HAL open science

Overrelaxed Sinkhorn-Knopp Algorithm for Regularized Optimal Transport

Alexis Thibault, Lénaïc Chizat, Charles Dossal, Nicolas Papadakis

To cite this version:

Alexis Thibault, Lénaïc Chizat, Charles Dossal, Nicolas Papadakis. Overrelaxed Sinkhorn-Knopp Algorithm for Regularized Optimal Transport. Algorithms, 2021, Optimal Transport: Algorithms and Applications, 14 (5), 10.3390/a14050143 . hal-03212175

HAL Id: hal-03212175

https://hal.science/hal-03212175v1

Submitted on 29 Apr 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Overrelaxed Sinkhorn–Knopp Algorithm for Regularized Optimal Transport

Alexis Thibault[1], Lénaïc Chizat[2], Charles Dossal[3], and Nicolas Papadakis[4]

[1]Inria Bordeaux Sud Ouest, Talence, France
[2]CNRS, Laboratoire de Mathématiques d'Orsay, Université Paris-Saclay, France
[3]INSA Toulouse, France
[4]CNRS, Univ. Bordeaux, IMB, F-33400 Talence, France

Abstract

This article describes a set of methods for quickly computing the solution to the regularized optimal transport problem. It generalizes and improves upon the widely used iterative Bregman projections algorithm (or Sinkhorn–Knopp algorithm). We first proposed to rely on regularized nonlinear acceleration schemes. In practice, such approaches lead to fast algorithms, but their global convergence is not ensured. Hence, we next proposed a new algorithm with convergence guarantees. The idea is to overrelax the Bregman projection operators, allowing for faster convergence. We proposed a simple method for establishing global convergence by ensuring the decrease of a Lyapunov function at each step. An adaptive choice of the overrelaxation parameter based on the Lyapunov function was constructed. We also suggested a heuristic to choose a suitable asymptotic overrelaxation parameter, based on a local convergence analysis. Our numerical experiments showed a gain in convergence speed by an order of magnitude in certain regimes.

1 Introduction

Optimal transport (OT) is an efficient and flexible tool to compare two probability distributions, which has been popularized in the computer vision community in the context of discrete histograms [26]. The introduction of entropic regularization of the optimal transport problem in [10] has made possible the use of the fast Sinkhorn–Knopp algorithm [32], scaling with high-dimensional data. Regularized optimal transport has thus been intensively used in machine learning with applications such as geodesic PCA [31], domain adaptation [9], data fitting [12], training of a Boltzmann machine [18], or dictionary learning [25, 27].

The computation of optimal transport between two data relies on the estimation of an optimal transport matrix, the entries of which represent the quantity of mass transported between data locations. Regularization of optimal transport with strictly convex regularization [10, 11], nevertheless, involves a spreading of the mass. Hence, for particular purposes such as color interpolation [23] or gradient flow [7], it is necessary to consider small parameters ε for the entropic regularization term. The Sinkhorn–Knopp (SK) algorithm is a state-of-the-art algorithm to solve the regularized transport problem. The SK algorithm performs alternated projections, and the sequence of generated iterates converges to a solution of the regularized transport problem. Unfortunately, the lower ε is, the slower the SK algorithm converges. To improve the convergence rate of the SK algorithm, several acceleration strategies have been proposed in the literature, based for example on mixing or overrelaxation.

1.1 Accelerations of the Sinkhorn–Knopp Algorithm

In the literature, several accelerations of the Sinkhorn–Knopp algorithm have been proposed, using for instance greedy coordinate descent [2] or screening strategies [1]. In another line of research, the introduction of relaxation variables through heavy ball approaches [22] has recently gained popularity to speed up the convergence of algorithms optimizing convex [13] or non-convex [36, 19] problems. In this context, the use of regularized nonlinear accelerations (RNAs) [3, 29, 30] based on Anderson mixing has led to important numerical improvements, although the global convergence is not guaranteed with such approaches, as is shown further. In this

paper, we also investigated another approach related to the successive overrelaxation (SOR) algorithm [35], which is a classical way to solve linear systems. Similar schemes have been empirically considered to accelerate the SK algorithm in [20, 27]. The convergence of these algorithms has nevertheless not been studied yet in the context of regularized optimal transport.

Recent progress has been made on computational complexity guarantees for the Sinkhorn–Knopp algorithm and accelerated versions [2, **?**, **?**, **?**]. Since the methods we discuss in this paper are based on asymptotic acceleration techniques, it is challenging to show their efficiency via global computational complexity guarantee, and we do not cover these aspects here.

1.2 Overview and Contributions

The contribution of this paper is twofold. First, the numerical efficiency of the RNA methods applied to the SK algorithm to solve the regularized transport problem is shown. Second, a new extrapolation and relaxation technique for accelerating the Sinkhorn–Knopp (SK) algorithm while ensuring convergence is given. The numerical efficiency of this new algorithm is demonstrated, and a heuristic rule is also proposed to improve the rate of the algorithm.

Section 2 is devoted to the Sinkhorn–Knopp algorithm. In Section 3, we propose to apply regularized nonlinear acceleration (RNA) schemes to the SK algorithm. We experimentally show that such methods lead to impressive accelerations for low values of the entropic regularization parameter. In order to have a globally converging method, we then propose a new overrelaxed algorithm: Sinkhorn–Knopp with successive overrelaxation (SK-SOR). In Section 4, we show the global convergence of this algorithm and analyze its local convergence rate to justify the acceleration. We finally demonstrate numerically in Section 5 the interest of our method. Larger accelerations are indeed observed for decreasing values of the entropic regularization parameter.

Remark This paper is an updated version of an unpublished work [34] presented at the NIPS 2017 Workshop on Optimal Transport & Machine Learning. In the meantime, complementary results on the global convergence of our method presented in Section 4 were provided in [17]. The authors showed the existence of a parameter θ_0 such that both global convergence and local acceleration were ensured for overrelaxation parameters $\omega \in (1, \theta_0)$. This result was nevertheless theoretical, and the numerical estimation of θ_0 is still an open question. With respect to our unpublished work [34], the current article presents an original contribution in Section 3: the application of RNA methods to accelerate the convergence of the SK algorithm.

2 Sinkhorn Algorithm

Before going into further details, we now briefly introduce the main notations and concepts used throughout this article.

2.1 Discrete Optimal Transport

We considered two discrete probability measures $\mu_k \in \mathbb{R}^{n_k}_{+*}$. Let us define the two following linear operators:

$$A_1 : \begin{cases} \mathbb{R}^{n_1 n_2} \to \mathbb{R}^{n_1} \\ (A_1 x)_i = \sum_j x_{i,j} \end{cases} \qquad A_2 : \begin{cases} \mathbb{R}^{n_1 n_2} \to \mathbb{R}^{n_2} \\ (A_2 x)_j = \sum_i x_{i,j}, \end{cases}$$

as well as the affine constraint sets:

$$\mathcal{C}_k = \{\gamma \in \mathbb{R}^{n_1 n_2} \mid A_k \gamma = \mu_k\}.$$

Given a cost matrix c with nonnegative coefficients, where $c_{i,j}$ represents the cost of moving mass $(\mu_1)_i$ to $(\mu_2)_j$, the optimal transport problem corresponds to the estimation of an optimal transport matrix γ solution of:

$$\min_{\gamma \in \mathcal{C}_1 \cap \mathcal{C}_2 \cap \mathbb{R}^{n_1 n_2}_+} \langle c, \gamma \rangle := \sum_{i,j} c_{i,j} \gamma_{i,j}.$$

This is a linear programming problem whose resolution becomes intractable for large problems.

