UNIVERSITÀ DI PISA

DIPARTIMENTO DI MATEMATICA

CORSO DI LAUREA IN MATEMATICA

TESI DI LAUREA TRIENNALE

Random optimal transport problems: two and three marginal distributions

RELATORE:

Dott. Dario Trevisan

CANDIDATO:

Alessandro Pinzi

ANNO ACCADEMICO 2018/2019

Ad Aurora...

Hurry up, we're dreaming

Contents

Introduction

In this work, we show the results we obtained studying two optimal transport problems on a finite space S: the first problem is a two marginal transport problem (i.e. we want to minimize a probability distribution on $S \times S$, with a cost function $c : S \times S \to [0, 1]$) and the other is a three marginal transport problem (the same of the first problem but we want to find a probability distribution on $S \times S \times S$ that minimizes a function $c : S \times S \times S \to [0, 1]$). These problems could be solved using the simplex algorithm, but in this work we consider the problems modified, adding a positive convex function (the Kullback-Leibler divergence between the unknown probability distribution and the uniform distribution) multiplied by a constant $\varepsilon > 0$ in the cost functions to make them convex. We modify the problem to find some algorithms more efficient than the simplex method. So we expose some algorithms to solve these problems, and we show the convergence and the accuracy of them. All the algorithms are based on the Bregman iterative method, a large class of algorithms to solve convex programming problems. Then we study, by numerical experiments, the efficiency of these algorithms and we compare them.

We study some intrinsic properties of the first two problems: we generate the cost functions randomly, with the uniform distribution on $[0, 1]$, and we study the expected values of the optimal values of the two problems as a function of $n \in \mathbb{N}$. By numerical experiments we conjecture that those sequences tend to 0, and moreover we can say the infinitesimal orders using the Dyer-Frieze-McDiarmid inequality, an important result of probability in linear programming. Then we show a good choice for the constant $\varepsilon > 0$ used in the modification of the problems, because we want that the solutions of the original problems and the solutions of the modified problems are 'near'. Finally, we study the Kullback-Leibler divergence between the optimal transport plans of the first two problems and the uniform distribution: this is a way to show that these transport plans are 'far' from the uniform distribution, just like we can expect.

Chapter 1

Optimal transport problems

In this chapter we'll show the optimal transport problems.

1.1 A two-marginal transport problem

Let S be a finite set and let $n \in \mathbb{N}$ be its cardinality. Suppose we have a function

$$C : S \times S \to [0, 1]$$

the first problem we want to solve is

$$\min_{\pi \in S_n} \sum_{i \in S} C_{i,\pi(i)} \tag{1.1}$$

where S_n is the symmetric group of $\{1, \dots, n\}$, i.e. S_n is the group whose elements are all the bijections from $\{1, \dots, n\}$ to itself. Note that, this problem is equivalent to the following problem of integer linear programming

$$\begin{aligned} \min & \sum_{i,j \in S} C_{i,j}\pi_{i,j} \qquad (1.2) \\ & \sum_{j=1}^{n} \pi_{i,j} = 1 \quad \forall i = 1, \dots, n \\ & \sum_{i=1}^{n} \pi_{i,j} = 1 \quad \forall j = 1, \dots, n \\ & \pi_{i,j} \in \{0, 1\} \quad \forall i, j = 1, \dots, n \end{aligned}$$

Integer linear programming problems are, generally, NP-problems, but in this case we have a theorem that assures us that 1.2 is equivalent to

$$\begin{aligned} \min & \sum_{i,j\in S} C_{i,j}\pi_{i,j} \\ & \sum_j \pi_{i,j} = 1 \quad \forall i \in \{1,\dots,n\} \\ & \sum_i \pi_{i,j} = 1 \quad \forall j \in \{1,\dots,n\} \\ & \pi_{i,j} \geq 0 \quad \forall i,j \in \{1,\dots,n\} \end{aligned} \tag{1.3}$$

and this is a linear programming problem which could be solved using the simplex algorithm.

We could give a probabilistic interpretation to the problem (1.3). Let $p_{i,j} = \frac{\pi_{i,j}}{n}$, the problem with the variables $p_{i,j}$ is

$$\begin{aligned} \min & \sum_{i,j\in S} C_{i,j}p_{i,j} \\ & \sum_j p_{i,j} = \frac{1}{n} \quad \forall i \in \{1,\dots,n\} \\ & \sum_i p_{i,j} = \frac{1}{n} \quad \forall j \in \{1,\dots,n\} \\ & p_{i,j} \geq 0 \quad \forall i,j \in \{1,\dots,n\} \end{aligned} \tag{1.4}$$

This is an optimal transport problem, i.e. in this problem we want to minimize the cost of a distribution of probability on $S \times S$ and the constraints assign the two marginal distributions: in this case both the marginals are the uniform distribution.

1.1.1 A modification of the problem

The problem (1.4) could be solved using the simplex algorithm, since it is a linear programming problem. But, in this work, we will analyze another iterative algorithm to approximate the solution of a modification of the classic transport problem: using the convex function $f(x) = x\log(x)$ we will make the function to minimize strictly convex. To introduce the new problem, we need to give a definition.

Definition 1. If P and Q are discrete probability distributions on a space Ω, we define the Kullback-Leibler divergence as

$$KL(P||Q) = \sum_{x\in\Omega} P(x) \log\left(\frac{P(x)}{Q(x)}\right)$$

Remark 2. In our case, the space $\Omega = S \times S$ is finite, and given another probability distribution q on it the Kullback-Leibler divergence is

$$\sum_{i,j} p_{i,j} \log\left(\frac{p_{i,j}}{q_{i,j}}\right)$$

Remark 3. If we consider q as the uniform distribution on $S \times S$, we obtain

$$KL(p||q) = 2\log(n) + \sum_{i,j} p_{i,j} \log p_{i,j}$$

Proposition 4. *If P, Q are discrete probability distributions on Ω, then*

$$KL(P||Q) \geq 0$$

with equality if and only if $P = Q$.

Proof. We observe that

$$-KL(P||Q) = \sum_{i,j} (p_{i,j} \log q_{i,j} - p_{i,j} \log p_{i,j})$$

is a strictly concave function in the variables $p_{i,j}$ (due to the fact that the function $-x \log x$ is a strictly concave function). Using Jensen's inequality, we obtain

$$\begin{aligned} -KL(P||Q) = \sum_{x\in\Omega} P(x) \log\left(\frac{Q(x)}{P(x)}\right) \leq \\ \leq \log\left(\sum_{x\in\Omega} P(x)\left(\frac{Q(x)}{P(x)}\right)\right) = \\ = \log\left(\sum_{x\in\Omega} Q(x)\right) = \log 1 = 0 \end{aligned}$$

with equality iff $P = Q$ due to the strictly concavity. □

Now we are ready to modify our problem, making the cost function a strictly convex function: let $\varepsilon > 0$ be a 'small' positive constant, then the modification of the linear programming problem is

$$\begin{aligned} \min & \sum_{i,j \in S} (C_{i,j} p_{i,j}) + \varepsilon KL\left((p_{i,j}) || \left(\frac{1}{n^2}\right)\right) \\ & \sum_j p_{i,j} = \frac{1}{n} \quad \forall i \in \{1, \dots, n\} \\ & \sum_i p_{i,j} = \frac{1}{n} \quad \forall j \in \{1, \dots, n\} \\ & p_{i,j} \geq 0 \quad \forall i, j \in \{1, \dots, n\} \end{aligned} \tag{1.5}$$

where $\left(\frac{1}{n^2}\right)$ is the uniform distribution on $S \times S$.

1.2 A three-marginal transport problem

Following the structure of the problem (1.4) we can define a transport problem with three marginals. Precisely, we consider a function

$$C : S \times S \times S \to [0, 1]$$

and, like in the previous section, we define the problem

$$\begin{aligned} \min & \sum_{i,j,k \in S} C_{i,j,k} p_{i,j,k} \\ & \sum_{j,k} p_{i,j,k} = \frac{1}{n} \quad \forall i \in \{1, \dots, n\} \\ & \sum_{i,k} p_{i,j,k} = \frac{1}{n} \quad \forall j \in \{1, \dots, n\} \\ & \sum_{i,j} p_{i,j,k} = \frac{1}{n} \quad \forall k \in \{1, \dots, n\} \\ & p_{i,j,k} \geq 0 \quad \forall i, j, k \in \{1, \dots, n\} \end{aligned} \tag{1.6}$$

Remember that in the problem with two marginals, we started from an assignment problem. With three dimensions, that problem could be written like

$$\min_{\pi_2, \pi_3 \in S_n} \sum_{i \in S} C_{i, \pi_2(i), \pi_3(i)} \tag{1.7}$$

but, in the next chapter (using the simplex algorithm), we'll see that (1.7) and (1.6) are not equivalent.

1.2.1 A modification of the problem

Remembering the definition of the Kullback-Leibler divergence, we consider a modification of the the problem (1.6) adding to the cost function the KL divergence between p and the uniform distribution $\left(\frac{1}{n^3}\right)$ on $S \times S \times S$. The new problem, fixed a positive costant $\varepsilon > 0$, is

$$\begin{aligned} \min & \sum_{i,j,k \in S} C_{i,j,k} p_{i,j,k} + \varepsilon KL\left((p_{i,j,k}) || \left(\frac{1}{n^3}\right)\right) \\ & \sum_{j,k} p_{i,j,k} = \frac{1}{n} \quad \forall i \in \{1, \dots, n\} \\ & \sum_{i,k} p_{i,j,k} = \frac{1}{n} \quad \forall j \in \{1, \dots, n\} \\ & \sum_{i,j} p_{i,j,k} = \frac{1}{n} \quad \forall k \in \{1, \dots, n\} \\ & p_{i,j,k} \geq 0 \quad \forall i, j, k \in \{1, \dots, n\} \end{aligned} \tag{1.8}$$

1.3 The simplex algorithm

The simplex algorithm is an iterative algorithm used to solve linear programming problems, i.e. problems with linear cost function and linear constraints. The algorithm is described in the second chapter.
It solves a wide range of problems, but we can't guarantee that the algorithm has polynomial time, in fact we are sure that the algorithm ends in $O(2^n)$ iterations, but this isn't a good estimation. Experimentally the number of iteration is far from that estimation, but the problem (1.4) and (1.6) have a particular form, so we prefer also to find other algorithms to solve them.
In the next section we introduce some algorithms to solve the problem (1.5) and (1.8).

1.4 The Sinkhorn algorithms

In this section we'll show some algorithms to solve the problem (1.5) and (1.8). The convergence of these algorithms and other theoretical results will be discussed in the fourth chapter.

1.4.1 Problem with two marginals

Before introducing the algorithm, we are going to do a consideration on the form of the solution using the method of Lagrange multipliers. The cost function f of our problem is

$$\sum_{i,j} \left(c_{i,j} p_{i,j} + \varepsilon p_{i,j} \log p_{i,j} + \varepsilon p_{i,j} \log n^2 \right)$$

and it has the following gradient

$$(\nabla f)_{i,j} = c_{i,j} + \varepsilon \log p_{i,j} + \varepsilon + \varepsilon \log n^2$$

With the method of Lagrange multipliers we obtain that the minimum point $\overline{p}$ satisfies

$$c_{i,j} + \varepsilon \log \overline{p}_{i,j} + \varepsilon + \varepsilon \log n^2 + \alpha_i + \beta j = 0$$

where α_i and β_j are two multipliers. So the minimum point has this form

$$\overline{p}_{i,j} = \frac{1}{n^2} e^{-\frac{c_{i,j}}{\varepsilon} - 1} e^{-\alpha(i)} e^{-\beta(j)}$$

Now, we illustrates the steps of the algorithm known as *Sinkhorn's algorithm*:

1. let $p_0(0)_{i,j} = \frac{1}{n^2} e^{-\frac{c_{i,j}}{\varepsilon} - 1}$
2. suppose we have $p_k(0)$, we define $p_k(1) = L_k p_k(0)$ where

$$L_k = diag\left(\frac{l_k}{n}\right) \in \mathbb{R}^{n \times n} \quad with \quad l_k(i) = \left(\sum_{j=1}^{n} p_k(0)_{i,j} \right)^{-1} \quad \forall i = 1, \dots, n$$

3. we define $p_k(2) = p_k(1) R_k$ where

$$R_k = diag\left(\frac{r_k}{n}\right) \in \mathbb{R}^{n \times n} \quad with \quad r_k(j) = \left(\sum_{i=1}^{n} p_k(1)_{i,j} \right)^{-1} \quad \forall j = 1, \dots, n$$

4. let $p_{k+1}(0) = p_k(2)$ and go to step 2

   In the numerical experiments, we will use a tolerance $\tau > 0$ to define the stopping criterion: in the step 4 we will control if $\|p_k(2) - p_k(0)\|_2 < \tau$, in that case we stop the algorithm, otherwise we go to step 2.

1.4.2 Problem with three marginals

Using the Lagrange multipliers method for the cost function of the problem (1.8), we can do the same considerations done for the problem (1.5), and to obtain that the minimum point is of the form

$$\overline{p}_{i,j,k} = \frac{1}{n^3} e^{-\frac{c_{i,j,k}}{\varepsilon}-1} e^{-\alpha(i)} e^{-\beta(j)} e^{-\gamma k}$$

For the problem (1.8) we'll introduce three algorithm, all based on the same idea, but with different features. Let's see the first algorithm: this is the same algorithm used for the problem with two marginals, adapted for this problem. These are the steps to follow:

1. let $p_0(0)_{i,j,k} = \frac{1}{n^3} e^{-\frac{c_{i,j,k}}{\varepsilon}-1}$
2. suppose we have $p_h(0)$, we define $p_h(1) = p_h(0) * (\alpha_h \bigotimes \mathrm{Id} \bigotimes \mathrm{Id})$ where
$$\alpha_h(i) = \frac{1}{n} \left(\sum_{j,k} p_h(0)_{i,j,k} \right)^{-1} \quad \forall i = 1, \dots, n$$
3. let $p_h(2) = p_h(1) * (\mathrm{Id} \bigotimes \beta_h \bigotimes \mathrm{Id})$ where
$$\beta_h(j) = \frac{1}{n} \left(\sum_{i,k} p_h(1)_{i,j,k} \right)^{-1} \quad \forall j = 1, \dots, n$$
4. let $p_h(3) = p_h(2) * (\mathrm{Id} \bigotimes \mathrm{Id} \bigotimes \gamma_h)$ where
$$\gamma_h(k) = \frac{1}{n} \left(\sum_{i,j} p_h(2)_{i,j,k} \right)^{-1} \quad \forall k = 1, \dots, n$$
5. let $p_{k+1}(0) = p_k(3)$ and go to step 2

where: $\mathrm{Id} = (1, \dots, 1) \in \mathbb{R}^n$; given three vector $a, b, c \in \mathbb{R}^n$ as $a \bigotimes b \bigotimes c$ we indicate a tensor T such that $T(i, j, k) = a(i)b(j)c(k)$; the operation $*$ is the component-wise multiplication between two tensors.

We observe that in the Sinkhorn algorithm for every iteration, we adopted the same order for the operations that update the tensor p_h. In the others two algorithm we'll adopt different strategy. We will call *operation one* the operation we did in step 2, *operation two* the one we did in step 3, *operation three* the one we did in step 4. Let's see the other two algorithms:

- (*Randomized Sinkhorn algorithm*) in the first algorithm, for every iteration, we choose randomly the order in which to do the three operations (note that there are 6 different ways), for example the algorithm could apply first the operation three using $p_h(0)$ to calculate $p_h(1)$, then it could calculate $p_h(2)$ using $p_h(1)$ and the operation one, and finally it could apply the last operation remained (the second) to calculate $p_h(3)$ using $p_h(2)$ and then it continues choosing another permutation of the three operations

- (*Bregman algorithm*) in this algorithm, for every iteration we apply the three operations using $p_h(0)$ (i.e. $p_h(1)$ =operation one applied using $p_h(0)$, $p_h(2)$ =operation two applied using $p_h(0)$, $p_h(3)$ =operation three applied using $p_h(0)$). Given two tensor $x, y \in \mathbb{R}^{n\times n\times n}$, we define

$$D(x,y) = \sum_{i,j,k} y_{i,j,k} - x_{i,j,k} + x_{i,j,k}(\log x_{i,j,k} - \log y_{i,j,k})$$

  and then we define $m_h(i) = D(p_h(i), p_h(0)) \ \ \forall i = 1,2,3$. Now we choose the index $l \in \{1,2,3\}$ (or one of the indexes) that realizes

$$\max_{i\in\{1,2,3\}} m_h(i)$$

  to define $p_{h+1}(0) = p_h(l)$, and then we repeat this procedure.

