ε.