2.2 Regularized Optimal Transport

In [10], it was proposed to regularize this problem by adding a strictly convex entropy regularization:

$$\min_{\gamma \in \mathcal{C}_1 \cap \mathcal{C}_2 \cap \mathbb{R}_+^{n_1 n_2}} K^\varepsilon(\gamma) \coloneqq \langle c, \gamma \rangle + \varepsilon \, \mathrm{KL}(\gamma, \mathbf{1}), \tag{1}$$

with $\varepsilon > 0$, $\mathbf{1}$ the matrix of size $n_1 \times n_2$ full of ones, and the Kullback–Leibler divergence is:

$$\mathrm{KL}(\gamma, \zeta) = \sum_{i,j} \gamma_{i,j} \left(\log \left(\frac{\gamma_{i,j}}{\zeta_{i,j}} \right) - 1 \right) + \sum_{i,j} \zeta_{i,j} \tag{2}$$

with the convention $0 \log 0 = 0$. It was shown in [4] that the regularized optimal transport matrix γ^*, which is the unique minimizer of Problem (1), is the Bregman projection of $\gamma^0 = e^{-c/\varepsilon}$ (here and in the sequel, exponentiation is meant entry-wise) onto $\mathcal{C}_1 \cap \mathcal{C}_2$:

$$\gamma^* = \operatorname*{argmin}_{\mathcal{C}_1 \cap \mathcal{C}_2} K^\varepsilon(\gamma) = P_{\mathcal{C}_1 \cap \mathcal{C}_2}(e^{-c/\varepsilon}), \tag{3}$$

where $P_{\mathcal{C}}$ is the Bregman projection onto $\mathcal{C}$ defined as:

$$P_{\mathcal{C}}(\zeta) \coloneqq \operatorname*{argmin}_{\gamma \in \mathcal{C}} \mathrm{KL}(\gamma, \zeta).$$

2.3 Sinkhorn–Knopp Algorithm

Iterative Bregman projections onto $\mathcal{C}_1$ and $\mathcal{C}_2$ converge to a point in the intersection $\mathcal{C}_1 \cap \mathcal{C}_2$ [5]. Hence, the so-called Sinkhorn–Knopp (SK) algorithm [32] that performs alternate Bregman projections can be considered to compute the regularized transport matrix:

$$\gamma^0 = e^{-c/\varepsilon} \qquad\qquad \gamma^{\ell+1} = P_{\mathcal{C}_2}(P_{\mathcal{C}_1}(\gamma^\ell)),$$

and we have $\lim_{\ell \to +\infty} \gamma^\ell = P_{\mathcal{C}_1 \cap \mathcal{C}_2}(\gamma^0) = \gamma^*$.

In the discrete setting, these projections correspond to diagonal scalings of the input:

$$\begin{aligned} P_{\mathcal{C}_1}(\gamma) &= \mathrm{diag}(u)\gamma \qquad &&\text{with} \quad u = \mu_1 \oslash A_1 \gamma \\ P_{\mathcal{C}_2}(\gamma) &= \gamma \, \mathrm{diag}(v) \qquad &&\text{with} \quad v = \mu_2 \oslash A_2 \gamma \end{aligned} \tag{4}$$

where $\oslash$ is the pointwise division. To compute the solution numerically, one simply has to store $(u^\ell, v^\ell) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ and to iterate:

$$u^{\ell+1} = \mu_1 \oslash \gamma^0 v^\ell \qquad\qquad v^{\ell+1} = \mu_2 \oslash {}^t\gamma^0 u^{\ell+1}. \tag{5}$$

We then have $\gamma^\ell = \mathrm{diag}(u^\ell)\gamma^0 \, \mathrm{diag}(v^\ell)$.

Another way to interpret the SK algorithm is as an alternate maximization algorithm on the dual of the regularized optimal transport problem; see [21], Remark 4.24. The dual problem of (1) is:

$$\max_{\substack{\alpha \in \mathbb{R}^{n_1} \\ \beta \in \mathbb{R}^{n_2}}} E(\alpha, \beta) \coloneqq \langle \alpha, \mu_1 \rangle + \langle \beta, \mu_2 \rangle - \varepsilon \sum_{i,j} e^{(\alpha_i + \beta_j - c_{i,j})/\varepsilon}. \tag{6}$$

As the function E is concave, continuously differentiable, and admits a maximizer, the following alternate maximization algorithm converges to a global optimum:

$$\alpha^{\ell+1} = \operatorname*{argmax}_{\alpha} E(\alpha, \beta^\ell) \tag{7}$$

$$\beta^{\ell+1} = \operatorname*{argmax}_{\beta} E(\alpha^{\ell+1}, \beta). \tag{8}$$

The explicit solutions of these problems are:

$$\alpha_i^{\ell+1} = \varepsilon \log \left(\sum_j \exp \left(\log(\mu_1)_i - \left(\beta_j^\ell - c_{i,j} \right) / \varepsilon \right) \right) \qquad i = 1 \cdots n_1 \tag{9}$$

$$\beta_j^{\ell+1} = \varepsilon \log \left(\sum_i \exp \left(\log(\mu_2)_j - \left(\alpha_i^{\ell+1} - c_{i,j} \right) / \varepsilon \right) \right) \qquad j = 1 \cdots n_2 \tag{10}$$

and we recover the SK algorithm (5) by taking $u_i = e^{\alpha_i/\varepsilon}$, $v_j = e^{\beta_j/\varepsilon}$, and $\gamma^0_{i,j} = e^{-c_{i,j}/\varepsilon}$.

Efficient parallel computations can be considered [10], and one can almost reach real-time computation for large-scale problems for certain classes of cost matrices c, allowing the use of separable convolutions [33]. For low values of the parameter ε, numerical issues can arise, and the log stabilization version of the algorithm presented in Relations (9) and (10) is necessary [7]. Above all, the linear rate of convergence degrades as $\varepsilon \to 0$ (see for instance Chapter 4 in [21]). In the following sections, we introduce different numerical schemes that accelerate the convergence in the regime $\varepsilon \to 0$.

3 Regularized Nonlinear Acceleration of the Sinkhorn–Knopp Algorithm

In order to accelerate the SK algorithm for low values of the regularization parameter ε, we propose to rely on regularized nonlinear acceleration (RNA) techniques. In Section 3.1, we first introduce RNA methods. The application to SK is then detailed in Section 3.2.

3.1 Regularized Nonlinear Acceleration

To introduce RNA, we first rewrite the SK algorithm (7) and (8) as:

$$\beta^{\ell+1} = \text{SK}(\beta^\ell) := \underset{\beta}{\operatorname{argmax}}\, E\left(\underset{\alpha}{\operatorname{argmax}}\, E(\alpha, \beta^\ell), \beta\right). \tag{11}$$

The goal of this algorithm is to build a sequence $(\beta^\ell)_{l\geqslant 1}$ converging to a fixed point of SK, i.e., to a point β^* satisfying:

$$\beta^* = SK(\beta^*). \tag{12}$$

Many optimization problems can be recast as fixed-point problems. The Anderson acceleration or Anderson mixing is a classical method to build a sequence $(x^n)_n$ that converges numerically fast to a fixed point of any operator T from $\mathbb{R}^N$ to $\mathbb{R}^N$. This method defines at each step a linear (but not necessarily convex) combination of some previous values of $(x^k)_k$ and $(Tx^k)_k$ to provide a value of x^n such that $\|x^n - Tx^n\|$ is as low as possible.

Numerically, fast local convergence rates can be observed when the operator T is smooth. This method can nevertheless be unstable even in the favorable setting where T is affine. Such case arises for instance when minimizing a quadratic function F with the descent operator $T = I - h\nabla F$, with time step $h > 0$. Unfortunately, there are no convergence guarantees, in a general setting or in the case $T = SK$, that the RNA sequence $(x^n)_n$ converges for any starting point x^0.

RNA is an algorithm that can be seen as a generalization of the Anderson acceleration. It can also be applied to any fixed-point problem. The RNA method [30] applied to algorithm (11) using at each step the N previous iterates is:

$$(s_l^{\ell+1})_{l=0}^{N-1} = \underset{\mathbf{s};\ \sum_{l=0}^{N-1} s_l = 1}{\operatorname{argmin}} \mathcal{P}_\lambda\left(\mathbf{s}, (\beta^{\ell-l} - y^{\ell-l})_{l=0}^{N-1}\right) \tag{13}$$

$$y^{\ell+1} = \sum_{l=0}^{N-1} s_l^{\ell+1}(y^{\ell-l} + \omega(\beta^{\ell-l} - y^{\ell-l})) \tag{14}$$

$$\beta^{\ell+1} = \text{SK}(y^{\ell+1}), \tag{15}$$

where extrapolation weights $\mathbf{s} = (s_l)_{l=0}^{N-1}$ are the solution to the optimization problem (16) defined below, and ω is a relaxation parameter. RNA uses the memory of the past trajectory when $N > 1$. We can remark that for $N = 1$, $s_0^\ell = 1$ for all $\ell \geqslant 0$, and RNA is reduced to a simple relaxation parameterized by ω:

$$y^{\ell+1} = y^\ell + \omega(\beta^\ell - y^\ell).$$

Let us now discuss the role of the different RNA parameters ω and $\mathbf{s}$.