The convergence of the Sinkhorn's algorithm for the problem with two marginals, i.e. (1.5), has been largely analyzed in other papers. In this work our purpose is to study some algorithms to solve the problem with three marginals (1.8). For this reason we didn't give a stop criterion for the three algorithms presented for that problem: in the next chapter we'll show the numerical experiments, and for these algorithms we will give in input a number $maxit \in \mathbb{N}$ that will be the number of iteration the algorithms will do.
In the next chapter we'll see the proof of the convergence of these algorithms, but the proof won't give us how they converge: also for this reason we didn't give a stop criterion for the algorithms for the problem (1.8)

The Lagrange multipliers method suggests us to choose as initial point for our algorithms the global minimum point for the cost function: in the next chapter we'll see that this choice is fundamental to obtain the convergence of the algorithms to the minimum point of our problems.

Chapter 2

Algorithmic results

2.1 The simplex algorithms

In this section, we will summarize briefly the simplex algorithms (primal algorithm and dual algorithm). They are used to solve linear programming problems, i.e. problems like (primal problem)

$$\begin{aligned} &\max c \cdot x \\ &A \cdot x \leq b \end{aligned} \tag{P}$$

where $c : \mathbb{R}^n \to \mathbb{R}$, $x \in \mathbb{R}^n$, $A \in R^{m \times n}$, $b \in \mathbb{R}^m$, and m is the number of constraints. Note that every linear programming problem can be written in that canonic form even if some constraints are $=$ or $\geq$. In general $m > n$ and $rank(A) = n$. The elements of the set $\{x \in \mathbb{R}^n \mid A \cdot x \leq b\}$ are called feasible solutions of the problem (P).

Definition 5. Given the problem (P), we define the **dual problem** as follows

$$\begin{aligned} &\min b \cdot y \\ &A^T \cdot y = c \\ &y \geq 0 \end{aligned} \tag{D}$$

We call feasible solutions for the problem (D) the $y \in \mathbb{R}^m$ such that $y \cdot A = c$ and $y \geq 0$.

Remark 6. The dual of the dual problem is the primal problem.

It's very easy to prove that, for every feasible $x \in \mathbb{R}^n$ and for every feasible $y \in \mathbb{R}^m$ we have that $c \cdot x \leq b \cdot y$. Now we need an important linear algebra lemma that allow us to prove the theorem (8) (duality theorem).

Lemma 7. *Let $A \in \mathbb{R}^{m\times n}$, $c \in \mathbb{R}^n$, the two systems*

$$\begin{cases} A \cdot x \leq 0 \\ c \cdot x > 0 \end{cases} \qquad \begin{cases} y^T \cdot A = c \\ y \geq 0 \end{cases}$$

are mutually exclusive.

The proof of this lemma is not difficult, it is based on the fact that the representation of a cone is the set $\{x \mid A \cdot x \leq 0\}$ for some matrix A.

Theorem 8. *If (P) and (D) has both feasible solutions, we have that*

$$z(P) = \max\{c \cdot x \mid A \cdot x \leq b\} = \min\{b \cdot y \mid A^T \cdot y = c\ ,\ y \geq 0\} = z(D)$$

Proof. (D) has feasible solutions, so, for the inequality $c \cdot x \leq b \cdot y$ we have that (P) has a finite optimal solution. If $c = 0$, $z(P) = 0$ and $y = 0$ is feasible and optimal for (D). So, suppose that $c \neq 0$. Let $\overline{x}$ be an optimal solution for (P). We call $I(\overline{x})$ the set of the active constraints of $\overline{x}$, i.e. $I(\overline{x}) = \{i \mid A_i \cdot \overline{x} = b_i\}$. Observe that $I(\overline{x}) \neq \emptyset$, because $\overline{x}$ is an optimal solution and if $I(\overline{x}) = \emptyset$ we could find an admissible growing direction. Now, if we could find a $\xi \in \mathbb{R}^n$ s.t.

$$\begin{cases} A_{I(\overline{x})} \cdot \xi \leq 0 \\ c \cdot \xi > 0 \end{cases}$$

$\overline{x}$ wouldn't be an optimal solution for (P), so the following system has a solution

$$\begin{cases} \nu^T \cdot A_{I(\overline{x})} = c \\ \nu \geq 0 \end{cases}$$

Let $\overline{y}_I$ a solution of that system, we have that $\overline{y} = [\overline{y}_I, 0]$ is a feasible solution for (D). The following equalities conclude the proof:

$$\overline{y} \cdot b = \overline{y}_I \cdot b_I = \overline{y}_I \cdot A_I \cdot \overline{x} = c \cdot \overline{x}$$

□

Remark 9. We defined $I(\overline{x}) = \{i \mid A_i \cdot \overline{x} = b_i\}$. For the dual problem we define the active constraints of a feasible dual solution $\overline{y}$ like $J(\overline{y}) = \{i \mid \overline{y}_i > 0\}$.

Now we know that if both the problem has feasible solutions, they have the same optimal value. So we have that

$$c \cdot \overline{x} = \overline{y} \cdot b \iff \overline{y} \cdot A \cdot \overline{x} = \overline{y} \cdot b \iff \overline{y} \cdot (b - A \cdot \overline{x}) = 0 \qquad (2.1)$$

We say that $\overline{x}$ and $\overline{y}$ are complementary solutions if they respect the last equality in (2.1). An obvious consequence of these results is:

Proposition 10. *Let $\overline{x}$ be a feasible solution for (P), $\overline{x}$ is an optimal solution if and only if there exists a feasible $\overline{y}$ for (D) complementary to $\overline{x}$.*

Definition 11. Let $B \subset \{1, \dots, m\}$. B is a base if and only if $|B| = n$ and A_B is a non-singular matrix. We will call $\overline{x} = A_B^{-1} \cdot b_B$ the primal base solution and $\overline{y} = [c \cdot A_B^{-1}, 0]$ the dual base solution. Moreover, we say that a base is primal admissible if $A \cdot \overline{x} \leq b$, and it is dual admissible if $\overline{y} \geq 0$.

Note that, if $\overline{x} = A_B^{-1} \cdot b_B$ is a feasible solution, then it is a vertex of the polytope generated by the constraints. Vice versa, every vertex of the polytope is determined by a primal admissible base (these facts are very easy to prove). Now, we want to obtain a dual solution that is complementary to $\overline{x} = A_B^{-1} \cdot b_B$, a base feasible solution of the primal problem. If we call $N = \{1, \dots, m\} \setminus B$, we can associate to B the dual base solution

$$\overline{y} = [\overline{y}_B, \overline{y}_N] = [c \cdot A_B^{-1}, 0]$$

This solution satisfies $\overline{y} \cdot A = c$ and the couple $(\overline{x}, \overline{y})$ is complementary. But when is this a feasible solution for the dual problem? To answer this question we need to observe that to a single vertex of the polytope, there could correspond more than a base, in fact if $|I(\overline{x})| > n$, we can obtain more than a base that represent $\overline{x}$. In this case we say that $\overline{x}$ is a degenerate base solution.

Theorem 12. *Let $\overline{x}$ be a feasible base solution for (P). $\overline{x}$ is an optimal solution if and only if there exists a base B such that $\overline{x} = A_B^{-1} \cdot b_B$ and $\overline{y} = [c \cdot A_B^{-1}, 0]$ is a feasible solution for (D).*

Proof. ($\Leftarrow$) It is a consequence of (2.1).

($\Rightarrow$) Suppose that the thesis is false, so for all B base such that $\overline{x} = A_B^{-1} \cdot b_B$ there exists an index $i \in B$ such that $\overline{y}_i < 0$. Now, for all B we call $h = \min\{i \in B \mid \overline{y}_i < 0\}$ and then we define $\xi_B = -A_B^{-1} \cdot e_{B(h)}$, where $B(j)$ is the j-th index in the base B and e_j is the vector that has 1 in the j-th element and 0's in the other positions. We note that $c \cdot \xi_B > 0$, so it cannot be an admissible direction for the primal problem. Moreover, the dual problem cannot be empty, so $A_N \cdot \xi_B \not\leq 0$: now we call $\lambda_B = \min\{\lambda_i = \frac{b_i - A_i \cdot \overline{x}}{A_i \cdot \xi_B} \mid A_i \cdot \xi_B > 0\ ,\ i \in N\}$ and $k = \min\{i \in N \mid \lambda_i = \lambda_B\}$. Now we define $\hat{B} = B \cup \{k\} \setminus \{h\}$ and we observe the following facts: the direction ξ is perpendicular to the constraints A_j s.t. $j \in B\ ,\ j \neq h$; λ_B is the length of the step we could do in the direction ξ and $\hat{B}$ is a base that determines the 'new' base solution obtained by a step of length λ_B in the direction ξ. We have said

before that ξ cannot be an admissible direction, so $\lambda_B = 0$ and $\hat{B}$ is another base that determines the point $\overline{x}$. With this method we change base for the point $\overline{x}$, but for the assumptions at the beginning of the proof we can't obtain that $\overline{y}$ is a feasible dual solution. If we show that iterating this method we can't go twice or more times on a base, we have the thesis, because there is a finite number of bases for a point (this is called *Blend anticycle rule*).

Suppose that a base B is visited twice. We define $B(i)$, $h(i)$ and $k(i)$ respectively the base, the incoming index and the outgoing index at the i-th iteration. There are two iteration $v < l$ such that $B(v) = B(l) = B$ and $B(i) \neq B \ \ ,\forall\ v < i < l$. We define

$$r := \max\{h(i) \mid v \leq i \leq l\} = \max\{k(i) \mid v \leq i \leq l\}$$

Let p be an iteration such that r is the incoming index, and let q be an iteration such that r is the outgoing index. We call $\overline{y} = [\overline{y}_{B(p)}, 0]$ the dual solution at the p-th iteration and let $\xi = -A^{-1}_{B(q)} \cdot e_{B(h(q))}$. Remember that $c \cdot \xi > 0$ and $\overline{y}_{B(p)} \cdot A_{B(p)} = c$, so

$$c \cdot \xi = \overline{y}_{B(p)} \cdot A_{B(p)} \cdot \xi = \sum_{i \in B(p)} \overline{y}_i A_i \cdot \xi > 0$$

Now, analyzing three cases, we'll show that every factor of the sum is less or equal than 0:

- $i = r$: r is the outgoing index at the iteration p, so $\overline{y}_r < 0$, while at the iteration q, r is the incoming index, so $A_r \cdot \xi > 0$
- $i > r$: by definition of r, all the indexes $i > r$ s.t. $i \in B(v)$ they also belong to all the bases visited between $B(p)$ and $B(q)$. No one of these indexes could be $h(q)$, the outgoing index at the iteration q, so for all of these indexes i we have $i \in B(q)$ and $i \neq h(q)$, so $A_i \cdot \xi = 0$
- $i < r$: we have

  $$r = \min\{j \in B(p) \mid \overline{y}_i < 0\} = \min\{j \in I(\overline{x}) \setminus B(q) \mid A_j \cdot \xi > 0\}$$

  From the first relation we have that $\overline{y}_i \geq 0$ for all $i < r$. From the second relation we obtain $A_i \cdot \xi \leq 0$ for $i < r$, in fact if $i \in B(q)$ we have $A_i \cdot \xi \leq 0$ by construction, if $i \notin B(q)$, r is the minimum index such that $A_j \cdot \xi > 0$.

So we have concluded the proof. □

The whole primal simplex algorithm follows from this proof! Here is the description of the algorithm in pseudocode:

```
procedure Primal_Simplex(A,b,c,B,state){
    for(state=‘‘’’; ; ){
        x = D^(-1) * d;
        y = [y1,y2] = [c * D^(-1),0];
        if(y1>=0) then {state=‘‘optimal’’; break;}
        h = min {i \in B | y1(i)<0};
        z = -D^(-1) * e(B(h));
        if(F*z<=0) then {state=‘‘P illimitate’’; break;}
        l = min {(b(i)-A_i*x)/(A_i*z) | A_i*z>0, i \in N};
        k = min {i \in N | (b(i)-A_i*x)/(A_i*z)=l};
        B =  (B \cup {k})\{h};
    }
}
```

where $D = A_B$, $F = A_N$, $x = \overline{x}$, $y = [y1, y2] = [\overline{y}_B, \overline{y}_N] = \overline{y}$, $d = b_B$, A_i is the i-th row of A. Let's analyze the algorithm: it receives in input the description of the problem (P) and a primal admissible base B, and then it iterates the following steps:

1. it checks the optimality of the primal base solution, in that case the algorithm stops and it provides us the primal solution $\overline{x}$ and the dual solution $\overline{y}$. In the other case it finds a growth direction and go to step 2
2. it computes the maximum step we can do in that direction (it could be 0 in the case we have a degenerate base and it could be $+\infty$ if $A_N \cdot z \leq 0$, in this case the primal is unlimited and the dual is empty)
3. it updates the base with h and k selected with the Blend anti-cycle rule and go to step 1

Thanks to the theorem (12) we know that this algorithm ends in finite steps, solving both the problem (P) and (D).

Let's see now the dual simplex algorithm: it is simply the primal simplex algorithm applied to the problem (D), but with some changes. We write the procedure in pseudocode and then we'll analyze it:

```
procedure Dual_Simplex(A,b,c,B,state) {
    for(state=‘‘’’; ; ) {
```

```
            x = D^(-1) * d;
            y = [y1,y2] = [c * D^(-1),0];
            if(F*x<=f) then {state=‘‘optimal’’; break;}
            k = min {i \in N | A_i*x>b(i)};
            eta = A_k*D^(-1);
            if(eta<=0) then {state=‘‘Empty primal’’; break;}
            Theta = min {y1(i)/eta(i) | eta(i)>0, i \in B};
            h = min {i \in B | y1(i)/eta(i)=theta };
            B = (B \cup {k})\{h};
        }
    }
```

with the same notation used for the primal simplex algorithm, and more $f = b_N$. Let's analyze this algorithm: it receives in input the description of the problems and a dual admissible base and then it iterates the following steps:

1. it checks if the primal base solution is admissible, in that case the algorithm ends. In the other case it calculates $\eta = A_k \cdot A_b^{-1}$, that let us to determine a decreasing direction d for $\overline{y}$, defined as follow

$$d_i = \begin{cases} -\eta_i & if\ i \in B \\ 1 & if\ i = k \\ 0 & otherwise \end{cases}$$

   Let's see that it is a decreasing direction: we define $y(\theta) = \overline{y} + \theta d$, with $\theta >= 0$, so for all $\theta > 0$ we have

$$\begin{aligned} y(\theta) \cdot b &= (\overline{y}_B - \theta\eta) \cdot b_B + \theta b_k = \overline{y}_B \cdot b_B + \theta(b_k - A_k \cdot A_B^{-1} \cdot b_B) \\ &= \overline{y} \cdot b + \theta(b_k - A_k \cdot \overline{x}) < \overline{y} \cdot b \end{aligned}$$

   Moreover we have, with an easy proof, that $y(\theta) \cdot A = c$

2. to ensure that it is an admissible direction, we need to verify $y(\theta) \geq 0$ for an appropriate step θ. Obviously only the indexes in B create a problem: fixing and index $i \in B$, if $\eta_i \leq 0$ we have $y(\theta)_i \geq 0$; if $\eta_I > 0$, $y(\theta)_i \geq 0$ if and only if $\theta \leq \frac{\overline{y}_i}{\eta_i}$. So, the maximum step we can do in the direction d is

$$\Theta = \min\left\{ \frac{\overline{y}_i}{\eta_i} \mid i \in B,\ \eta_i > 0 \right\}$$

   If we could do an infinite step, we could say that the dual problem is unllimited and the primal problem is empty

3. now, choosing
$$h = \min\{i \in B \mid \Theta = \frac{\overline{y}_i}{\eta_i}\}$$
we select another base as described in the code, and we can go to the step 1

To show the convergence of this algorithm, we need the following theorem.

Theorem 13. *Let $\overline{y} = [c \cdot A_B^{-1}, 0]$ be a dual admissible base solution. $\overline{y}$ is an optimal solution for (D) if and only if there exists a base B' associated to $\overline{y}$ such that the primal base solution $\overline{x} = A_B^{-1} \cdot b_B$ is an admissible solution.*

Proof. The proof of this theorem is the same as the proof of the theorem (12). □

In both algorithm, we supposed to have, respectively, a primal admissible base and a dual admissible base. It's not easy to determine them, but in both the situation, we can use others linear programming problems for which we know, respectively, a primal admissible base and a dual admissible base, and from those we can find the base to start the algorithm described previously. To be short we won't show these procedure. Regarding the Blend anticycle rule, it assures us the convergence of the algorithms, but it is not efficient: there is a problem (in primal form) on an ipercube in $\mathbb{R}^n$ that need to do 2^n steps using these rules (in fact the algorithm analyze every vertex to solve the problem). But, experimentally the algorithm ends in fewer steps using better rules for the choice of the indexes h and k. Sometimes this strategy is adopted: the vector c is lightly perturbed to assures us that every vertex of the polytope is determined by only one base, so we don't need the Blend anticycle rule to have the convergence of the algorithm.

Let's analyze the problems (1.4) and (1.6): to solve them we will use the function *linprog* on Matlab; it uses an implementation of the simplex algorithm. Remember that the two algorithms described in this section are only a general scheme for the implementations we find in some programming languages.

2.1.1 Equivalence between two problems

We want to show that exists a point that realizes the minimum for both of the problems (1.2) and (1.4). If we show that the polytope formed by the constraints of (1.4) has integer vertex ($\in \mathbb{Z}^{n^2}$) we have proved the thesis, because the minimum is realized in one of the vertex.

First, we look at the problem (1.4) in graph-theoretical terms: in fact it is a matching problem, that could be interpreted with a flow problem on a bipartite graph.

Definition 14. We say that a graph $G = (V, E)$ is a bipartite graph if the vertex set $V(G)$ has a partition into V_1 and V_2 such that the edge of E has one vertex in V_1 and one vertex in V_2.

Definition 15. Let $G = (V, E)$ be a bipartite graph, with $A = (a_{ve})$ we call the vertex-edge incidence matrix defined by

$$a_{ve} = 1 \quad if \quad v \in e \quad and \quad a_{ve} = 0 \quad if \quad v \notin e$$

It's easy to show that the problem (1.4) is equivalent to the following problem

$$\begin{aligned} \min & \sum_{i,j \in S} C_{i,j} p_{i,j} \\ & A \cdot p = 1 \\ & p_{i,j} \geq 0 \quad \forall i, j \in \mathbb{N} \end{aligned} \tag{2.2}$$

in fact, they have the same restrictions. It's important to note that the matrix A is a $0 - 1$ matrix with a column for each edge and a row for each vertex.

Lemma 16. *If A is the vertex-edge incidence matrix of a bipartite graph, then every square sub-matrix of A has determinant 0,1 or -1.*

Proof. We suppose that S is a $k \times k$ sub-matrix of A and we will show that $\det(S) \in \{0, 1, -1\}$. The case $k = 1$ is easy since A is a $0 - 1$ matrix. Now, consider the possible column expansions of $\det(S)$. Since each edge meets two vertices, each column of S has at most two 1's. If some column has no 1's, $\det(S) = 0$, if one column has just a 1, we can expand about that column and proceed by induction. So, now suppose that every column has exactly two 1's. Then the sum of the V_1-class rows of S is equal to $(1, \ldots, 1)$ and similarly for V_2. Then we have a linear dependency between the rows, so $\det(S) = 0$. □

Now, as we have seen in the previous paragraph, we know that every vertex of the polytope described by constraints like

$$A \cdot x \leq b$$

is of the kind $x = x_B = A_B^{-1} \cdot b_B$, where $A \in \mathbb{R}^{m \times n}$, $B \subset \{1, \ldots, m\}$, $|B| = n$, assuming that $m > n$ and $rank(A) = n$. Using this fact, finally we will show the equivalence between (1.2) and (1.4).

Theorem 17. *The vertices of the polytope described by the constraints of the problem (1.4) consist only of 0-1 vectors.*

Proof. We will show that the polytope defined by

$$\begin{aligned} A \cdot x &= 1 \\ x &\geq 0 \end{aligned}$$

has only integer vertices. We rewrite that constraints in the following way

$$\tilde{A} \cdot x \leq b, \quad with \quad \tilde{A} = \begin{pmatrix} A \\ -A \\ -I \end{pmatrix} \quad and \quad b = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} \tag{2.3}$$

Given a vertex x of the polytope described by (2.3), we know that there exists a base $B \subset \{1, \ldots, 2m+n\}$ such that $x = x_B = \tilde{A}_B^{-1}$. Observe that x is a feasible solution, so $\tilde{A} \cdot x \leq b$. Moreover, if the row $\tilde{A}_i = A_i$, $i \leq n$, is a row of $\tilde{A}_B$, the row $\tilde{A}_{m+i} = -A_i$ cannot be a row of $\tilde{A}$, because $\tilde{A}_B$ is a non-singular matrix. So, if a row like $\tilde{A}_{m+i} = -A_i$ is a row of $\tilde{A}_B$, we can replace it by the row $\tilde{A}_i = A_i$, i.e. without loss of generality we can replace B by $B' = B \cup \{i\} \setminus \{m+i\}$.

We want to show that $\tilde{A}_B^{-1}$ is an integer matrix. Observe that

$$\tilde{A}_B = \begin{pmatrix} A^* \\ -I^* \end{pmatrix}$$

where A^* and $-I^*$ are, respectively, submatrix of A and $-I$. Let us define B_A and B_{-I} as, respectively, the base indexes that determine the rows of A and $-I$ (note that $B = B_A \cup B_{-I}$). Now we will solve the systems

$$\tilde{A}_B \cdot x_i = e_i \quad \forall i = 1, \ldots, n$$

where $e_i \in \mathbb{R}^n$ is the vector which has 1 in the i-th element, and 0 in the other elements; then

$$\tilde{A}_B^{-1} = \big(x_1 | \ldots | x_n\big)$$

Using the Cramer's rule, to calculate the j-th element of x_i we have to replace the j-th column of $\tilde{A}_B^{-1}$ with e_i (we call $\tilde{A}_B^{-1}(j)$ this matrix) and then

$$x_i(j) = \frac{\det(\tilde{A}_B^{-1}(j))}{\det \tilde{A}_B^{-1}}$$

Now we calculate the two determinants:

- $\det(\tilde{A}_B^{-1})$: if we use Laplace expansion along the rows of $-I$, in the end we'll obtain the determinant of a submatrix of A (also a 1×1 matrix), and the lemma 16 assures us that this determinant is 0,1 or -1 (in this case we know that $\tilde{A}_B$ is a non-singular matrix, so it has determinant 1 or -1)
- $\tilde{A}_B^{-1}(j)$: we use Laplace expansion along the j-th column, and then, using tha same argues of the previous point, we can conclude that this determinant is $0, 1$ or -1

So $\tilde{A}_B^{-1}$ is an integer matrix, added to the fact the b is an integer vector, we can conclude that $x = x_B = \tilde{A}_B^{-1} \cdot b_B$ is an integer vector.

□

2.2 The relaxation method

The method we'll describe in this section will let us to prove the convergence of the algorithm proposed to solve the problems (1.5) and (1.8). (see [2])
Suppose we have a linear topological space X, and let $\{A_i \mid i \in I\}$ be a family of closed convex sets. We'll assume that $R = \bigcap_{i \in I} A_i$ is not empty. We want to find some point of the intersection of the sets A_i. Let $S \subset X$ be a convex set such that $S \cap R \neq \emptyset$. Suppose we have a function $D : S \times S \to \mathbb{R}$ that satisfies these six properties:

1. $D(x, y) \geq 0$, $D(x, y) = 0$ iff x=y
2. $\forall y \in S$ and $\forall i \in I$, a point $x = P_i(y) \in A_i \cap S$ exists s.t. $D(x, y) = \min_{z \in A_i \cap S} D(z, y)$ This point will be called the D-projection of the point y onto the set A_i
3. $\forall i \in I$ and $\forall y \in S$ the function $G(z) = D(z, y) - D(z, P_i(y))$ is convex over $A_i \cap S$
4. a derivative of the function D(x,x) exists, i.e.

$$\lim_{t \to 0} \frac{[D(y + tz, y)]}{t} = 0$$

$\forall z \in X$ such that $y + tz \in S$ definitely for $t \to 0$

5. $\forall z \in R \cap S$ and $\forall L \geq 0$ the set

$$\Gamma = \{x \in S \mid D(z, x) \leq L\}$$

is a subset of a compact set

6. if $D(x_n, y_n) \to 0$ and $y_n \to y_\infty \in \overline{S}$ and the set $\{x_n \mid n \in \mathbb{N}\}$ is contained in a compact set, we have $x_n \to y_\infty$

Now, we present three algorithms to find a common point of the sets A_i under other hypothesis :

1. suppose $I = \{1, \ldots, m\}$, we select randomly an $x_0 \in S$ and then we define $i_0(x_0) = 1$, $i_1(x_1) = 2, \ldots, i_{m-1}(x_{m-1}) = m$, $i_m(x_m) = 1$ and so on, and then we choose x_{n+1} as the D-projection of x_n on the set $A_{i_n(x_n)}$

2. suppose $I = \{1, \ldots, m\}$, we select randomly an $x_0 \in S$ and then we follow this strategy: choose randomly, for each block of m steps, the order in which project. So there are $m!$ ways to choose the order

3. suppose $\forall y \in S$ there exists

$$\max_{i \in I} \min_{x \in A_i} D(x, y) \tag{2.4}$$

Now, we choose randomly an $x_0 \in S$, then for $i_n(x_n)$ we'll choose the index which realizes (2.4) and we select the sequence x_n as in the others algorithms

To prove the convergence of these algorithms, we need some intermediate results.

Definition 18. The sequence (x_n) we define in the algorithms is called relaxation sequence, and the sequence $i_n(x_n)$ is called the control of the relaxation.

Lemma 19. *Let $z \in A_i \cap S$, then for any $y \in S$ the inequality $D(P_i(y), y) \leq D(z, y) - D(z, P_i(y))$ is valid.*

Proof. According to the condition 3, for all $\lambda \in [0, 1]$ we have

$$\begin{aligned} &G(\lambda z + (1-\lambda)P_i(y)) \\ &\quad = D(\lambda z + (1-\lambda)P_i(y), y) - D(\lambda z + (1-\lambda)P_i(y), P_i(y)) \\ &\quad \leq \lambda(D(z, y) - D(z, P_i(y))) + (1-\lambda)(D(P_i(y), y) - D(P_i(y), P_i(y))) \end{aligned}$$

$$=\lambda(D(z,y)-D(z,P_i(y)))+(1-\lambda)D(P_i(y),y)$$

When $\lambda > 0$ we obtain

$$\begin{aligned} &D(z,y)-D(z,P_i(y))-D(P_i(y),y)\geq \\ &\frac{D(\lambda z+(1-\lambda)P_i(y),y)}{\lambda}-\frac{D(\lambda z+(1-\lambda)P_i(y),P_i(y))}{\lambda} \end{aligned} \tag{2.5}$$

Since $\lambda z+(1-\lambda)P_i(y)\in A_i\cap S$, the first term on the right hand side of (2.5) is non-negative, thanks to condition 2, nnd the second term tends to zero when $\lambda\to 0$, thanks to condition 4. Hence $D(P_i(y),y)\leq D(z,y)-D(z,P_i(y))$. □

Proposition 20. *For any relaxation control we have the following:*

1. *the set of elements of the relaxation sequence $\{x_n \mid n\in\mathbb{N}\}$ is contained in a compact set*
2. *for any $z\in R$, there exists*

$$\lim_{n\to+\infty} D(z,x_n)$$

3. *$D(x_{n+1},x_n)\to 0$ for $n\to+\infty$*

Proof. We take $z\in R\cap S$. According to lemma 2.5, we have

$$D(x_{n+1},x_n)\leq D(z,x_n)-D(z,x_{n+1}) \tag{2.6}$$

Since $D(x_{n+1},x_n)\geq 0$, we have $D(z,x_n)\geq D(z,x_{n+1})$. So the limit $\lim D(z,x_n)$ exists, and consequently we have $D(x_{n+1},x_n)\to 0$ thanks to (2.6).
Since $D(z,x_1)\geq 0$, thanks to (2.6) we also have $\{x_n\}\subset\{x\in S \mid D(z,x)\leq D(z,x_0)\}$, which is a compact set according to condition (5). □

Now we are ready to prove some convergence results about the three algorithms proposed in this section. And successively we'll show that the operations showed in the algorithms in section 1 are a D-projection of a particular function D.

Theorem 21. *Suppose we have a relaxation sequence $\{x_n\}$ given by the algorithm 1. Then any limiting point x^* of the relaxation sequence is a common point of the sets A_i.*

Proof. Let x^* be a limiting point of the sequence x_n and $x_{n_k} \to x^*$. We separate out form the sequence $\{x_{n_k}\}$ a subsequence (wlog we don't rename this subsequence) which is contained in one of the sets A_i, wlog in the set A_1. We have that the sequence $\{x_{n_k+i-1}\} \subset A_i \quad \forall i = 2, \dots, n$. We can assume that those sequences are convergent, because $\{x_{n_k+i-1}$ is contained in a compact set for each $i = 1, \dots, n$, so we can separate a convergnece subsequence. Let

$$\begin{aligned} &x_{n_k} \to x^* = x_1^* \\ &x_{n_k+1} \to x_2^* \\ &\dots \\ &x_{n_k+m-1} \to x_m^* \end{aligned}$$

Since the sets A_i are closed, we have $x_i^* \in A_i \quad \forall i = 1, \dots, n$. According to proposition (20), $D(x_{n_k+1}, x_{n_k}) \to 0$, thanks to the condition 6 we can say

$$x_2^* = \lim_{k \to +\infty} x_{n_k+1} = \lim_{k \to +\infty} x_{n_k} = x_1^* = x^*$$

so $x^* \in A_2$. Similarly we can say that $x^* \in A_3$, $x^* \in A_4$, and so on. We obtain

$$x^* \in \bigcap_{i \in I} A_i$$

□

Theorem 22. *Suppose we have a relaxation sequence $\{x_n\}$ given by the algorithm 2. Then any limiting point x^* of the relaxation sequence is a common point of the sets A_i almost surely.*

Proof. This proof is similarly to the proof of the theorem 21. Wlog, we have a subsequence $\{x_{n_k}\} \subset A_1$ such that $x_{n_k} \to x^*$. Thanks to Borel paradox, we can say that almost surely there are infinite indexes $k \in \mathbb{N}$ such that $x_{n_{k_j}+1} \in A_2$. Using the same argumentation, we can find a subsequence $x_{n_{k_j}}$ such that

$$\begin{aligned} &x_{n_{k_j}} \to x^* = x_1^* \\ &x_{n_{k_j}+1} \to x_2^* \\ &\dots \\ &x_{n_{k_j}+m-1} \to x_m^* \end{aligned}$$

Now we can conclude as we did in the previous theorem.

□

Theorem 23. *Suppose we have a relaxation sequence $\{x_n\}$ given by the algorithm 3. Then any limiting point x^* of the relaxation sequence is a common point of the sets A_i.*

Proof. Let $x_{n_k} \to x^*$. For each $i \in I$, we have

$$D(\pi_i(x_{n_k}), x_{n_k}) \leq \max_{j \in I} D(\pi_j(x_{n_k}), x_{n_k}) = D(x_{n_k+1}, x_{n_k}) \to 0$$

according to proposition 20. Therefore $D(\pi_j(x_{n_k}), x_{n_k}) \to 0$. The lemma 2.5 assures us that for each $z \in A_i \cap S$ we have

$$D(z, \pi(x_{n_k})) \leq D(z, x_{n_k}) \leq D(z, x_0)$$

So, according to condition 5 the set $\{\pi_i(x_{n_k}) \mid k \in \mathbb{N}\}$ is contained in a compact set, which, together the condition 6, gives $\pi_i(x_{n_k}) \to x^* \ \forall i \in I$. So

$$x^* \in \bigcap_{i \in I} A_i$$

□

The next fact gives us a result of uniqueness of the limiting point for a relaxation sequence.

Proposition 24. *If the function $D(x, y)$ is defined also when $x \in \overline{S} = clos(S)$, and if $y_n \to y^* \in S$ then $D(y^*, y_n) \to 0$, then the sequence $\{x_n\}$ has an unique limiting point.*

Proof. Suppose $x_{n_k} \to x^* \in R$ and $x_{n_l} \to x^{**} \in R$, then, thanks to proposition 20, we know that there exists $\lim D(x^*, x_n)$. Hence we have

$$0 = \lim_{k \to +\infty} D(x^*, x_{n_k}) = \lim_{n \to +\infty} D(x^*, x_n) = \lim_{l \to +\infty} D(x^*, x_{n_l})$$

and thanks to condition 6, it follows that $x^* = x^{**}$. □

Now, we start to apply these results to prove the convergence of the algorithm proposed in the first chapter to solve the problems (1.5) and (1.8).