Relaxation: Taking origins from Richardson's method [24], relaxation leads to numerical convergence improvements in gradient descent schemes [15]. Anderson suggested underrelaxing with $\omega \in (0; 1]$, while the authors of [30] proposed to take $\omega = 1$.

Extrapolation: Let us define the residual $r(y) = \text{SK}(y) - y$. As the objective is to estimate the fixed point of SK, the extrapolation step builds a vector y such that $||r(y)||$ is minimal. A relevant guess of such y

is obtained by looking at a linear combination of previous iterates that reaches this minimum. More precisely, RNA methods estimate the weight vector $(s_l^{\ell+1})_{l=0}^{N-1}$ as the unique solution of the regularized problem:

$$(s_l^{\ell+1})_{l=0}^{N-1} = \underset{\mathbf{s};\ \sum_{l=0}^{N-1} s_l = 1}{\operatorname{argmin}} \mathcal{P}_\lambda(\mathbf{s}, R) := ||R\mathbf{s}||^2 + \lambda ||\mathbf{s}||^2 \tag{16}$$

$$= \frac{({}^tRR + \lambda \mathrm{Id}_N)^{-1}\mathbf{1}_N}{\langle ({}^tRR + \lambda \mathrm{Id}_N)^{-1}\mathbf{1}_N, \mathbf{1}_N\rangle}. \tag{17}$$

where the columns of $R := [r(y^\ell), \cdots, r(y^{\ell+1-N})]$ are the N previous residuals. The regularization parameter $\lambda > 0$ generalizes the original Anderson acceleration [3] introduced for $\lambda = 0$. Taking $\lambda > 0$ indeed leads to a more stable numerical estimation of the extrapolation parameters.

3.2 Application to SK

We now detail the whole Sinkhorn–Knopp algorithm using regularized nonlinear acceleration, which is presented in Algorithm 1.

In all our experiments corresponding to $N > 1$, we considered a regularization $\lambda = 1 \times 10^{-10}$ for the weight estimation (16) within the RNA scheme. For the SK algorithm, we considered the log-stabilization implementation proposed in [7, 28, 27] to avoid numerical errors for low values of ε. This algorithm acts on the dual variables α, β. We refer to the aforementioned papers for more details.

As the SK algorithm successively projects the matrix γ^ℓ onto the set of linear constraints $\mathcal{C}_k$, $k = 1, 2$, we took as the convergence criterion the error realized on the first marginal of the transport matrix $\gamma = \mathrm{diag}(\exp(\alpha/\varepsilon)) \exp(-c/\varepsilon) \,\mathrm{diag}(\exp(\beta/\varepsilon))$, that is $\sum_i |\sum_j \gamma_{i,j}^\ell - (\mu_1)_i| < \eta = 1e-9$. Note that the variable α is introduced into the algorithm only for computing the convergence criterion.

Algorithm 1 RNA SK Algorithm in the Log Domain

Require: $\mu_1 \in \mathbb{R}^{n_1}$, $\mu_2 \in \mathbb{R}^{n_2}$, $c \in \mathbb{R}_+^{n_1 \times n_2}$
  Set $\ell = 0$, $\beta^0, y^0 = \mathbf{0}_{n_2}$, $\gamma^0 = \exp -c/\varepsilon$, $\omega \in \mathbb{R}$, $N > 0$ and $\eta > 0$
  Set $x_i = -\max_j(c_{i,j} - y_j^0)$ and $\alpha_i = -\max_j(c_{i,j} - \beta_j^0)$, $\quad i = 1 \cdots n_1$
  while $|| \exp(\alpha/\varepsilon) \otimes (\gamma^0 \exp(\beta^\ell/\varepsilon)) - \mu_1|| > \eta$ **do**
    $\tilde{N} = \min(N, \ell + 1)$
    $R = [\beta^\ell - y^\ell, \cdots, \beta^{\ell+1-\tilde{N}} - y^{\ell+1-\tilde{N}}]$
    $\mathbf{w} = ({}^tRR + \lambda \mathrm{Id}_{\tilde{N}})^{-1}\mathbf{1}_{\tilde{N}} / \langle ({}^tRR + \lambda \mathrm{Id}_{\tilde{N}})^{-1}\mathbf{1}_{\tilde{N}}, \mathbf{1}_{\tilde{N}}\rangle$
    $y^{\ell+1} = \sum_{l=0}^{\tilde{N}-1} w_l((1-\omega)y^{\ell-l} + \omega\beta^{\ell-l})$
    $\tilde{x}_i = -\max_j(c_{i,j} - y_j^{\ell+1})$, $\quad i = 1 \cdots n_1$
    $x_i = \tilde{x}_i - \varepsilon \log(\sum_j \exp((-c_{i,j} + \tilde{x}_i + y_j^{\ell+1})/\varepsilon - \log(\mu_1)_i))$, $\quad i = 1 \cdots n_1$
    $\beta_j^{\ell+1} = y^{\ell+1} - \varepsilon \log(\sum_i \exp((-c_{i,j} + x_i + y_j^{\ell+1})/\varepsilon - \log(\mu_2)_j))$, $\quad j = 1 \cdots n_2$
    $\alpha_i = -\max_j(c_{i,j} - \beta_j^{\ell+1})$, $\quad i = 1 \cdots n_1$
    $\ell \leftarrow \ell + 1$
  end while
  return $\gamma_{i,j} = \mathrm{diag}(\exp(\alpha/\varepsilon))\gamma^0 \,\mathrm{diag}(\exp(\beta/\varepsilon))$

We now present numerical results obtained with random cost matrices of size 100×100 with entries uniform in $[0, 1]$ and uniform marginals μ_1 and μ_2. All convergence plots are mean results over 20 realizations.

We first considered the relaxation parameter $\omega = 1$, in order to recover the original SK algorithm for $N = 1$. In Figure 1, we show the convergence results obtained with RNA orders $N \in \{1, 2, 4, 8\}$ on four regularized transport problems corresponding to entropic parameters $\varepsilon \in \{0.003, 0.01, 0.03, 0.1\}$. Figure 1 first illustrates that the convergence is improved with higher RNA orders N.

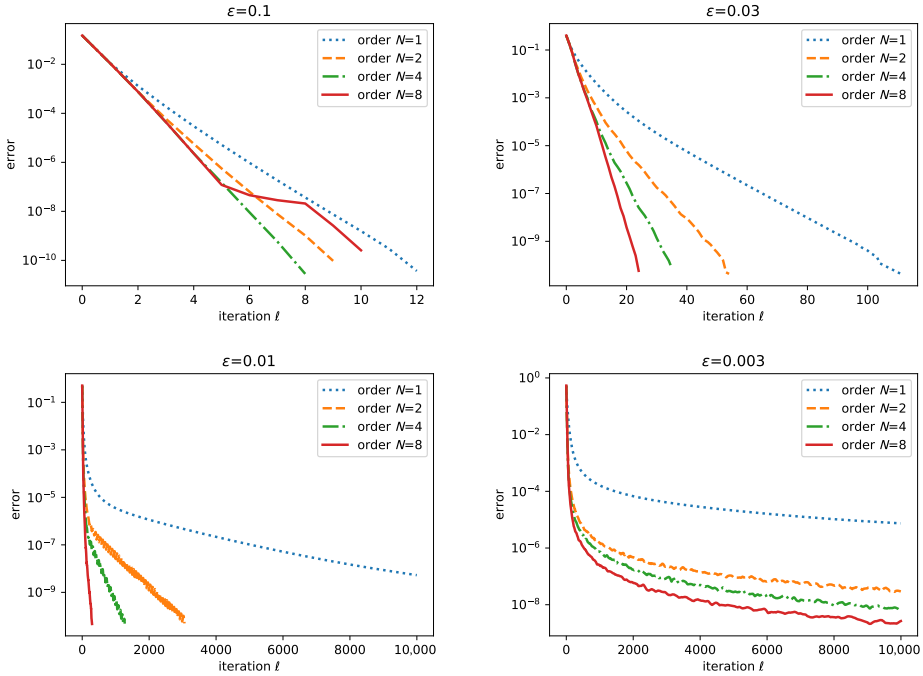


Figure 1: Convergence of RNA schemes for a relaxation parameter $\omega = 1$ and different orders $N \in \{1, 2, 4, 8\}$. All approaches lead to similar convergence as the original SK algorithm (order $N = 1$, with blue dots) for high values of the entropic parameter such as $\varepsilon = 0.1$. When facing more challenging regularized optimal transport problems, high-order RNA schemes ($N > 1$) realize important accelerations. This behavior is highlighted in the bottom row for $\varepsilon = 0.01$ and $\varepsilon = 0.003$. In these settings, with respect to SK, RNA of order $N = 8$ (plain red curves) reduces by a factor 100 the number of iterations required to reach the same accuracy.

The acceleration is also greater for low values of the regularization parameter ε. This is an important behavior, as many iterations are required to have an accurate estimation of these challenging regularized transport problems. In the settings $\varepsilon \in \{0.003, 0.01\}$, a speedup of more than $\times 100$ in terms of the iteration number was observed between RNA orders $N = 8$ and $N = 1$ (SK) to reach the same convergence threshold. We did not observe a significant improvement by considering higher RNA orders such as $N = 16$.

Next, we focus on the influence of the relaxation parameter $\omega \in \{0.5, 1, 1.5, 1.9\}$ onto the behavior of RNA schemes of orders $N \in \{1, 2, 8\}$. We restricted our analysis to the more challenging settings $\varepsilon = 0.01$ and $\varepsilon = 0.003$. As illustrated in Figure 2, increasing ω systematically led to improvements in the case $N = 1$. For other RNA orders satisfying $N > 1$, we did not observe clear tendencies. Taking $\omega \approx 1.5$ generally allowed accelerating the convergence.

To sum up, we recommend the use of the parametrization $N = 8$ and $\omega = 1.5$ when applying the RNA method to the SK algorithm.

We recall that the convergence of such approaches is not ensured. This last experiment nevertheless suggested that in the case $N = 1$, there is room to accelerate the original SK algorithm ($\omega = 1$), while keeping its global convergence guarantees, by looking at overrelaxed schemes with parameters $\omega > 1$.

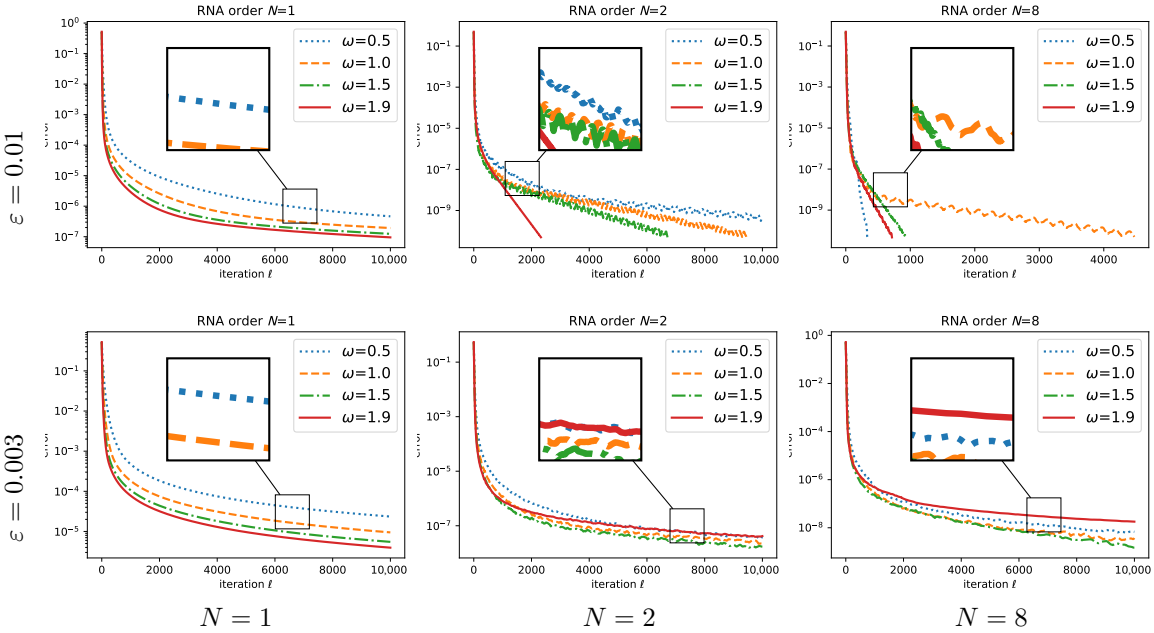


Figure 2: Comparison of RNA schemes on an optimal transport problem regularized with the entropic parameters $\varepsilon = 0.01$ (**top line**) and $\varepsilon = 0.003$ (**bottom line**). The convergence of RNA scheme $N \in \{1, 2, 8\}$ is illustrated for different relaxation parameters $\omega \in \{0.5, 1, 1.5, 1.9\}$. Higher values of ω lead to larger improvements in the case $N = 1$ (first column). When $N > 1$ (as in the middle column for $N = 2$ and the right column for $N = 8$), it is not possible to conclude and to suggest a choice for the parameter ω with the obtained numerical results.

3.3 Discussion

The nonlinear regularized acceleration algorithm involves relevant numerical accelerations without convergence guarantees. To build an algorithm that ensures the convergence of iterates, but also improves the numerical behavior of the SK algorithm, we now propose to follow a different approach using Lyapunov sequences, which is a classical tool to study optimization algorithms. The new scheme proposed here uses the specific form of the SK algorithm with the two variables α and β. It performs two successive overrelaxations (SOR) at each step, one for the update of α and one for β. The algorithm does not use any mixing scheme, but the simple structure allows defining a sequence, called a Lyapunov sequence, which decreases at each step. This Lyapunov approach allows ensuring the convergence of the algorithm for a suitable choice of the overrelaxation parameter.

The algorithm can be summarized as follows:

$$\alpha^{\ell+1} = (1-\omega)\alpha^{\ell} + \omega \operatorname*{argmax}_{\alpha} E(\alpha, \beta^{\ell}) \tag{18}$$

$$\beta^{\ell+1} = (1-\omega)\beta^{\ell} + \omega \operatorname*{argmax}_{\beta} E(\alpha^{\ell+1}, \beta). \tag{19}$$

Our convergence analysis relied on an online adaptation of the overrelaxation parameter $\omega \in [1, 2)$. As illustrated by Figure 3, in the case for $\varepsilon = 0.01$, the proposed SK-SOR method was not as performant as high RNA orders $N > 1$ with $\omega = 1.5$. It nevertheless gave an important improvement with respect to the original SK method, while being provably convergent.

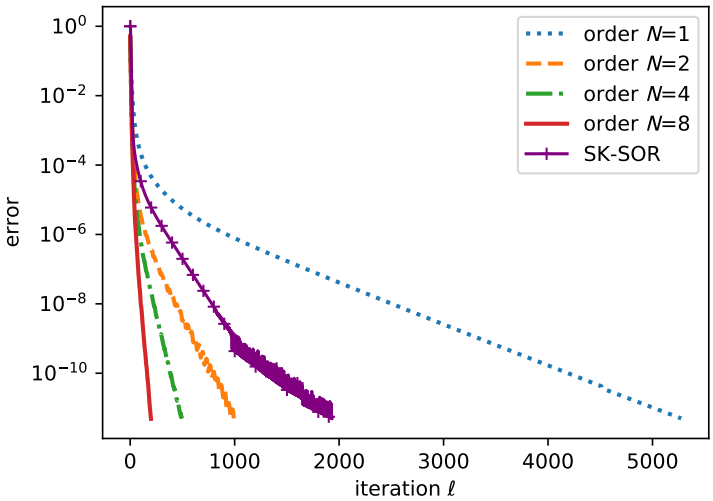
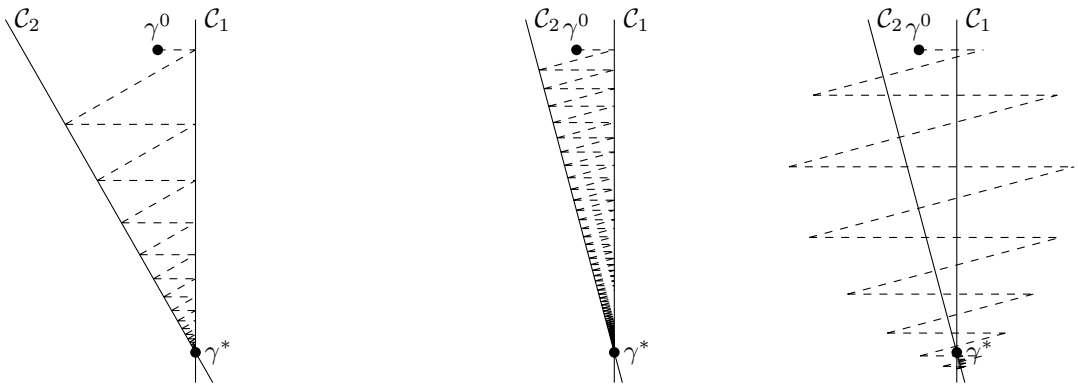


Figure 3: Comparison between RNA schemes with $\omega = 1.5$ and SK-SOR for a transport regularized with $\varepsilon = 0.01$. The SK-SOR performance is in between the ones of RNA of orders $N = 1$ and $N = 2$.