Definition 25. Let $f : S \times S \to \mathbb{R}$ be a strictly convex differentiable function. We define the Bregman divergence as the function

$$D(x, y) = f(x) - f(y) - < \nabla f(y), x - y >$$

Remark 26. We want to use this definition using as $f(x)$ the cost function of the problems (1.5) and (1.8). Respectively we obtain that the Bregman divergences are:

$$D_2(p,q) = \sum_{i,j} q_{i,j} - p_{i,j} + p_{i,j}(\log(p_{i,j}) - \log(q_{i,j})) \tag{2.7}$$

$$D_3(p,q) = \sum_{i,j,k} q_{i,j,k} - p_{i,j,k} + p_{i,j,k}(\log(p_{i,j,k}) - \log(q_{i,j,k})) \tag{2.8}$$

We observe that, if p and q are probabilty distributions, those Bregman divergences coincide with the Kullback-Leibler divergence.

Definition 27. For the problem (1.5) we define

$$A_1 = \{p \in \mathbb{R}^{n\times n} \mid \sum_j p_{i,j} = \frac{1}{n}\ \forall i = 1,\dots,n,\ \ p \geq 0\} \tag{2.9}$$

$$A_2 = \{p \in \mathbb{R}^{n\times n} \mid \sum_i p_{i,j} = \frac{1}{n}\ \forall j = 1,\dots,n,\ \ p \geq 0\} \tag{2.10}$$

$$S^2 = \{p \in \mathbb{R}^{n\times n} \mid p > 0\} \tag{2.11}$$

The function D_2 is defined over S^2, and it is also defined if $p \in \overline{S^2}$ (the condition of the proposition 24).

Definition 28. For the problem (1.8) we define

$$A_1 = \{p \in \mathbb{R}^{n\times n\times n} \mid \sum_{j,j} p_{i,j,k} = \frac{1}{n}\ \forall i = 1,\dots,n,\ \ p \geq 0\} \tag{2.12}$$

$$A_2 = \{p \in \mathbb{R}^{n\times n\times n} \mid \sum_{i,k} p_{i,j,k} = \frac{1}{n}\ \forall j = 1,\dots,n,\ \ p \geq 0\} \tag{2.13}$$

$$A_3 = \{p \in \mathbb{R}^{n\times n\times n} \mid \sum_{i,j} p_{i,j,k} = \frac{1}{n}\ \forall k = 1,\dots,n,\ \ p \geq 0\} \tag{2.14}$$

$$S^3 = \{p \in \mathbb{R}^{n\times n\times n} \mid p > 0\} \tag{2.15}$$

The function D_3 is defined over S^3, and it is also defined if $p \in \overline{S^3}$ (the condition of the proposition 24).

Theorem 29. *The functions D_2 and D_3 satisfy the conditions (1),..., (6).*

Proof. We will show the proof only for the function D_2, for the function D_3 the proof is the same. Let's verify the six conditions:

1. the cost function $f(p)$ of the problem (1.5) is strictly convex, the condition $D(p,q) = f(p) - f(q) - < \nabla f(q), p - q > \geq 0$ is equivalent to the strictly convexity of the function $f(p)$, in fact it is equivalent to say that the hyperplane tangent to the graphic of the function f in the point q lies below the graphic of the function and it touches the graphic in the point q, i.e. if and only if $p = q$

2. A_i is a closed set, so $A_i \cap S$ is a closed set in the topology of S

3. the function

$$G(z) = - f(y) + f(P_i(y)) - < \nabla f(y), y > + < \nabla f(P_i(y)), P_i(y) > - \\ - < \nabla f(y) - \nabla f(P_i(y)), z >$$

   is linear in the variable z, so the condition 3 is satisfied.

4. the following equalities prove that condition 4 is verified

$$\lim_{t \to 0} \frac{D(y + tz, y)}{t} = \lim_{t \to 0} \frac{f(y + tz) - f(y) - < \nabla f(y), tz >}{t} = \\ = \lim_{t \to 0} \frac{f(y + tz) - f(y)}{t} - < \nabla f(y), z >= 0$$

   where the last equality follows from the definition of directional derivative

5. the condition 5 doesn't depend on the function f, it is true in $\mathbb{R}^n$, so it's satisfied in our case

6. Let's verify the condition 6 in our case (we'll show it for the function D_2 described by (2.7), and the same proof can be used for the function D_3).
   Let $D_2(p_n, q_n) \to 0$ and $q_n \to q_\infty = (q_\infty^{1,1}, \ldots, q_\infty^{n,n}) \in \overline{S}$. If $q_\infty^{i,j} = 0$ for some i, j, then $p_n^{i,j} \to 0$, since otherwise $D_2(p_n, q_n)\ to + \infty$. If $q_\infty^{i,j} > 0$ then $p_n^{i,j} \to q_\infty^{i,j}$ due to the continuity of the function $p_{i,j}(log(p_{i,j}) - log(q_{i,j}))$ when $q_{i,j} > 0$

□

We have seen a method to find a common point of some convex closed sets. Now we'll show that if we choose a particular initial point x_0 with this method

we can also minimize a function.
Our problems, i.e. the problems (1.5) and (1.8) have this kind of form

$$\begin{aligned} \min \ & f(x) \\ & A \cdot x = b \\ & x \in \overline{S} \end{aligned} \tag{2.16}$$

where S is a convex set, $f(x)$ is a convex and continous function on $\overline{S}$, $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^n$. The function D is obtained according to the definition 25 using the function to minimize $f(x)$.

Definition 30. We define $Z = \{x \in S \mid \exists u \in \mathbb{R}^m \ s.t. \ \nabla f(x) = u \cdot A\}$. With $\overline{Z}$ we denote the closure of the set Z.

Lemma 31. *Suppose the function $D(x, y)$ is defined also when $x \in \overline{S} = clos(S)$, and suppose that if $y_n \to y^* \in S$ then $D(y^*, y_n) \to 0$, then if $y^* \in R \cap \overline{Z}$, y^* is a solution of the problem (2.16).*

Proof. Since $y^* \in R \cap \overline{Z}$, we have

$$f(y^*) \geq \inf_{x \in R \cap S} f(x)$$

There exists $x^* \in R \cap S$ such that

$$f(y^*) - f(x^*) = a \geq 0$$

We will prove that $a = 0$. We can find a sequence $\{y_n\} \subset Z$ such that $y_n \to y^*$. For every n we can find $u_n \in \mathbb{R}^m$ such that $\nabla f(y_n) = u_n \cdot A$. It follows that $< \nabla f(y_n), v >= 0$ for every v such that $A \cdot v = 0$. So we have

$$< \nabla f(y_n), y^* - x^* >= 0$$

Now we have the following equalities

$$\begin{aligned} a =& f(y^*) - f(x^*) = \\ &=< \nabla f(y_n), y^* - y_n > + D(y^*, y_n) - < \nabla f(y_n), x^* - y_n > - D(x^*, y_n) = \\ &= D(y^*, y_n) - D(x^*, y_n) \end{aligned}$$

so we obtain $a \leq D(y^*, y_n) \to 0$ because we supposed that the hypothesis in proposition 24 are valid. □

We define $\forall i = 1, \dots, m$

$$A_i = \left\{ x \in \mathbb{R}^n \mid \sum_{j=1}^{n} a_{i,j} x_j = b_i \right\}$$

So the function P_i is the D-projection on the set A_i. The following theorem gives us a condition such that the relaxation sequence converges to a solution of the problem (2.16).

Theorem 32. *Suppose we select a relaxation sequence such that $x_n \to x^* \in R$. Suppose the inital point $x_0 \in Z \cap S$, then x^* is a solution of the problem (2.16).*

Proof. Let x_{n+1} be the D-projection onto the set A_i. Then we have

$$\nabla f(x_{n+1}) = \nabla f(x_n) + \lambda A_i$$

$$< A_i, x_{n+1} >= b_i$$

So, by induction, we can say that $x_n \subset Z$, so $x^* \in \overline{Z}$, and thanks to lemma 31 we obtain that x^* is a solution to the problem (2.16). □

In the algorithms proposed in chapter 1, we use the global minimum of the cost function as initial vector x_0: this assures us that $x_0 \in Z$, because $\nabla f(x_0) = (0, \dots, 0) = \overline{0}$, so we can take $u = \overline{0}$ and observe that $\nabla f(x_0) = u \cdot A$.
To see that the relaxation methods is equivalent to those algorithms, we have to show that, respectively for the problems (1.5) and (1.8), the D_2-projection and the D_3-projection are the operations we use in the algorithms.
For simplicity, we'll see this result only for the problem with two marginals, but the proof for the other problem is the same.

Definition 33. Continuing the definition 27, we define

$$A_{1,i} = \{p \in \mathbb{R}^{n \times n} \mid \sum_{j} p_{i,j} = \frac{1}{n}\} \quad \forall\, i = 1, \dots, n \tag{2.17}$$

$$A_{2,j} = \{p \in \mathbb{R}^{n \times n} \mid \sum_{i} p_{i,j} = \frac{1}{n}\} \quad \forall\, j = 1, \dots, n \tag{2.18}$$

Then we define, respectively, $P_{1,i}$ and $P_{2,j}$ the D_2-projection on the spaces $A_{1,i}$ and $A_{2,j}$.

Definition 34. $\forall\, p \in S^2$, $\forall\, h = 1, \dots, n$ we define

$$\hat{p}(1)_{i,j} = \begin{cases} p_{i,j} & if\ i \neq h \\ \frac{p_{i,j}}{np_{i*}} & if\ i = h \end{cases} \qquad where \qquad p_{i*} = \sum_j p_{i,j}$$

$$\hat{p}(2)_{i,j} = \begin{cases} p_{i,j} & if\ j \neq h \\ \frac{p_{i,j}}{np_{*j}} & if\ j = h \end{cases} \qquad where \qquad p_{*j} = \sum_i p_{i,j}$$

Our purpose is to prove that $\forall\, p \in S^2$, $\forall\, i = 1, \dots, n$ we have $\hat{p}(1) = P_{1,h}(p)$ and $\hat{p}(2) = P_{2,h}(p)$.

Theorem 35. *$\forall p \in S^2$, $\forall i = 1, \dots, n$, we have $\hat{p}(1) = P_{1,h}(p)$ and $\hat{p}(2) = P_{2,h}(p)$.*

Proof. We'll prove that $\hat{p}(1) = P_{1,h}(p)$, the proof for the second assertion is the same. We'll use the Lagrange multiplier method: we fix $p \in \mathbb{R}^{n\times n}$ and $i \in \{1, \dots, n\}$, we want to minimize the function

$$D_2(q, p) = \sum_{i,j} p_{i,j} - q_{i,j} + q_{i,j}(\log(q_{i,j}) - \log(p_{i,j}))$$

in the variable $q \in \mathbb{R}^{n\times n}$, under the constraint

$$g(q) = \sum_j q_{i,j} = \frac{1}{n} \tag{2.19}$$

The Lagrange multiplier method says that $\exists \lambda \in \mathbb{R}$ such that

$$\nabla D_2(\_, p) = \lambda \nabla g$$

So, we have $\forall j = 1, \dots, n$

$$\nabla D_2(\_, p)_{i,j} = \log(q_{i,j}) - \log(p_{i,j}) = \lambda \implies q_{i,j} = e^{\lambda} p_{i,j}$$

while, for each $h \neq i$ we have

$$\log(q_{h,j}) - \log(p_{h,j}) = 0 \implies q_{h,j} = p_{h,j}$$

Using the constraint (2.19), we obtain

$$\sum_j q_{i,j} = e^{\lambda} \sum_j p_{i,j} = e^{\lambda} p_{i*} = \frac{1}{n} \implies e^{\lambda} = \frac{1}{np_{i*}}$$

Finally we have $q = P_{1,i}(p) = \hat{p}(1)$. □

With this theorem we have completed the proof of the convergence of the algorithms proposed in chapter 1.

Chapter 3

Numerical experiments

Using *Matlab*, we implemented the algorithms exposed in the previous chapter, the aim of this chapter is to show the results we obtained by numerical experiments (other experiments can be seen in [1]).

We won't study only how to solve the problems we proposed, but we'll study the behavior of the solutions of the problems with a cost function generated randomly in $[0, 1]$.
For each algorithm we'll follow this strategy: after we fix the necessary parameters (for example the $\varepsilon > 0$ in the Sinkhorn algorithm), we'll execute 50 iterations of the algorithm with different cost vector generated randomly in $[0, 1]$, and then we'll make the average of the results obtained (thanks to the law of large numbers this will be an approximation of the results expected value); for example in the Sinkhorn algorithm for the problem with three marginals, we'll make the average of: the cost without the entropic factor, the time necessary to execute the program, the value of the entropic factor, the error in norm two between an iteration and the previous one, the error in norm one between the marginal distributions we have and the marginal distributions we want (the uniform distribution on S).

3.1 Problem with three marginals

3.1.1 The simplex algorithm

We used the simplex algorithm to solve the problem (1.6). We used the Matlab's function *linprog* to implements the simplex algorithm. The results we obtain are showed in the next pages. We used 7 values for n:

10,20,30,40,50,60,70; then we studied the expected value of the minimum. Looking those graphics, our claim is that the expected value of the minimum tends to 0 as $1/n^2$: this fact will be proved in the fourth chapter.

3.1.2 Sinkhorn's algorithm

We see the behavior of the Sinkhorn's algorithm. We executed the program in Matlab with all combinations of these parameters: $n = 50, 90, 130, 170, 210, 250$; 50 is the number of iterations did by the algorithm; $\varepsilon = 10^{-4}, 10^{-5}, \frac{1}{n^2 \log^2(n)}$. As we said at the beginning of the chapter, we executed the program 50 times, to do an average of the results. In the next pages there are some graphics, with the appropriate descriptions, which show the results we obtained.

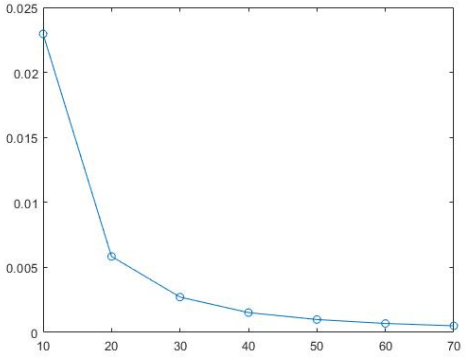
We can observe that if we fix $\varepsilon > 0$, the expected value of the costs don't seem to be asymptotic to $\frac{1}{n^2}$, while if we choose $\varepsilon = \frac{1}{n^2 \log^2(n)}$ we have the same results obtained with the simplex algorithm, and the expected value of the entropic factor tends to 0 too like $o(1/n^2)$. In the fourth chapter we'll see that this choice of ε is supported by theoretical results.

To see that $\mathbb{E}[f(p)]$ can't tend to 0 if we fix $\varepsilon > 0$ (where f is the cost function of the problem (1.8) and p is the minimum point of that function), we need Pinsker's inequality. These results can be seen in the fourth chapter.

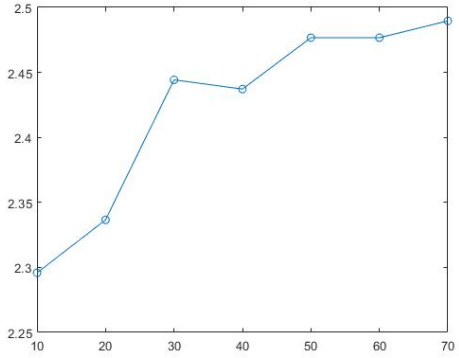
3.1.3 Randomized Sinkhorn's algorithm

We follow the same strategy of the previous algorithm to study the randomized Sinkhorn's algorithm: in particular we used the same parameters, to compare the two algorithms. In the next pages we can see the results produced by the numerical experiments.