4 Sinkhorn–Knopp with Successive Overrelaxation

In this section, we propose a globally convergent overrelaxed SK algorithm. Different from the RNA point of view of the previous section, our algorithm SK-SOR relies on successive overrelaxed (SOR) projections.

As illustrated in Figure 4a,b, the original SK algorithm (5) performs alternate Bregman projections (4) onto the affine sets $\mathcal{C}_1$ and $\mathcal{C}_2$. In practice, the convergence may degrade when $\varepsilon \to 0$. The idea developed in this section is to perform overrelaxed projections in order to accelerate the process, as displayed in Figure 4c.



(a) (b) (c)

Figure 4: The trajectory of γ^ℓ given by the SK algorithm is illustrated for decreasing values of ε in (**a**,**b**). Overrelaxed projections (**c**) typically accelerate the convergence rate.

In what follows, we first define overrelaxed Bregman projections. We then propose a Lyapunov function that is used to show the global convergence of our proposed algorithm in Section 4.3. The local convergence rate is then discussed in Section 4.4.

4.1 Overrelaxed Projections

We recall that $P_{\mathcal{C}_k}$ are operators realizing the Bregman projection of matrices $\gamma \in \mathbb{R}^{n_1 n_2}$ onto the affine sets $\mathcal{C}_k$, $k = 1, 2$, as defined in (4). For $\omega \geq 0$, we now define the ω-relaxed projection operator $P^\omega_{\mathcal{C}_k}$ as:

$$\log P^\omega_{\mathcal{C}_k}(\gamma) = (1 - \omega) \log \gamma + \omega \log P_{\mathcal{C}_k}(\gamma), \tag{20}$$

where the logarithm is taken coordinate-wise. Note that $P^0_{\mathcal{C}_k}$ is the identity, $P^1_{\mathcal{C}_k} = P_{\mathcal{C}_k}$ is the standard Bregman projection, and $P^2_{\mathcal{C}_k}$ is an involution (in particular because $\mathcal{C}_k$ is an affine subspace). In the following, we will consider overrelaxations corresponding to $\omega \in [1; 2)$. A naive algorithm would then consist of iteratively applying $P^\omega_{\mathcal{C}_2} \circ P^\omega_{\mathcal{C}_1}$ for some choice of ω. While it often behaves well in practice, this algorithm may sometimes not converge even for reasonable values of ω. Our goal in this section is to make this algorithm robust and to guarantee its global convergence.

Remark 1. Duality gives another point of view on the iterative overrelaxed Bregman projections: they indeed correspond to a successive overrelaxation (SOR) algorithm on the dual objective E given in (6). This is a procedure that, starting from $(\alpha^0, \beta^0) = (\mathbf{0}, \mathbf{0})$, defines for $\ell \in \mathbb{N}^*$,

$$\alpha^{\ell+1} = (1-\omega)\alpha^\ell + \omega \operatorname*{argmax}_\alpha E(\alpha, \beta^\ell) \tag{21}$$

$$\beta^{\ell+1} = (1-\omega)\beta^\ell + \omega \operatorname*{argmax}_\beta E(\alpha^{\ell+1}, \beta), \tag{22}$$

From the definition of the projections in (4) and using again the relationships $u_i = e^{\alpha_i/\varepsilon}$, $v_j = e^{\beta_j/\varepsilon}$ and $\gamma^0_{i,j} = e^{-c_{i,j}/\varepsilon}$, Expressions (21) and (22) correspond to the overrelaxed projections (20).

4.2 Lyapunov Function

Convergence of the successive overrelaxed projections is not guaranteed in general. In order to derive a robust algorithm with provable convergence, we introduced the Lyapunov function:

$$F(\gamma) = \mathrm{KL}(\gamma^*, \gamma), \tag{23}$$

where γ^* denotes the solution of the regularized OT problem. We used this function to enforce the strict descent criterion $F(\gamma^{\ell+1}) < F(\gamma^\ell)$ as long as the process has not converged.

The choice of (23) as a Lyapunov function is of course related to the fact that Bregman projections are used throughout the algorithm. Further, we show (Lemma 1) that its decrease is simple to compute, and this descent criterion still allows enough freedom in the choice of the overrelaxation parameter.

Crucial properties of this Lyapunov function are gathered in the next lemma.

Lemma 1. *For any $M \in \mathbb{R}^*_+$, the sublevel set $\{\gamma \mid F(\gamma) \leq M\}$ is compact. Moreover, for any γ in $\mathbb{R}^{n_1 n_2}_{+*}$, the decrease of the Lyapunov function after an overrelaxed projection can be computed as:*

$$F(\gamma) - F(P^\omega_{\mathcal{C}_k}(\gamma)) = \langle \mu_k, \varphi_\omega\left((A_k\gamma) \oslash \mu_k\right)\rangle, \tag{24}$$

where

$$\varphi_\omega(x) = x(1 - x^{-\omega}) - \omega \log x \tag{25}$$

is a real function, applied coordinate-wise.

Proof. The fact that the Kullback–Leibler divergence is jointly lower semicontinuous implies in particular that K is closed. Moreover, $K \subset \mathbb{R}^{n_1 \times n_2}_+$ is bounded because F is the sum of nonnegative, coercive functions of each component of its argument γ.

Formula (24) comes from the expression:

$$F(\gamma^1) - F(\gamma^2) = \sum_{i,j} \left(\gamma^*_{i,j} \log(\gamma^2_{i,j}/\gamma^1_{i,j}) + \gamma^1_{i,j} - \gamma^2_{i,j}\right)$$

and Relations (4) and (20). □

It follows from Lemma 1 that the decrease of F for an overrelaxed projection is very inexpensive to estimate, since its computational cost is linear with respect to the dimension of data μ_k. In Figure 5, we display the function $\varphi_\omega(x)$. Notice that for the Sinkhorn–Knopp algorithm, which corresponds to $\omega = 1$, the function φ_ω is always nonnegative. For other values $1 \leq \omega < 2$, it is nonnegative for x close to one.

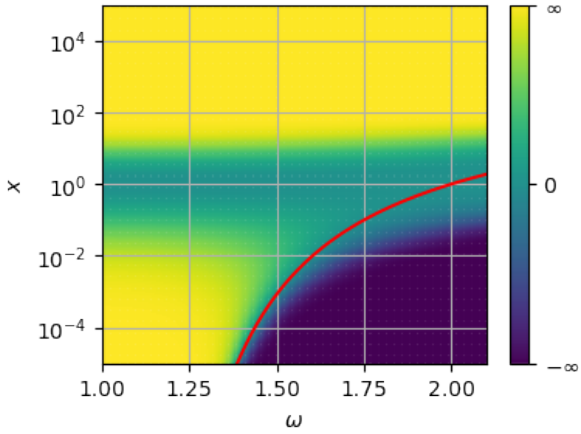


Figure 5: Value of $\varphi_\omega(x)$. The function is positive above the red line, negative below. For any relaxation parameter ω smaller than two, there exists a neighborhood of one on which $\varphi_\omega(\cdot)$ is positive.

4.3 Proposed Algorithm

We first give a general convergence result that later serves as a basis to design an explicit algorithm.

Theorem 1. *Let Θ_1 and Θ_2 be two continuous functions of γ such that:*

$$\forall \gamma \in \mathbb{R}_{+*}^{n_1 n_2}, \quad F(P_{\mathcal{C}_k}^{\Theta_k(\gamma)}(\gamma)) \leq F(\gamma), \tag{26}$$

where the inequality is strict whenever $\gamma \notin \mathcal{C}_k$. Consider the sequence defined by $\gamma^0 = e^{-c/\varepsilon}$ and:

$$\begin{aligned} \tilde{\gamma}^{\ell+1} &= P_{\mathcal{C}_1}^{\Theta_1(\gamma^\ell)}(\gamma^\ell) \\ \gamma^{\ell+1} &= P_{\mathcal{C}_2}^{\Theta_2(\tilde{\gamma}^{\ell+1})}(\tilde{\gamma}^{\ell+1}). \end{aligned}$$

Then, the sequence (γ^ℓ) converges to γ^*.