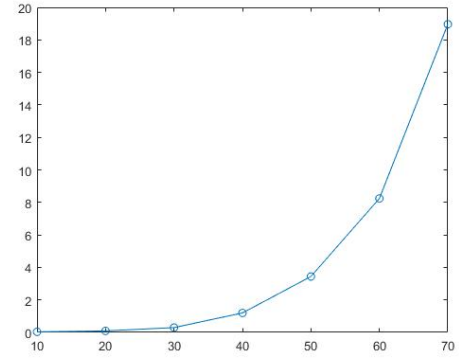
The costs and the value of the entropic factor has the same behavior we observed for the results we obtained using the Sinkhorn algorithm without randomization. The errors (between the marginals at each step and the marginals we want and between an iteration and the privious one) has the same behavior too, but they decrease slowly. So we can conclude that the Sinkhorn algorithm, experimentally, is better than the randomized Sinkhorn algorithm.



(a) *Costs average.*

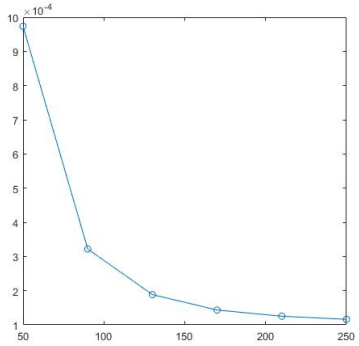


(b) *Costs average multiplied by* n^2.

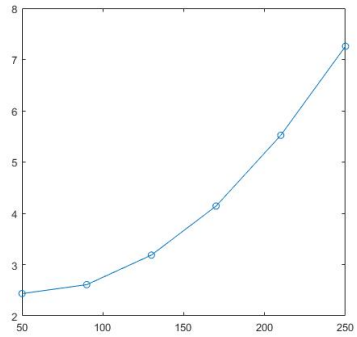


(c) *The time necessary to execute the algorithm.*

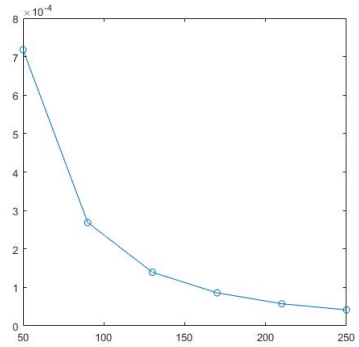
Figure 3.1: (Three marginal problem) The results we obtained using the simplex algorithm for the problem (1.6).



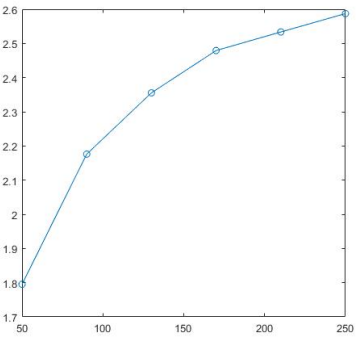
(a) *Costs average with* $\varepsilon = 10^{-4}$



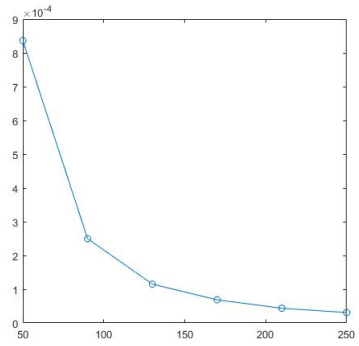
(b) *Costs average multiplied by* n^2 *with* $\varepsilon = 10^{-4}$



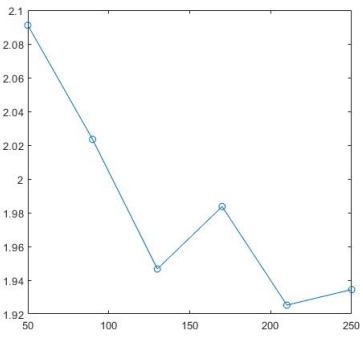
(c) *Costs average with* $\varepsilon = 10^{-5}$



(d) *Costs average multiplied by* n^2 *with* $\varepsilon = 10^{-5}$

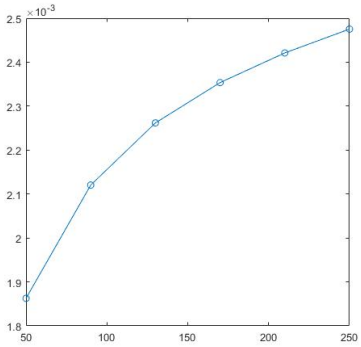


(e) *Costs average with* $\varepsilon = \frac{1}{n^2 \log^2(n)}$

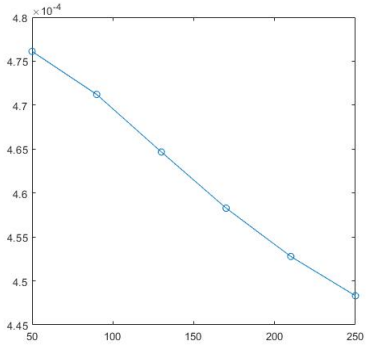


(f) *Costs average multiplied by* n^2 *with* $\varepsilon = \frac{1}{n^2 \log^2(n)}$

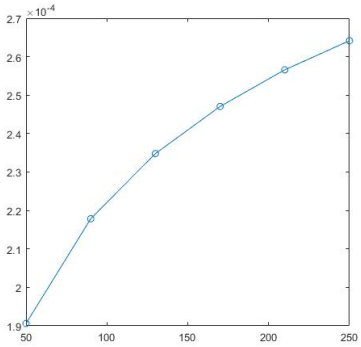
Figure 3.2: (Three marginal problem) The costs averages (without the entropic factor) obtained using the Sinkhorn algorithm.



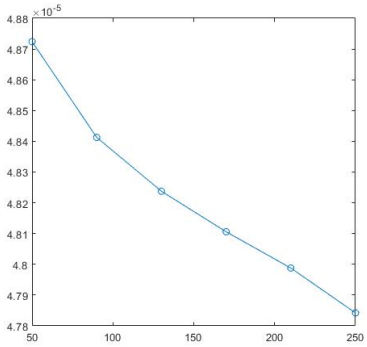
(a) *Entropy average with* $\varepsilon = 10^{-4}$



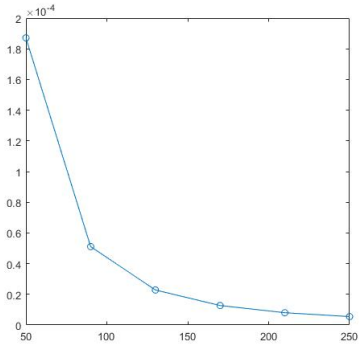
(b) *Entropy average divided by* $\log(n)$ *with* $\varepsilon = 10^{-4}$



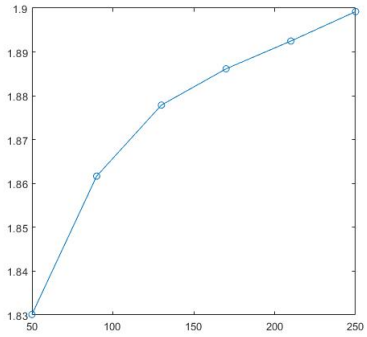
(c) *Entropy average with* $\varepsilon = 10^{-5}$



(d) *Entropy average divided by* $\log(n)$ *with* $\varepsilon = 10^{-5}$

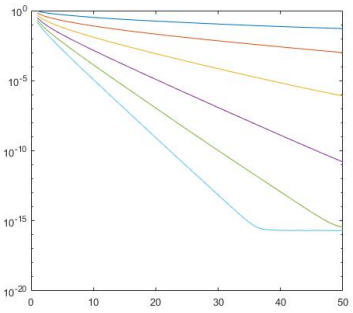


(e) *Entropy average with* $\varepsilon = \frac{1}{n^2 \log^2(n)}$

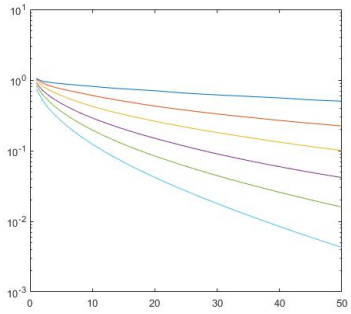


(f) *Entropy average multiplied by* $n^2 \log(n)$ *with* $\varepsilon = \frac{1}{n^2 \log^2(n)}$

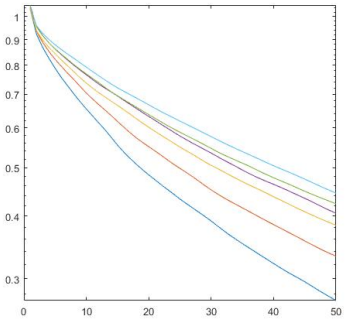
Figure 3.3: (Three marginal problem) The entropy averages obtained using the Sinkhorn algorithm.



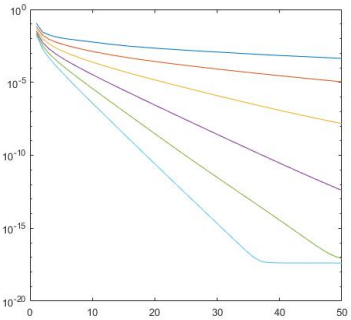
(a) *Error between the marginals at each iteration with* $\varepsilon = 10^{-4}$



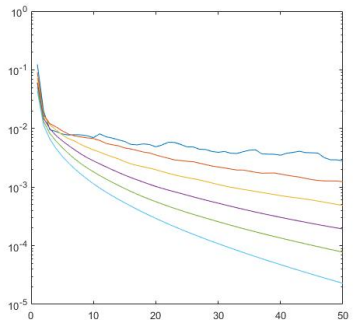
(b) *Error between the marginals at each iteration with* $\varepsilon = 10^{-5}$



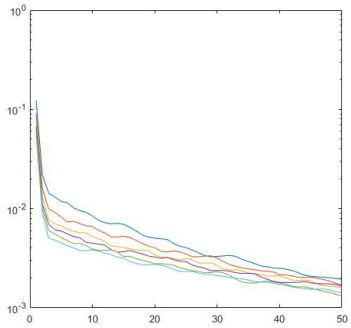
(c) *Error between the marginals at each iteration with* $\varepsilon = \frac{1}{n^2 \log^2(n)}$



(d) *Error between each iteration and the previous one with* $\varepsilon = 10^{-4}$

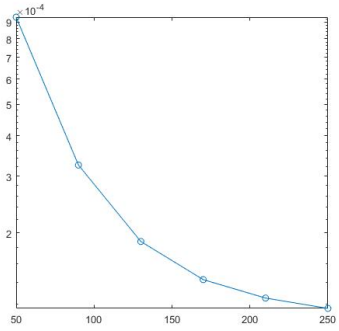


(e) *Error between each iteration and the previous one with* $\varepsilon = 10^{-5}$

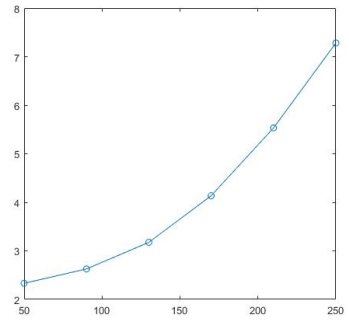


(f) *Error between each iteration and the previous one with* $\varepsilon = \frac{1}{n^2 \log^2(n)}$

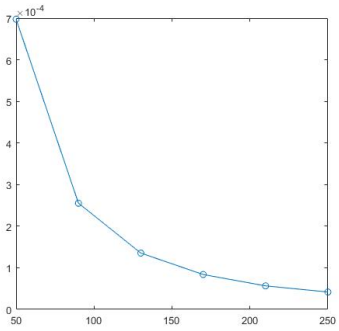
Figure 3.4: (Three marginal problem) The errors in the Sinkhorn algorithm. In each figures, there are 6 graphics, one for each n used: blue→n=50, red→n=90, yellow→n=130, violet→n=170, green→n=210, light blue→n=250.



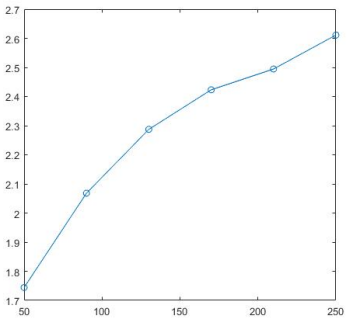
(a) *Costs average with* $\varepsilon = 10^{-4}$



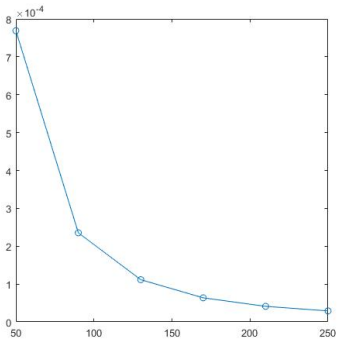
(b) *Costs average multiplied by* n^2 *with* $\varepsilon = 10^{-4}$



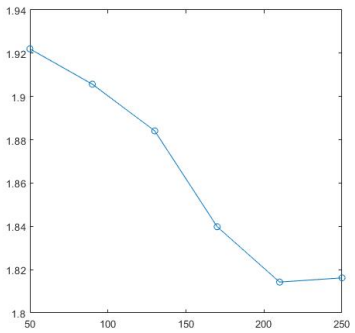
(c) *Costs average with* $\varepsilon = 10^{-5}$



(d) *Costs average multiplied by* n^2 *with* $\varepsilon = 10^{-5}$

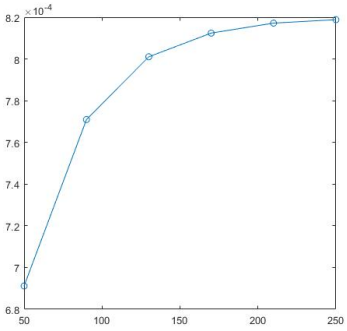


(e) *Costs average with* $\varepsilon = \frac{1}{n^2 \log^2(n)}$

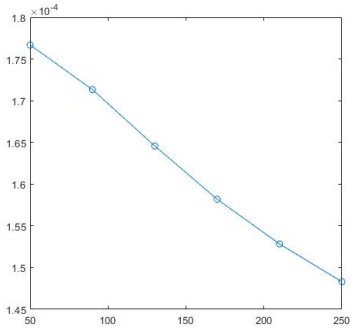


(f) *Costs average multiplied by* n^2 *with* $\varepsilon = \frac{1}{n^2 \log^2(n)}$

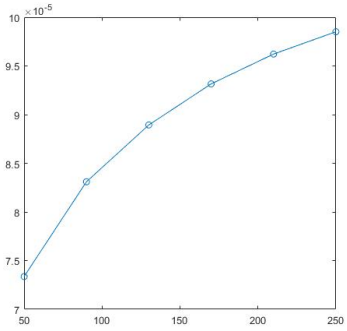
Figure 3.5: (Three marginal problem) The costs averages (without the entropic factor) obtained using the randomized Sinkhorn algorithm.



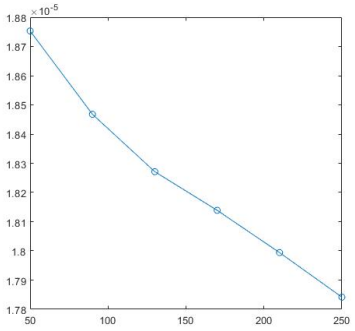
(a) *Entropy average with* $\varepsilon = 10^{-4}$



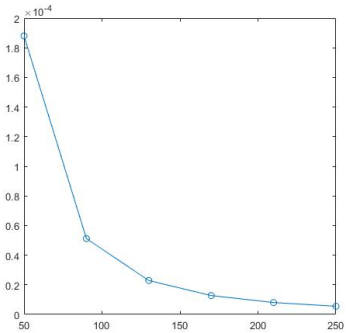
(b) *Entropy average divided by* $\log(n)$ *with* $\varepsilon = 10^{-4}$



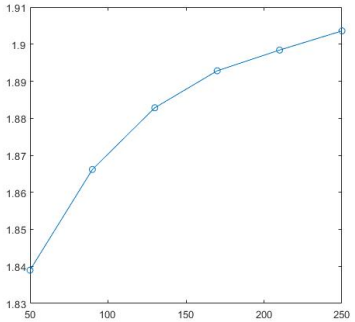
(c) *Entropy average with* $\varepsilon = 10^{-5}$



(d) *Entropy average divided by* $\log(n)$ *with* $\varepsilon = 10^{-5}$

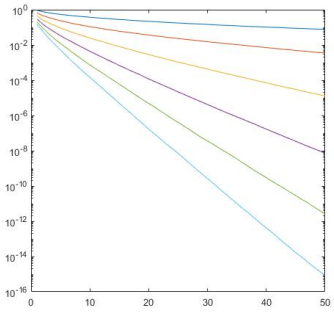


(e) *Entropy average with* $\varepsilon = \frac{1}{n^2 \log(n)}$

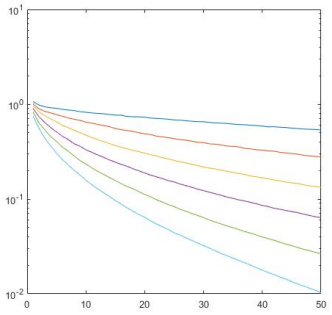


(f) *Entropy average multiplied by* $n^2 \log(n)$ *with* $\varepsilon = \frac{1}{n^2 \log^2(n)}$

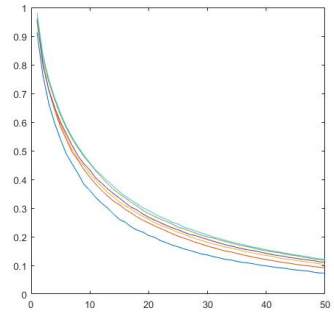
Figure 3.6: (Three marginal problem) The entropy averages obtained using the randomized Sinkhorn algorithm.