Lemma 2. *Let us take γ^0 in $\mathbb{R}_{+*}^{n_1 n_2}$, and denote:*

$$S = \left\{ \operatorname{diag}(u)\gamma^0 \operatorname{diag}(v), \quad (u,v) \in \mathbb{R}_{+*}^{n_1+n_2} \right\}$$

the set of matrices that are diagonally similar to γ^0. Then, the set $S \cap \mathcal{C}_1 \cap \mathcal{C}_2$ contains exactly one element $\gamma^* = P_{\mathcal{C}_1 \cap \mathcal{C}_2}(\gamma^0)$.

Proof. We refer to [10] for a proof of this lemma. □

Proof of the Theorem. First of all, notice that the operators $P_{\mathcal{C}_k}^{\theta}$ apply a scaling to lines or columns of matrices. All (γ^ℓ) are thus diagonally similar to γ^0:

$$\forall \ell \geq 0, \quad \gamma^\ell \in S$$

By construction of the functions Θ_k, the sequence of values of the Lyapunov function $(F(\gamma^\ell))$ is non-increasing. Hence, (γ^ℓ) is precompact. If ζ is a cluster point of (γ^ℓ), let us define:

$$\begin{aligned} \tilde{\zeta} &= P_{\mathcal{C}_1}^{\Theta_1(\zeta)}(\zeta) \\ \zeta' &= P_{\mathcal{C}_2}^{\Theta_2(\tilde{\zeta})}(\tilde{\zeta}). \end{aligned}$$

Then, by the continuity of the applications, $F(\zeta) = F(\tilde{\zeta}) = F(\zeta')$. From the hypothesis made on Θ_1 and Θ_2, it can be deduced that ζ is in $\mathcal{C}_1$ and is thus a fixed point of $P_{\mathcal{C}_1}$, while $\tilde{\zeta}$ is in $\mathcal{C}_2$. Therefore, $\zeta' = \tilde{\zeta} = \zeta$ is in the intersection $\mathcal{C}_1 \cap \mathcal{C}_2$. By Lemma 2, $\zeta = \gamma^*$, and the whole sequence (γ^ℓ) converges to the solution. □

We can construct explicitly functions Θ_k as needed in Theorem 1 using the following lemma.

Lemma 3. *Let $1 \leq \omega < \theta$. Then, for any $\gamma \in \mathbb{R}_{+*}^{n_1 n_2}$, one has:*

$$F(P_{\mathcal{C}_k}^{\omega}(\gamma)) \leq F(P_{\mathcal{C}_k}^{\theta}(\gamma)). \tag{27}$$

Moreover, equality occurs if and only if $\gamma \in \mathcal{C}_k$.

Proof. Thanks to Lemma 1, one knows that:

$$F(P_{\mathcal{C}_k}^{\omega}(\gamma)) - F(P_{\mathcal{C}_k}^{\theta}(\gamma)) = \langle \mu_k, (\varphi_\theta - \varphi_\omega)\left((A_k\gamma) \oslash \mu_k\right)\rangle .$$

The function that maps $t \in [1, \infty)$ to $\varphi_t(x)$ is non-increasing since $\partial_t \varphi_t(x) = (x^{1-t} - 1)\log x$. Moreover, for $x \neq 1$, it is strictly decreasing. Thus, Inequality (27) is valid, with equality iff $A_k\gamma = \mu_k$. □

We now argue that a good choice for the functions Θ_k may be constructed as follows. Pick a target parameter $\theta_0 \in [1; 2)$, which will act as an upper bound for the overrelaxation parameter ω, and a small security distance $\delta > 0$. Define the functions Θ^* and Θ as:

$$\Theta^*(w) = \sup\left\{\omega \in [1; 2] \mid \varphi_\omega\left(\min w\right) \geq 0\right\}, \tag{28}$$

$$\Theta(w) = \min(\max(1, \Theta^*(w) - \delta), \theta_0), \tag{29}$$

where $\min w$ denotes the smallest coordinate of the vector w.

Proposition 1. *The function:*

$$\Theta_k(\gamma) = \Theta\left((A_k\gamma) \oslash \mu_k\right) \tag{30}$$

is continuous and verifies the descent condition (26).

Proof. Looking at Figure 5 can help understand this proof. Since the partial derivative of $\partial_\omega \varphi_\omega(x)$ is nonzero for any $x < 1$, the implicit function theorem proves the continuity of Θ^*. The function $\Theta^*\left((A_k\gamma) \oslash \mu_k)\right)$ is such that every term in Relation (24) is non-negative. Therefore, by Lemma 3, using this parameter minus δ ensures the strong decrease (26) of the Lyapunov function. Constraining the parameter to $[1, \theta_0]$ preserves this property. □

This construction, which is often an excellent choice in practice, has several advantages:

- it allows choosing arbitrarily the parameter θ_0 that will be used eventually when the algorithm is close to convergence (we motivate what are good choices for θ_0 in Section 4.4);
- it is also an easy approach to having an adaptive method, as the approximation of Θ^* has a negligible cost (it only requires solving a one-dimensional problem that depends on the smallest value of $(A_k\gamma) \oslash \mu_k$, which can be done in a few iterations of Newton's method).

The resulting algorithm, which is proven to be convergent by Theorem 1, is written in pseudocode in Algorithm 2. The implementation in the log domain is also given in Algorithm 3. Both processes use the function Θ defined implicitly in (29). The evaluation of Θ is approximated in practice with a few iterations of Newton's method on the function $\omega \mapsto \varphi_\omega(\min u)$, which is decreasing, as can be seen in Figure 5. With the choice $\theta_0 = 1$, one recovers exactly the original SK algorithm.

Algorithm 2 Overrelaxed SK Algorithm (SK-SOR)

Require: $\mu_1 \in \mathbb{R}^{n_1}$, $\mu_2 \in \mathbb{R}^{n_2}$, $c \in \mathbb{R}_+^{n_1 \times n_2}$
  Set $u = \mathbf{1}_{n_1}$, $v = \mathbf{1}_{n_2}$, $\gamma^0 = e^{-c/\varepsilon}$, $\theta_0 \in [1; 2)$ and $\eta > 0$
  while $||u \otimes \gamma^0 v - \mu_1|| > \eta$ **do**
    $\tilde{u} = \mu_1 \oslash (\gamma^0 v)$,
    $\omega = \Theta(u \oslash \tilde{u})$
    $u = u^{1-\omega} \otimes \tilde{u}^{\omega}$
    $\tilde{v} = \mu_2 \oslash ({}^t\gamma^0 u)$
    $\omega = \Theta(v \oslash \tilde{v})$
    $v = v^{1-\omega} \otimes \tilde{v}^{\omega}$
  end while
  return $\gamma = \mathrm{diag}(u)\gamma^0\,\mathrm{diag}(v)$

Algorithm 3 Overrelaxed SK Algorithm (SK-SOR) in the Log Domain

Require: $\mu_1 \in \mathbb{R}^{n_1}$, $\mu_2 \in \mathbb{R}^{n_2}$, $c \in \mathbb{R}_+^{n_1 \times n_2}$
  Set $\alpha = \mathbf{0}_{n_1}$, $\beta = \mathbf{0}_{n_2}$, $\gamma^0 = e^{-c/\varepsilon}$, $\theta_0 \in [1; 2)$ and $\eta > 0$
  while $\| \exp(\alpha/\varepsilon) \otimes (\gamma^0 \exp(\beta/\varepsilon)) - \mu_1 \| > \eta$ **do**
    $r_i = \sum_j \exp((-c_{i,j} + \alpha_i + \beta_j)/\varepsilon - \log(\mu_1)_i)$, $\quad i = 1 \cdots n_1$
    $\tilde{\alpha} = \alpha - \varepsilon \log r$
    $\omega = \Theta(r)$
    $\alpha = (1-\omega)\alpha + \omega\tilde{\alpha}$
    $s_j = \sum_i \exp((-c_{i,j} + \alpha_i + \beta_j)/\varepsilon - \log(\mu_2)_j)$, $\quad j = 1 \cdots n_2$
    $\tilde{\beta} = \beta - \varepsilon \log s$
    $\omega = \Theta(s)$
    $\beta = (1-\omega)\beta + \omega\tilde{\beta}$
  end while
  return $\gamma = \mathrm{diag}(\exp(\alpha/\varepsilon))\gamma^0 \,\mathrm{diag}(\exp(\beta/\varepsilon))$

4.4 Acceleration of the Local Convergence Rate

In order to justify the acceleration of convergence that is observed in practice, we now study the local convergence rate of the overrelaxed algorithm, which follows from the classical convergence analysis of the linear SOR method. Our result involves the second largest eigenvalue of the matrix:

$$M_1 = \mathrm{diag}(1 \oslash \mu_1)\, \gamma^* \,\mathrm{diag}(1 \oslash \mu_2)\, {}^t\gamma^* \tag{31}$$

where γ^* is the solution to the regularized OT problem (the largest eigenvalue is one, associated with the eigenvector $\mathbf{1}$). We denote the second largest eigenvalue by $1-\eta$; it satisfies $\eta > 0$ [16].

Proposition 2. *The SK algorithm converges locally at a linear rate $1-\eta$. For the optimal choice of extrapolation parameter $\theta^* = 2/(1+\sqrt{\eta})$, algorithm SK-SOR converges locally linearly at a rate $(1-\sqrt{\eta})/(1+\sqrt{\eta})$. The local convergence of SK-SOR is guaranteed for $\theta \in]0, 2[$, and the linear rate is given in Figure 6 as a function of $1-\eta$ and θ.*

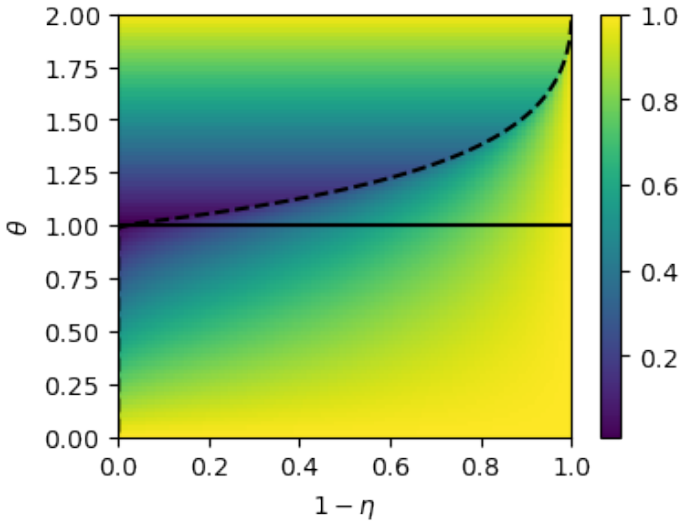


Figure 6: Local linear rate of convergence of the SK-SOR algorithm as a function of $1-\eta$, the local convergence rate of the SK algorithm, and θ the overrelaxation parameter. (Plain curve) The original rate is recovered for $\theta = 1$. (Dashed curve) Optimal overrelaxation parameter θ^*.

Proof. In this proof, we focus on the dual problem, and we recall the relationship $\gamma^\ell = e^{\alpha^\ell/\varepsilon}\gamma^0 e^{\beta^\ell/\varepsilon}$ between the iterates of the overrelaxed projection algorithm γ^ℓ and the iterates $(\alpha^\ell, \beta^\ell)$ of the SOR algorithm on the

dual problem (21), initialized with $(\alpha^0, \beta^0) = (0, 0)$. The dual problem (6) is invariant by translations of the form $(\alpha, \beta) \mapsto (\alpha - k, \beta + k)$, $k \in \mathbb{R}$, but is strictly convex up to this invariance. In order to deal with this invariance, consider the subspace S of pairs of dual variables (α, β) that satisfy $\sum \alpha = \sum \beta$; let π_S be the orthogonal projection on S of kernel $(\mathbf{1}, -\mathbf{1})$, and let $(\alpha^*, \beta^*) \in S$ be the unique dual maximizer in S.

Since one SK-SOR iteration is a smooth map, the local convergence properties of the SK-SOR algorithm are characterized by the local convergence of its linearization, which here corresponds to the SOR method applied to the maximization of the quadratic Taylor expansion of the dual objective E at (α^*, β^*). This defines an affine map $M_\theta : (\alpha^\ell, \beta^\ell) \mapsto (\alpha^{\ell+1}, \beta^{\ell+1})$ whose spectral properties are well known [8, 35] (see also [6] (Chapter 4) for the specific case of convex minimization and [14] for the non-strictly convex case). For the case $\theta = 1$, this is the matrix M_1 defined in Equation (31). The operator norm of $\pi_S \circ M_1$ is smaller than $1 - \eta$, so the operator $(\pi_S \circ M_1)^\ell = \pi_S \circ M_1^\ell$ converges at the linear rate $1 - \eta$ towards zero (observe that by construction, π_S and M_1 are co-diagonalizable and thus commute): for any $(\alpha, \beta) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$, it holds $\|\pi_S \circ M_1^\ell(\alpha - \alpha^*, \beta - \beta^*)\|_2 \leq \|\pi_S(\alpha - \alpha^*, \beta - \beta^*)\|_2 (1 - \eta)^\ell$. More generally, the convergence of $\pi_S \circ M_\theta^\ell$ is guaranteed for $\theta \in]0, 2[$, with the linear rate:

$$f(\theta, \eta) = \begin{cases} \theta - 1 & \text{if } \theta > \theta^* \\ \frac{1}{2}\theta^2(1 - \eta) - (\theta - 1) + \frac{1}{2}\sqrt{(1 - \eta)\theta^2(\theta^2(1 - \eta) - 4(\theta - 1))} & \text{otherwise.} \end{cases} \tag{32}$$

This function is minimized with $\theta^* := 2/(1 + \sqrt{\eta})$, which satisfies $f(\theta^*, \eta) = (1 - \sqrt{\eta})/(1 + \sqrt{\eta})$. The function f is plotted in Figure 6.

To switch from these dual convergence results to primal convergence results, we remark that $\gamma^\ell \to \gamma^*$ implies $\mathrm{KL}(\gamma^\ell, \gamma^0) \to \mathrm{KL}(\gamma^*, \gamma^0)$, which in turn implies $E(\alpha^\ell, \beta^\ell) \to \max E$, so by invoking the partial strict concavity of E, we have $\pi_S(\alpha^\ell, \beta^\ell) \to (\alpha^*, \beta^*)$. The converse implication is direct, so it holds $[\pi_S(\alpha^\ell, \beta^\ell) \to (\alpha^*, \beta^*)] \Leftrightarrow [\gamma^\ell \to \gamma^*]$. We conclude by noting the fact that $\pi_S(\alpha^\ell, \beta^\ell)$ converges at a linear rate, which implies the same rate on γ^ℓ, thanks to the relationship between the iterates. □

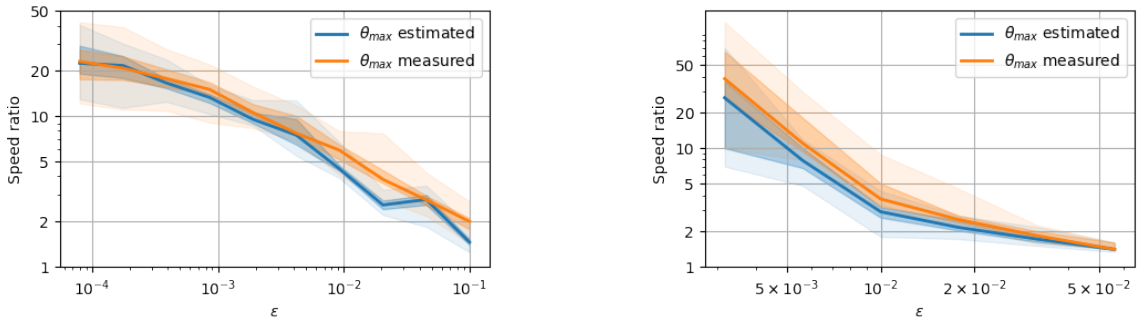
Corollary 1. *The previous local convergence analysis applies to Algorithm 3 with Θ defined as in Equation* (29)*, and the local convergence rate is given by the function of Equation* (32) *evaluated at the target extrapolation parameter θ_0.*

Proof. What we need to show is that eventually, one always has $\Theta(\gamma^\ell) = \theta_0$. This can be seen from the quadratic Taylor expansion $\varphi_{\theta_0}(1 + z) = z^2(\theta_0 - {\theta_0}^2/2) + o(z^2)$, which shows that for any choice of $\theta_0 \in]1, 2[$, there is a neighborhood of one on which $\varphi_{\theta_0}(\cdot)$ is nonnegative. □

5 Experimental Results

We compared Algorithm 2 to the SK algorithm on two very different optimal transport settings. In the setting of Figure 7(a), we considered the domain $[0, 1]$ discretized into 100 samples and the squared Euclidean transport cost on this domain. The marginals were densities made of the sum of a base plateau of height 0.1 and another plateau of height and boundaries chosen uniformly in $[0, 1]$, subsequently normalized. In the setting of Figure 7(b), the cost was a 100×100 random matrix with entries uniform in $[0, 1]$, and the marginals were uniform.