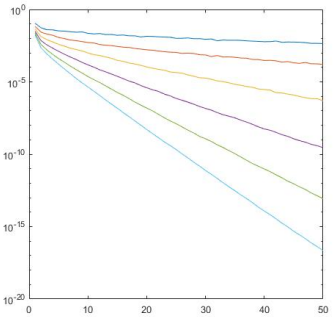
(a) *Error between the marginals at each iteration with* $\varepsilon = 10^{-4}$



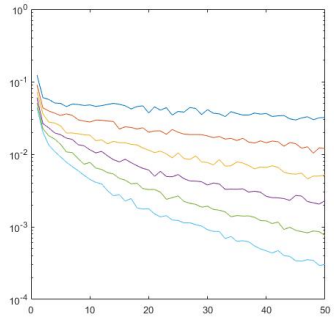
(b) *Error between the marginals at each iteration with* $\varepsilon = 10^{-5}$



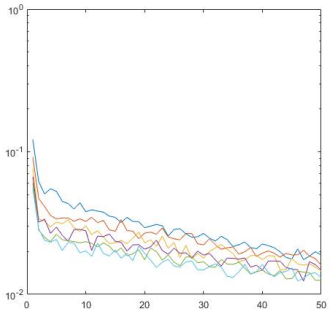
(c) *Error between the marginals at each iteration with* $\varepsilon = \frac{1}{n^2 \log(n)}$



(d) *Error between each iteration and the previous one with* $\varepsilon = 10^{-4}$



(e) *Error between each iteration and the previous one with* $\varepsilon = 10^{-5}$



(f) *Error between each iteration and the previous one with* $\varepsilon = \frac{1}{n^2 \log(n)}$

Figure 3.7: (Three marginal problem) The errors in the randomized Sinkhorn algorithm. In each figures, there are 6 graphics, one for each n used: blue→n=50, red→n=90, yellow→n=130, violet→n=170, green→n=210, light blue→n=250.

3.1.4 Bregman's algorithm

We called *Bregman's algorithm* the third algorithm we proposed for the problem (1.8). We used the same parameters and the same strategy used for the others two algorithms, to compare the algorithms. The results produced by the experimentation are showed in the next pages. The costs and the Kullback-Leibler divergence have the same behavior we observed for the other algorithms. But, the errors decrease slowly than the other algorithms. Remember that in one step of this algorithm we execute only one of the three operations we do in the other algorithms, but to choose this operation we need to calculate all the projections and, moreover, the Bregman divergence between the projections and the initial probability distribution. So we need more time to execute this algorithm than the others.
Our conclusion is that, experimentally, the Sinkhorn algorithm (without randomization) is better than the others two algorithms proposed to solve the problem (1.8).

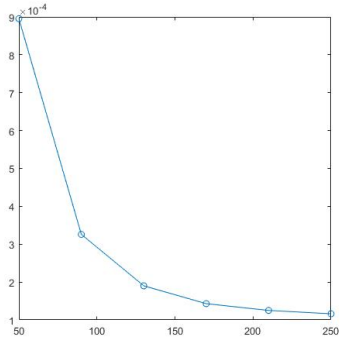
3.2 Problem with two marginals

3.2.1 Linear programming problem

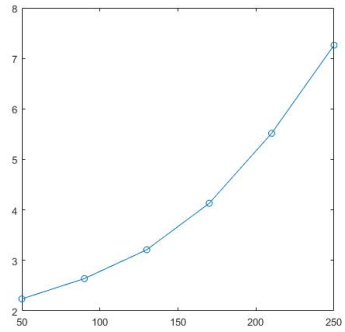
The problem (1.4) is a linear programming problem, so it can be solved using the simplex algorithm. Our program uses the Matlab's function *linprog*: it solves a linear programming problem taking in input the cost vector and the constraints matrix. The results we obtained are showed in the figure 3.11.

3.2.2 Sinkhorn's algorithm

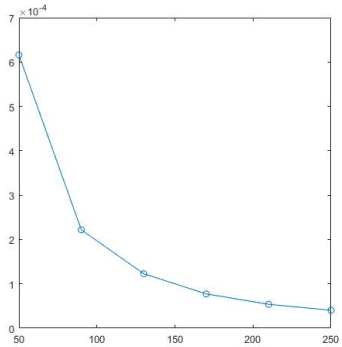
As we said in the first chapter, we know that this algorithm converges exponentially to the minimum point, so we didn't analyze the errors, but only the cost and the entropy factor. We used the following parameters to study the algorithm: $n = 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100$; the constant of the Kullback-Leibler divergence $\varepsilon = 10^{-3}, 10^{-4}, 1/\left(n \log^2(n)\right)$; the tolerance for the stop criterion $\tau = 10^{-7}$. In the next pages we can see the results obtained.



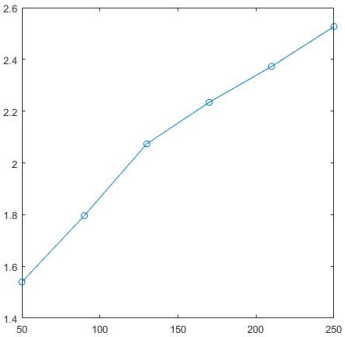
(a) *Costs average with* $\varepsilon = 10^{-4}$



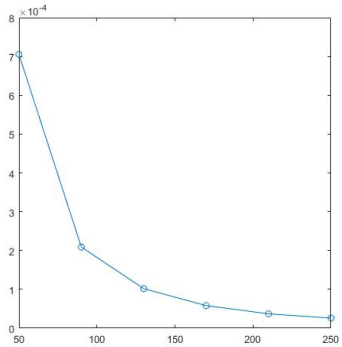
(b) *Costs average multiplied by* n^2 *with* $\varepsilon = 10^{-4}$



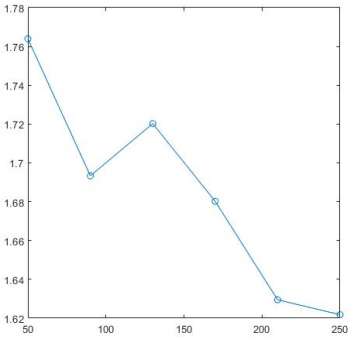
(c) *Costs average with* $\varepsilon = 10^{-5}$



(d) *Costs average multiplied by* n^2 *with* $\varepsilon = 10^{-5}$

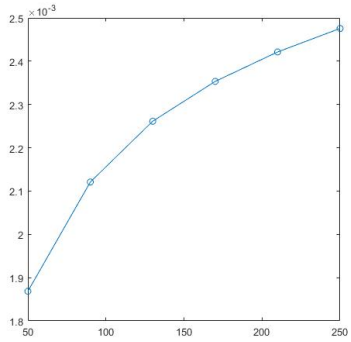


(e) *Costs average with* $\varepsilon = \frac{1}{n^2 \log^2(n)}$

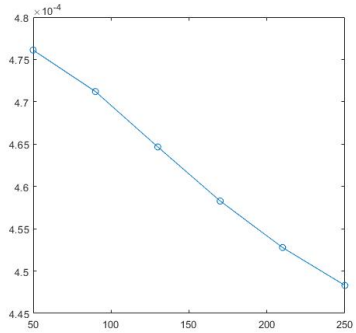


(f) *Costs average multiplied by* n^2 *with* $\varepsilon = \frac{1}{n^2 \log^2(n)}$

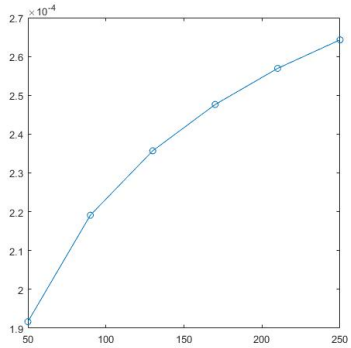
Figure 3.8: (Three marginal problem) The costs averages (without the entropic factor) obtained using the Bregman algorithm.



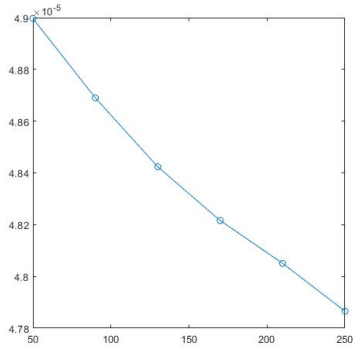
(a) *Entropy average with* $\varepsilon = 10^{-4}$



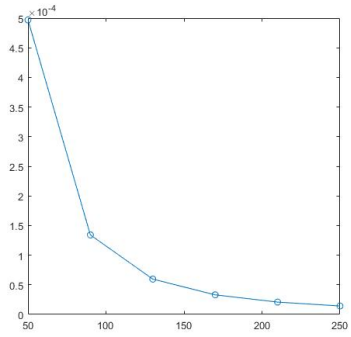
(b) *Entropy average divided by* $\log(n)$ *with* $\varepsilon = 10^{-4}$



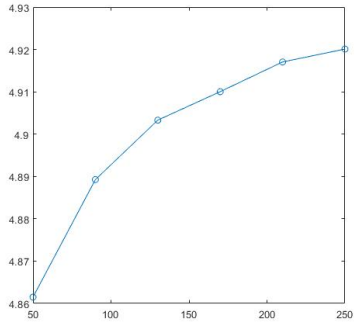
(c) *Entropy average with* $\varepsilon = 10^{-5}$



(d) *Entropy average divided by* $\log(n)$ *with* $\varepsilon = 10^{-5}$

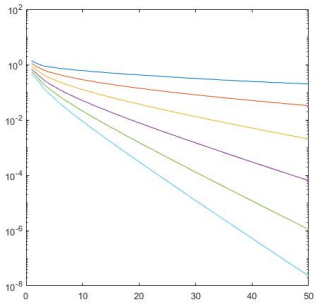


(e) *Entropy average with* $\varepsilon = \frac{1}{n^2 \log^2(n)}$

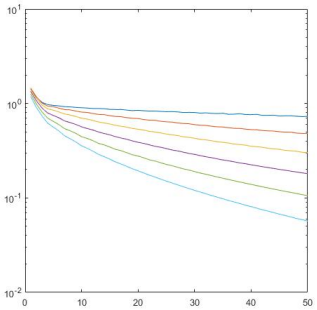


(f) *Entropy average multiplied by* $n^2 \log(n)$ *with* $\varepsilon = \frac{1}{n^2 \log^2(n)}$

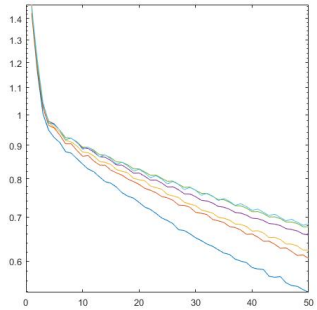
Figure 3.9: (Three marginal problem) The entropy averages obtained using the Bregman algorithm.



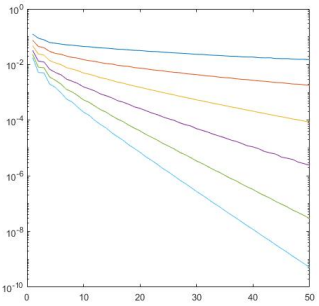
(a) *Error between the marginals at each iteration with* $\varepsilon = 10^{-4}$



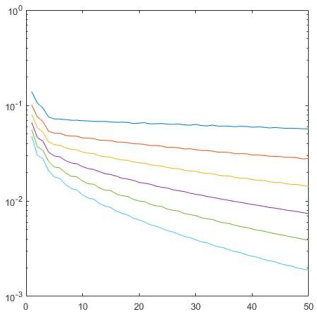
(b) *Error between the marginals at each iteration with* $\varepsilon = 10^{-5}$



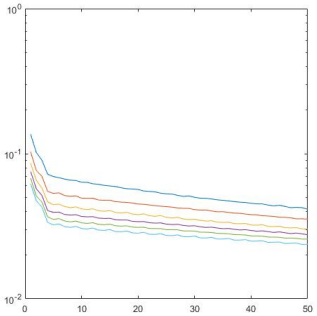
(c) *Error between the marginals at each iteration with* $\varepsilon = \frac{1}{n^2 \log^2(n)}$



(d) *Error between each iteration and the previous one with* $\varepsilon = 10^{-4}$

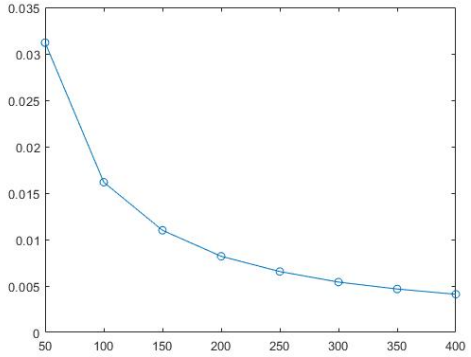


(e) *Error between each iteration and the previous one with* $\varepsilon = 10^{-5}$

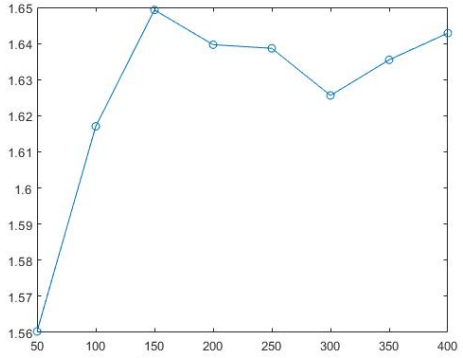


(f) *Error between each iteration and the previous one with* $\varepsilon = \frac{1}{n^2 \log^2(n)}$

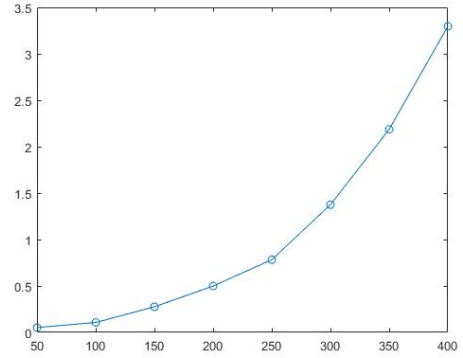
Figure 3.10: (Three marginal problem) The errors in the Bregman algorithm. In each figures, there are 6 graphics, one for each n used: blue→n=50, red→n=90, yellow→n=130, violet→n=170, green→n=210, light blue→n=250.



(a) *Costs average.*

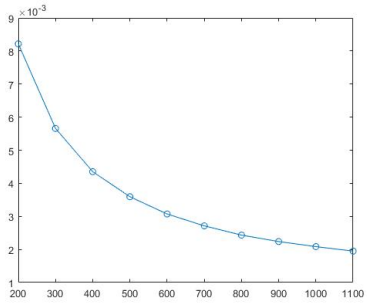


(b) *Costs average multiplied by n.*

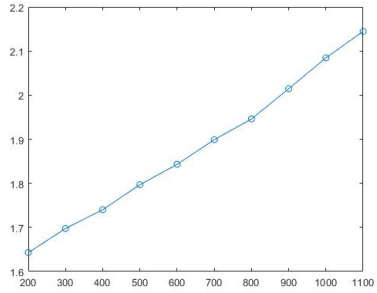


(c) *The time necessary to execute the algorithm.*

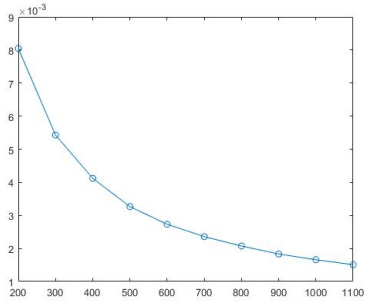
Figure 3.11: (Two marginal problem) The results we obtained using the simplex algorithm to solve the problem (1.4).



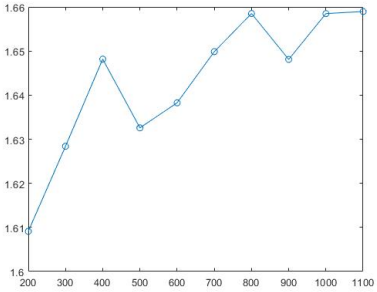
(a) *Costs average with* $\varepsilon = 10^{-3}$



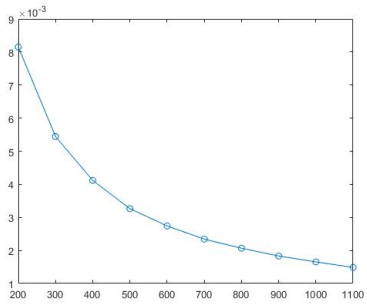
(b) *Costs average multiplied by* n *with* $\varepsilon = 10^{-3}$



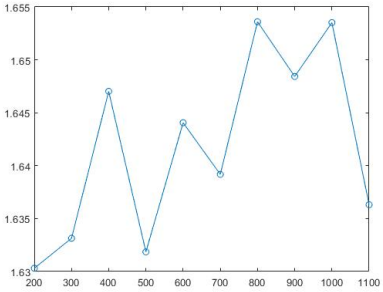
(c) *Costs average with* $\varepsilon = 10^{-4}$



(d) *Costs average multiplied by* n *with* $\varepsilon = 10^{-4}$

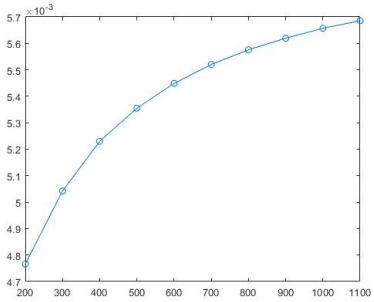


(e) *Costs average with* $\varepsilon = \frac{1}{n \log^2(n)}$

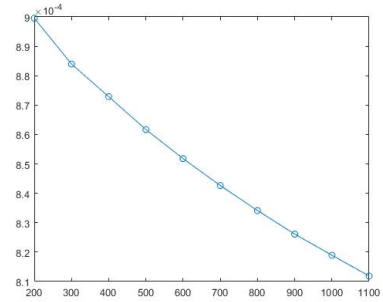


(f) *Costs average multiplied by* n *with* $\varepsilon = \frac{1}{n \log^2(n)}$

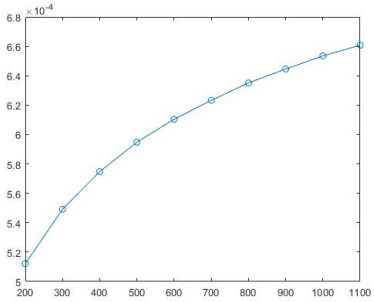
Figure 3.12: (Two marginal problem) The costs averages (without the entropic factor) obtained using the Sinkhorn algorithm with $\tau = 10^{-7}$.



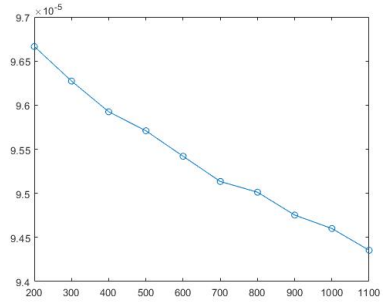
(a) *Entropy average with* $\varepsilon = 10^{-3}$



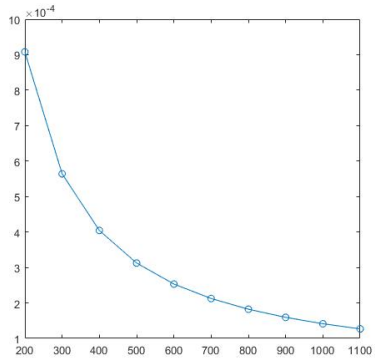
(b) *Entropy average divided by* $\varepsilon \log(n)$ *with* $\varepsilon = 10^{-3}$



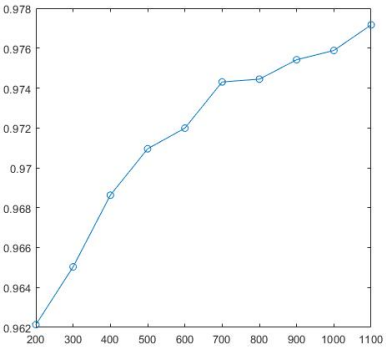
(c) *Entropy average with* $\varepsilon = 10^{-4}$



(d) *Entropy average divided by* $\varepsilon \log(n)$ *with* $\varepsilon = 10^{-4}$



(e) *Entropy average with* $\varepsilon = \frac{1}{n \log^2(n)}$



(f) *Entropy average multiplied by* $n \log(n)$ *with* $\varepsilon = \frac{1}{n \log^2(n)}$

Figure 3.13: (The two marginal problem) The entropy averages obtained using the Sinkhorn algorithm with $\tau = 10^{-7}$.

Chapter 4

Probabilistic results

4.1 The Dyer-Frieze-McDiarmid inequality

In this section we'll see an inequality that let us to say that the expected values of the optimal value z^* of the problems (1.5) and (1.8) are, respectively, less than or equal to $\frac{1}{n}$ and $\frac{1}{n^2}$.

We introduce a kind of problem (it is like a dual problem introduced in first chapter)

$$\begin{aligned} \min\ z = & \sum_{j=1}^{n} c_j x_j \\ & \sum_{j=1}^{n} a_{i,j} x_j = b_i \ \ \forall i = 1, \ldots, m \\ & x_j \geq 0 \ \ \forall j = 1, \ldots, n \end{aligned} \tag{4.1}$$

where $a_{i,j} \in \mathbb{R}$ and $b_i \in \mathbb{R}$ are fixed constants, and the $c_j \in \mathbb{R}$ are non-negative random variables. In our problems, we consider the components of the cost function as independent uniformly distributed random variable on $[0, 1]$, so we have the following

$$\mathbb{E}[c_i | c \geq h] = \mathbb{E}[c] + \frac{1}{2}h \ \ \forall 0 < h < 1$$

where $\mathbb{E}[X|A]$ is the expected value of the random variable X conditioned by the event A. Guided by this observation, we are ready for the Dyer-Frieze-McDiarmid inequality (see [3]).

Theorem 36. *(Dyer-Frieze-McDiarmid inequality)*
Suppose c_j, $1 \leq j \leq n$ are independent non-negative random variables defined

by a density function, $\beta \in (0,1]$ is a constant and $(\hat{x}_j) \in \mathbb{R}^n$ is a feasible solution for the problem (4.1). If $\forall h > 0$ such that $P(c_j \geq h) > 0$ we have

$$\mathbb{E}[c_j | c_j \geq h] \geq \mathbb{E}[c_j] + \beta h$$

then, if we indicate z^* the optimal value of the problem (4.1), we have

$$\beta \mathbb{E}[z^*] \leq \max_{S:|S|=m} \sum_{j \in S} \hat{x}_j \mathbb{E}[c_j]$$

Proof. Let us make some considerations using soe facts we illustrated in the simplex algorithm's section: we can assume A is of full rank m, we have N feasible bases $B(r) \subset \{1, \ldots, n\}$ with $1 \leq r \leq n$, the feasible basis $B(r)$ is optimal if and only if

$$c_j - c_{B(r)} A_{B(r)}^{-1} a_j \geq 0 \quad \forall j \in N(r) = B(r)^c \tag{4.2}$$

where a_j is the j-th column of the matrix A. This optimal criterion directly descends from theorem (13). Let E_r denotes the event that (4.2) is valid for the basis $B(r)$: the union of the sets E_r has probability one since there exists an optimal basic solution.

Now we would like that only one of the E_r occurs. It's easy to prove that almost surely the problem has only one solution: if the problem has more than one solution, then it has at least two different solutions which are two vertexes, let x, $\tilde{x}$ be two different vertexes that are solutions of the problem, then necessarily the cost function c is orthogonal to the difference $x - \tilde{x}$, and this event, choosing each pair of vertexes, has probability 0. But we can't conclude that almost surely only one E_r occurs, because the solution could be determined by more different basis. To avoid this problem, we can observe that if two basis determine the solution, a base component of the solution is 0. We can choose a matrix $\tilde{A}$ arbitrarily close (in a fixed norm on $\mathbb{R}^{m \times n}$) that has an unique basic solution (because, fixing a component j, the event $(cA_B^{-1})_j = 0$ in the space $\mathbb{R}^{m \times n}$ has Lebesgue measure 0, for each B such that A_B is a non-singular matrix, and these events are finite): now we show the proof of the inequality for the problem (4.1) where we replace A with $\tilde{A}$, and then we conclude using the dominated convergence theorem. Observe that, we need to consider another feasible solution $\hat{x}'$, but we can consider it 'near' to the original $\hat{x}$ (i.e. if $\tilde{A} \to A$ then $\hat{x}' \to \hat{x}$).

Now, applying the optimal criterion and the hypothesis, we obtain

$$\begin{aligned} \mathbb{E}[c_j | E_r \ \textit{and} \ c_{B(r)}] =& \mathbb{E}[c_j | c_j \geq c_{B(r)} \tilde{A}_{B(r)}^{-1} \tilde{a}_j \ \textit{and} \ c_{B(r)}] \\ \geq& \mathbb{E}[c_j] + \beta c_{B(r)} \tilde{A}_{B(r)}^{-1} \tilde{a}_j \end{aligned} \tag{4.3}$$

If we multiply for $\hat{x}'_j$ and we sum over j we obtain

$$\mathbb{E}\left[\sum_{j=1}^{n} c_j \hat{x}'_j | E_r \ \ and \ \ c_{B(r)}\right] = \sum_{j\in B(r)} c_j \hat{x}'_j + \sum_{j\in N(r)} \mathbb{E}[c_j | E_r \ \ and \ \ c_{B(r)}]\hat{x}'_j$$
$$\geq \sum_{j\in B(r)} c_j \hat{x}'_j + \sum_{j\in N(r)} \left(\mathbb{E}[c_j] + \beta c_{B(r)} \tilde{A}^{-1}_{B(r)} \tilde{a}_j\right)_j \hat{x}'_j$$

Now, we need to develop an expression that relates z^* to the feasible solution $(\hat{x}'_j)$. When $B(r)$ is an optimal basis, we have $\tilde{z}^* = c_{B(r)} \tilde{a}^{-1}_{B(r)} b$ is the optimal value of the problem (4.1), because, as we have seen in the second chapter, we have that an optimal basic solution has the form $[x_B, x_N] = [\tilde{A}_B^{-1} b, 0]$. Remembering that

$$\sum_{j=1}^{n} \tilde{a}_j \hat{x}'_j = b$$

we have

$$\mathbb{E}\left[\sum_{j=1}^{n} c_j \hat{x}'_j | E_r \ \ and \ \ c_{B(r)}\right]$$
$$\geq \sum_{j\in B(r)} c_j \hat{x}'_j + \sum_{j\in N(r)} \left(\mathbb{E}[c_j]\hat{x}'_j\right) + \beta c_{B(r)} \tilde{A}^{-1}_{B(r)} \left(b - \sum_{j\in B(r)} \tilde{a}_j \hat{x}'_j\right)$$
$$\geq \sum_{j\in B(r)} \left(c_j - \beta c_{B(r)} A^{-1}_{B(r)} \tilde{a}_j\right) \hat{x}'_j + \sum_{j\in N(r)} \left(\mathbb{E}[c_j]\hat{x}'_j\right) + \beta \mathbb{E}[\tilde{z}^* \mid E_r \ \ and \ \ c_{B(r)}]$$

Let $p_r = P(E_r)$, thanks to properties of the conditional expectation, we have the same inequality without conditioning on $c_{B(r)}$, so we obtain

$$\sum_r p_r \mathbb{E}\left[\sum_{j=1}^{n} c_j \hat{x}'_j | E_r\right] \geq \sum_r p_r \sum_{j\in N(r)} \mathbb{E}[c_j]\hat{x}'_j + \beta \sum_r p_r \mathbb{E}\left[\tilde{z}^* | E_r\right]$$

Using $\mathbb{E}[X] = \mathbb{E}\left[\mathbb{E}\left[X|A\right]\right]$, we have

$$\sum_{j=1}^{n} \mathbb{E}[c_j] v \hat{x}'_j \geq \sum_r p_r \sum_{j\in N(r)} \mathbb{E}[c_j]\hat{x}'_j + \beta \mathbb{E}[\tilde{z}^*]$$

And finally

$$\beta \mathbb{E}[\tilde{z}^*] \leq \sum_{j=1}^{n} \mathbb{E}[c_j]\hat{x}'_j - \sum_r p_r \sum_{j\in N(r)} \mathbb{E}[c_j]\hat{x}'_j$$

$$= \sum_r p_r \left\{ \sum_{j=1}^{n} \mathbb{E}[c_j]\hat{x}'_j - \sum_{j \in N(r)} \mathbb{E}[c_j]\hat{x}'_j \right\}$$
$$= \sum_r p_r \sum_{j \in N(r)} \mathbb{E}[c_j]\hat{x}'_j \leq \max_{S:|S|=m} \sum_{j \in S} \hat{x}'_j \mathbb{E}[c_j]$$

Now, if we consider $\tilde{A} \to A$, the feasible solution $\hat{x}' \to \hat{x}$ and thanks to the dominated convergence theorem, dominating $\tilde{z}^*$ with the constant function $g \equiv n$, we obtain

$$\beta \mathbb{E}[\tilde{z}^*] \to \beta \mathbb{E}[z^*]$$

and

$$\max_{S:|S|=m} \sum_{j \in S} \hat{x}'_j \mathbb{E}[c_j] \to \max_{S:|S|=m} \sum_{j \in S} \hat{x}_j \mathbb{E}[c_j]$$

so the inequality holds true.

□

Thanks to this inequality, we are ready to prove a conjecture we did in the previous chapter thanks to the numerical experiments.

Corollary 37. *Considering the problem (1.8), if the components of the cost are chosen randomly in* $[0, 1]$*, then, indicating as* z^* *the optimal value, we have*

$$\mathbb{E}[z^*] = \Theta\left(\frac{1}{n^2}\right)$$

Proof. We observe that $\forall 0 < h \leq 1$

$$\mathbb{E}[c_{i,j,k} | c_{i,j,k} \geq h] = \frac{1}{2} + \frac{1}{2}h = \mathbb{E}[c_{i,j,k}] + \frac{1}{2}h$$

so the hypothesis of the Dyer-Frieze-McDiarmid inequality are verified. Let $\hat{p}$ be the uniform probability distribution on $S \times S \times S$, i.e. $(\hat{p})_{i,j,k} = 1/n^3$, then $\hat{p}$ is a feasible solution for the problem (1.8), so, thanks to the Dyer-Frieze-McDiarmid inequality we have

$$\frac{1}{2}\mathbb{E}[z^*] \leq \max_{T:|T|=3n} \sum_{(i,j,k) \in T} \frac{1}{2}\hat{p}_{i,j,k}$$

$$\implies \mathbb{E}[z^*] \leq \max_{T:|T|=3n} \sum_{(i,j,k) \in T} \hat{p}_{i,j,k} = \frac{3}{n^2}$$

The other inequality follows from these passages

$$\begin{aligned}\mathbb{E}[z^*] =& \mathbb{E}\left[\sum_{i,j,k} c_{i,j,k}\pi^*_{i,j,k}\right] = \mathbb{E}\left[\sum_i \sum_{j,k} c_{i,j,k}\pi^*_{i,j,k}\right] \\ \geq& \mathbb{E}\left[\sum_i \left(\sum_{j,k}\left(\min_{j,k} c_{i,j,k}\right)\pi^*_{i,j,k}\right)\right] \\ =& \sum_i \mathbb{E}\left[\frac{1}{n}\left(\min_{j,k} c_{i,j,k}\right)\right] \\ =& \frac{1}{n^2+1}\end{aligned}$$

□

Similarly, we can prove the following corollary.

Corollary 38. *Considering the problem (1.5), if the components of the cost are chosen randomly in* $[0,1]$*, then, indicating as* z^* *the optimal value, we have*

$$\mathbb{E}[z^*] = \Theta\left(\frac{1}{n}\right)$$

4.2 The right choice for $\varepsilon > 0$

Let's see why we can't fix $\varepsilon > 0$ $\forall n \in \mathbb{N}$ in the problem (1.5) and 1.8).

Definition 39. Let P,Q be two probability distributions on a finite set Ω, then we define the total variation distance as

$$\|P - Q\|_1 = \sum_{x\in\Omega} |p(x) - q(x)|$$

Theorem 40. *(Pinsker's inequality) With the same setting of the previous definition, we have the following inequality*

$$\sqrt{2KL(P||Q)} \geq \|P - Q\|_1$$

Proof. If there exists $x \in \Omega$ such that $p(x) > 0$ and $q(x) = 0$, then $KL(P||Q) = +\infty$, so the inequality is obvious. So, assume that

$$\sup_{x\in\Omega} \frac{p(x)}{q(x)} < +\infty \tag{4.4}$$

It's easy prove the following inequality

$$(1+t)\log(1+t) - t \geq \frac{3}{2} \cdot \frac{t^2}{3+t} \tag{4.5}$$

Now we define $r(x) = \frac{p(x)}{q(x)} - 1 \ \forall x \in \Omega$, and with easy algebraic passages we can observe

$$\mathbb{E}_{X\sim Q}[r(X)] = \sum_{x\in\Omega} q(x)r(x) = 0 \tag{4.6}$$

$$\mathbb{E}_{X\sim Q}[|r(X)|] = \sum_{x\in\Omega} q(x)|r(x)| = \|P-Q\|_1 \tag{4.7}$$

$$KL(P||Q) = \mathbb{E}_{X\sim Q}[(1+r(X))\log((1+r(X)) - r(X)] \tag{4.8}$$

Combining (4.5) and (4.8) we obtain

$$KL(P||Q) \geq \frac{1}{2}\mathbb{E}_{X\sim Q}\left[\frac{r(X)^2}{1+\frac{r(X)}{3}}\right]$$

Thank to (4.6) we have that

$$\mathbb{E}_{X\sim q}\left[1+\frac{r(X)}{3}\right] = 1$$

so we have

$$KL(P||Q) \geq \frac{1}{2}\mathbb{E}_{X\sim Q}\left[\frac{r(X)^2}{1+\frac{r(X)}{3}}\right]\mathbb{E}_{X\sim Q}\left[1+\frac{r(X)}{3}\right]$$

Using $f(x) = \sqrt{r(x)^2/(1+r(x)/3)}$ and $g(x) = \sqrt{1+r(x)/3}$, thanks to the Cauchy-Schwarz inequality we have

$$\begin{aligned}
KL(P||Q) \geq & \frac{1}{2}\left(\mathbb{E}_{X\sim Q}\left[f(X)^2\right]\mathbb{E}_{X\sim Q}\left[g(X)^2\right]\right) \\
\geq & \frac{1}{2}\left(\mathbb{E}_{X\sim Q}\left[f(X)g(X)\right]\right)^2 \\
\geq & \frac{1}{2}\left(\mathbb{E}_{X\sim Q}\left[\frac{|r(X)|}{\sqrt{1+r(X)/3}}\cdot\sqrt{1+r(X)/3}\right]\right)^2 \\
\geq & \frac{1}{2}\left(\mathbb{E}_{X\sim Q}\left[|r(X)|\right]\right)^2 = \frac{1}{2}\|P-Q\|_1
\end{aligned}$$

□

Now we are ready to prove that the expected value of the the minimums of the problems (1.5) and (1.8) don't tend to 0 by $n \to +\infty$, if we fix $\varepsilon > 0$. We'll show this result only for the problem (1.8), for the other problem the proof is analogue. Remember that we're assuming that $c \in \mathbb{R}^{n\times n\times n}$ is a random vector, where the components are indipendent uniform random variables in $[0, 1]$.

Proposition 41. *Let $\varepsilon > 0$ be a fixed constant. Let z_n^* be the minimum of the problem (1.8), we have the following result*

$$\liminf_{n\to+\infty} \mathbb{E}[z_n^*] \geq \frac{1}{2} + \frac{\left(1-\sqrt{1+\varepsilon}\right)}{\varepsilon}$$

Proof. For some $n \in \mathbb{N}$, indicating $\pi^n \in S \times S \times S$ the probability distribution that realizes the minimum point of the problem (1.8), suppose we have

$$\mathbb{E}\left[z_n^*\right] \leq \Lambda$$

for some $\Lambda \in (0, +\infty)$. Then we have

$$\mathbb{E}\left[KL\left(\pi^n || \left(\frac{1}{n^3}\right)\right)\right] \leq \frac{\Lambda}{\varepsilon} \tag{4.9}$$

$$\mathbb{E}\left[\sum_{i,j,k} c_{i,j,k}\pi^n_{i,j,k}\right] \leq \Lambda \tag{4.10}$$

Thanks to the Pinsker's inequality and the Jensen's inequality, we have

$$\mathbb{E}\left[\sum_{i,j,k}\left|\pi^n_{i,j,k} - \frac{1}{n^3}\right|\right] \leq \sqrt{\frac{2\Lambda}{\varepsilon}} \implies \mathbb{E}\left[\sum_{i,j,k} c_{i,j,k}\pi^n_{i,j,k}\right] =$$

$$= \mathbb{E}\left[\sum_{i,j,k} c_{i,j,k}\left(\pi^n_{i,j,k} + \frac{1}{n^3} - \frac{1}{n^3}\right)\right] = \frac{1}{2} + \mathbb{E}\left[\sum_{i,j,k} c_{i,j,k}\left(\pi^n_{i,j,k} - \frac{1}{n^3}\right)\right] \geq$$

$$\geq \frac{1}{2} - \sqrt{\frac{2\Lambda}{\varepsilon}}$$

So, thanks to (4.10), we have necessarily

$$\Lambda \geq \frac{1}{2} - \sqrt{\frac{2\Lambda}{\varepsilon}}$$

Solving this inequality, we obtain

$$\Lambda \geq \frac{1}{2}\left(1 + \frac{2}{\varepsilon} - \sqrt{\frac{4}{\varepsilon}\left(1+\frac{1}{\varepsilon}\right)}\right) = \frac{1}{2} + \frac{\left(1-\sqrt{1+\varepsilon}\right)}{\varepsilon} \doteqdot \Lambda_\varepsilon$$

So, necessarily, we have that $\forall\ 0 < \eta < \Lambda_\varepsilon\ \mathbb{E}[z_n^*] > \eta$.

□

A consequence of this result is that if we fix $\varepsilon > 0$ then, increasing $n \in \mathbb{N}$, $\mathbb{E}[z_n^*]$ doesn't have the same behavior of the expected value of the optimal value of the linear problems.

For the problem (1.8), if we choose $\varepsilon = o(1/(n^2 \log(n)))$, increasing $n \in \mathbb{N}$ the behavior of the optimal value is similar to the behavior of the optimal value of the linear problem, in fact we have

$$0 \leq KL\left(p \,||\, \left(\frac{1}{n^3}\right)\right) = \sum_{i,j,k} \left(p_{i,j,k} \log(p_{i,j,k})\right) + 3\log(n) \leq 3\log(n)$$

so, in this case, we have that

$$\varepsilon KL\left(p \,||\, \left(\frac{1}{n^3}\right)\right) = o\left(\frac{1}{n^2}\right)$$

and due to the fact that the expected value of the optimal value of the linear problem is $\Theta(1/n^2)$ we can conclude. This fact justifies the choice of $\varepsilon = 1/(n^2 \log^2(n))$ on the numerical experiments.
An analogue argumentation can be used for the problem (1.5), and obtain that a good choice is $\varepsilon = o(1/(n \log(n)))$.

Obviously we have this result for the problem (1.5).

Proposition 42. *Let $\varepsilon > 0$ be a fixed constant. Let z_n^* be the minimum of the problem (1.5), we have the following result*

$$\mathbb{E}[z_n^*] \geq \frac{1}{2} + \frac{\left(1 - \sqrt{1+\varepsilon}\right)}{\varepsilon}$$

4.3 The Kullback-Leibler divergence between the optimal transport plans and the uniform distribution

The equivalence between the problem (1.4) and (1.2) let us to prove that increasing n the optimal transport plan for the problem (1.4) moves away from the uniform distribution.

Proposition 43. *For each $n \in \mathbb{N}$ and $c \in \mathbb{R}^{n\times n}$, if we indicate p^n the optimal transport plan ofor the problem (1.4), we have*

$$KL\left(p^n || \left(\frac{1}{n^2}\right)\right) = \log(n)$$

Proof. We know that $\exists \sigma \in S_n$ such that

$$p^n_{i,j} = \begin{cases} 0 & if \;\; j \neq \sigma(i) \\ \frac{1}{n} & if \;\; j = \sigma(i) \end{cases}$$

i.e. the optimal transport plan is identifiable with the function $\sigma \in S_n$. It follows

$$\begin{aligned} KL\left(p^n || \left(\frac{1}{n^2}\right)\right) &= \sum_{i,j} p^n_{i,j} \log\left(n^2 p^n_{i,j}\right) \\ &= \sum_{i} p_{i,\sigma(i)} \log(n) = \log(n) \end{aligned}$$

□

Let's see that the optimal transport plan for the problem (1.6) is not always a 'function': generating randomly the cost function $c \in [0,1]^{n\times n\times n}$, we have a unique solution a.s., because to have more than one solution a necessary condition is that the cost function is orthogonal to at least one of the constraints, and this happens with probability 0. So, generating a cost function on *Matlab* and solving the problem with the simplex algorithm, if the optimal transport plan obtained has more than n components not equal to 0, we can conclude that an optimal transport plan identifiable with a function doesn't exists.
However, we can conclude that the Kullback-Leibler divergence between the optimal transport plan and the uniform distribution increases as $2\log(n)$, but the proof is more hardworking.

Proposition 44. *For each $n \in \mathbb{N}$ and $c \in \mathbb{R}^{n\times n\times n}$ we indicate p^n the optimal transport plan for the problem (1.6). We have*

$$KL\left(p^n || \left(\frac{1}{n^3}\right)\right) \sim 2\log(n)$$

Proof. We have seen in the simplex algorithm section, that the optimal solution p^n has at most $3n$ components not equal to 0. We group those components in n groups (one for each first component) with k_i elements (i.e.

k_i is the number of elements $\neq 0$ in the matrix $p^n(i,:,:))$ for $i = 1, \ldots, n$, which we will call $\lambda_{1,i}, \ldots, \lambda_{k_i,i}$, such that

$$\sum_{j=1}^{k_i} \lambda_{j,i} = \frac{1}{n}$$

We observe that $k_i \geq 1$. So we have

$$\begin{aligned}
KL\left(p^n || \left(\frac{1}{n^3}\right)\right) =& \left(\sum_{i=1}^{n}\sum_{j=1}^{k_i} \lambda_{j,i}\log(\lambda_{j,i})\right) + 3\log(n) \\
=& \frac{1}{n}\left(\sum_{i=1}^{n}\sum_{j=1}^{k_i} n\lambda_{j,i}\log\left(\frac{n\lambda_{j,i}}{n}\right)\right) + 3\log(n) \\
=& \frac{1}{n}\left(\sum_{i=1}^{n}\sum_{j=1}^{k_i} n\lambda_{j,i}\left(\log\left(n\lambda_{j,i}\right) - \log(n)\right)\right) + 3\log(n) \\
=& \frac{1}{n}\left(\sum_{i=1}^{n}\sum_{j=1}^{k_i} n\lambda_{j,i}\log\left(n\lambda_{j,i}\right)\right) + 2\log(n) \\
=& \frac{1}{n}\left(\sum_{i=1}^{n} k_i \sum_{j=1}^{k_i} \frac{n\lambda_{j,i}}{k_i}\log\left(n\lambda_{j,i}\right)\right) + 2\log(n)
\end{aligned}$$

Now, using the convexity inequality for the function $f(x) = x\log(x)$, we have:

$$\begin{aligned}
KL\left(p^n || \left(\frac{1}{n^3}\right)\right) \geq& \frac{1}{n}\sum_{i=1}^{n} k_i \left(\sum_{j=1}^{k_i} \frac{n\lambda_{j,i}}{k_i}\right)\log\left(\sum_{j=1}^{k_i} \frac{n\lambda j, i}{k_i}\right) + 2\log(n) \\
=& \frac{1}{n}\sum_{i=1}^{n}\left(\log\left(\sum_{j=1}^{k_i} n\lambda_{j,i}\right) - \log(k_i)\right) + 2\log(n) \\
=& \frac{1}{n}\left(\sum_{i=1}^{n} -\log(k_i)\right) + 2\log(n) \\
\geq& \frac{1}{n}\left(\sum_{i=1}^{n} -k_i\right) + 2\log(n) \\
\geq& 2\log(n) - 3
\end{aligned}$$

To obtain the other inequality, we use the same argument, but then we use

the concavity of the function $g(x) = \log(x)$:

$$\begin{aligned}
KL\left(p^n || \left(\frac{1}{n^3}\right)\right) =& \left(\sum_{i=1}^{n}\sum_{j=1}^{k_i} \lambda_{j,i} \log(\lambda_{j,i})\right) + 3\log(n) \\
=& \frac{1}{n}\left(\sum_{i=1}^{n}\sum_{j=1}^{k_i} n\lambda_{j,i} \log\left(n\lambda_{j,i}\right)\right) + 2\log(n) \\
\leq& \frac{1}{n}\left(\sum_{i=1}^{n} \log\left(\sum_{j=1}^{k_i} n^2\lambda_{j,i}^2\right)\right) + 2\log(n) \\
\leq& \frac{1}{n}\left(\sum_{i=1}^{n} \left(\sum_{j=1}^{k_i} n^2\lambda_{j,i}^2\right)\right) + 2\log(n) \\
\leq& \frac{1}{n}\left(\sum_{i=1}^{n} \left(\sum_{j=1}^{k_i} n\lambda_{j,i}\right)^2\right) + 2\log(n) \\
=& 2\log(n) + 1
\end{aligned}$$

□

Bibliography

[1] Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

[2] Lev M Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 1967.

[3] J Michael Steele. *Probability theory and combinatorial optimization*, volume 69. Siam, 1997.

Ringraziamenti

Non so se questa sia la parte più difficile da scrivere, perché si vorrebbero dire mille cose ma forse non ci saranno mai parole adatte per spiegare tutto, o se sia la più facile perché è l'unica che non devo scrivere in inglese. In ogni caso proverò a far capire a tutti quanto siete importanti per me, anche se spesso sono scontroso e schivo, e infatti inizio col chiedere scusa a tutti quelli a cui voglio bene che ho trattato male anche solo una volta, sono fatto così e sto provando a cambiare in questo...
Devo ringraziare i miei genitori e tutta la mia famiglia, è grazie a loro se ho avuto l'opportunità di arrivare fin qui, mi hanno sempre supportato, e da lontano mi sono stati vicini per assicurarsi che andasse sempre tutto bene.
Mi tocca ringraziare anche quei delinquenti dei 'tignosi', coi quali mi sono divertito non so quanto ad ogni festa che abbiamo organizzato, che ti strappano sempre una risata ad ogni boiata che inviano su WhatsApp, anche nelle giornate peggiori.
Ringrazio tutti quelli che per essere qui con me a festeggiare questo traguardo si faranno centinaia di chilometri, non so se riuscite ad immaginarvi quanto io sia felice che siate tutti qua...
Come non ringraziare gli 'inquilini' del dipartimento? Dato che quel posto è come casa per noi, posso dire che siete una seconda famiglia, e stare sempre in quel posto insieme a tante altre persone è stato fondamentale per me per entrare nella matematica, ma oltre a questo, in dipartimento mi diverto un sacco e spesso vado lì anche solo per stare spensierato insieme a voi.
Ringrazio Sara, che mi è sempre stata vicina in ogni momento di difficoltà, e fidatevi che ne ho avuti tanti, magari dovuti a paure e dubbi stupidi, ma in ogni caso lei c'è sempre stata, e questo mi ha aiutato davvero tanto.
Nonostante non sia presente né alla discussione, né alla festa, ringrazio Giovanni per essere sempre stato un mio amico, e per fargli capire che non è importante che sia presente oggi, ma sono importanti tutte le cose che abbiamo fatto insieme, feste, cene, preparazione agli esami.
Ringrazio i professori che svolgono il loro lavoro col massimo impegno, è anche

grazie ad alcuni di loro che ho capito la bellezza della matematica.
Ringrazio infine chiunque mi abbia fatto passare dei bei momenti, o mi abbia anche solo strappato un sorriso, perché alla fine la felicità è fatta di attimi di dimenticanza...