Given an estimation of $1 - \eta$, the local convergence rate of the SK algorithm, we define θ_0 as the optimal parameter as given in Proposition 2. For estimating η, we followed two strategies. For strategy "estimated" (in blue on Figure 7), η was measured by looking at the local convergence rate of the SK algorithm run on another random problem of the same setting and for the same value of ε. For strategy "measured" (in orange on Figure 7), the parameter was set using the local convergence rate of the SK algorithm run on the same problem. Of course, the latter was an unrealistic strategy, but it was interesting to see in our experiments that the "estimated" strategy performed almost as well as the "measured" one, as shown in Figure 7.



(a) Quadratic cost, random marginals.

(b) Random cost, uniform marginals.

Figure 7: Speed ratio between the SK algorithm and its accelerated SK-SOR version Algorithm 2 w.r.t. parameter ε.

Figure 7 displays the ratio of the number of iterations required to reach a precision of 10^{-6} on the dual variable α for the SK algorithm and Algorithm 2. It is worth noting that the complexity per iteration of these algorithms is the same modulo negligible terms, so this ratio is also the runtime ratio (our algorithm can also be parallelized on GPUs just as the SK algorithm). In both experimental settings, for low values of the regularization parameter ε, the acceleration ratio was above 20 with Algorithm 2.

6 Conclusions and Perspectives

The SK algorithm is widely used to solve entropy-regularized OT. In this paper, we first showed that RNA methods are adapted to the numerical acceleration of the SK algorithm. Nevertheless, the global convergence of such approaches is not guaranteed.

Next, we demonstrated that the use of successive overrelaxed projections is a natural and simple idea to ensure and accelerate the convergence, while keeping many nice properties of the SK algorithm (first order, parallelizable, simple). We proposed an algorithm (SK-SOR) that adaptively chooses the overrelaxation parameter so as to guarantee global convergence. The acceleration of the convergence speed was numerically impressive, in particular in low regularization regimes. It was theoretically supported in the local regime by the standard analysis of SOR iterations. Nevertheless, the SK-SOR algorithm was not as performant as RNA, and no guarantee was given on the global computational complexity of either algorithm.

This idea of overrelaxation can be generalized to solve more general problems such as multi-marginal OT, barycenters, gradient flows, and unbalanced OT [6] (Chapter 4), but there is no systematic way to derive globally convergent algorithms. Our work is a step in the direction of building and understanding the properties of robust first-order algorithms for solving OT. More understanding is needed regarding SOR itself (global convergence speed, choice of θ_0), as well as its relation to other acceleration methods [29, 2].

References

[1] M. Z. Alaya, M. Berar, G. Gasso, and A. Rakotomamonjy. Screening sinkhorn algorithm for regularized optimal transport. *arXiv preprint arXiv:1906.08540*, 2019.

[2] J. Altschuler, J. Weed, and P. Rigollet. Near-linear time approximation algorithms for optimal transport via sinkhorn iteration. *arXiv preprint arXiv:1705.09634*, 2017.

[3] D. G. Anderson. Iterative procedures for nonlinear integral equations. *Journal of the ACM (JACM)*, 12(4):547–560, 1965.

[4] J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré. Iterative Bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

[5] L. 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*, 7(3):200–217, 1967.

[6] L. Chizat. *Unbalanced optimal transport: models, numerical methods, applications*. PhD thesis, Université Paris Dauphine, 2017.

[7] L. Chizat, G. Peyré, B. Schmitzer, and F.-X. Vialard. Scaling Algorithms for Unbalanced Transport Problems. *ArXiv e-prints*, arXiv:1607.05816, 2016.

[8] P. Ciarlet. *Introduction à l'analyse numérique matricielle et à l'optimisation*. masson, 1982.

[9] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy. Optimal Transport for Domain Adaptation. *ArXiv e-prints*, arXiv:1507.00504, 2015.

[10] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems (NIPS'13)*, pages 2292–2300, 2013.

[11] A. Dessein, N. Papadakis, and J.-L. Rouas. Regularized optimal transport and the rot mover's distance. *The Journal of Machine Learning Research*, 19(1):590–642, 2018.

[12] C. Frogner, C. Zhang, H. Mobahi, M. Araya-Polo, and T. Poggio. Learning with a Wasserstein Loss. *ArXiv e-prints*, arXiv:1506.05439, 2015.

[13] E. Ghadimi, H. R. Feyzmahdavian, and M. Johansson. Global convergence of the Heavy-ball method for convex optimization. *ArXiv e-prints*, arXiv:1412.7457, 2014.

[14] A. Hadjidimos. On the optimization of the classical iterative schemes for the solution of complex singular linear systems. *SIAM Journal on Algebraic Discrete Methods*, 6(4):555–566, 1985.

[15] F. Iutzeler and J. M. Hendrickx. A generic online acceleration scheme for optimization algorithms via relaxation and inertia. *Optimization Methods and Software*, 34(2):383–405, 2019.

[16] P. A. Knight. The Sinkhorn–Knopp algorithm: convergence and applications. *SIAM Journal on Matrix Analysis and Applications*, 30(1):261–275, 2008.

[17] T. Lehmann, M.-K. von Renesse, A. Sambale, and A. Uschmajew. A note on overrelaxation in the sinkhorn algorithm. *arXiv preprint arXiv:2012.12562*, 2020.

[18] G. Montavon, K.-R. Müller, and M. Cuturi. Wasserstein training of restricted boltzmann machines. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3718–3726, 2016.

[19] P. Ochs. Local Convergence of the Heavy-ball Method and iPiano for Non-convex Optimization. *ArXiv e-prints*, arXiv:1606.09070, 2016.

[20] G. Peyré, L. Chizat, F.-X. Vialard, and J. Solomon. Quantum optimal transport for tensor field processing. *ArXiv e-prints*, arXiv:1612.08731, 2016.

[21] G. Peyré and M. Cuturi. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[22] B. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1 – 17, 1964.

[23] J. Rabin and N. Papadakis. Non-convex relaxation of optimal transport for color transfer between images. In *NIPS Workshop on Optimal Transport for Machine Learning (OTML'14)*, 2014.

[24] L. F. Richardson. Ix. the approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 210(459-470):307–357, 1911.

[25] A. Rolet, M. Cuturi, and G. Peyré. Fast dictionary learning with a smoothed Wasserstein loss. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 51 of *JMLR Workshop and Conference Proceedings*, pages 630–638, 2016.

[26] Y. Rubner, C. Tomasi, and L. J. Guibas. The Earth Mover's Distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[27] M. A. Schmitz, M. Heitz, N. Bonneel, F. Ngole, D. Coeurjolly, M. Cuturi, G. Peyré, and J.-L. Starck. Wasserstein dictionary learning: Optimal transport-based unsupervised nonlinear dictionary learning. *SIAM Journal on Imaging Sciences*, 11(1):643–678, 2018.

[28] B. Schmitzer. Stabilized sparse scaling algorithms for entropy regularized transport problems. *arXiv preprint arXiv:1610.06519*, 2016.

[29] D. Scieur, A. d'Aspremont, and F. Bach. Regularized Nonlinear Acceleration. *ArXiv e-prints*, arXiv:1606.04133, 2016.

[30] D. Scieur, E. Oyallon, A. d'Aspremont, and F. Bach. Online regularized nonlinear acceleration. *arXiv preprint arXiv:1805.09639*, 2018.

[31] V. Seguy and M. Cuturi. Principal geodesic analysis for probability measures under the optimal transport metric. In *Advances in Neural Information Processing Systems*, pages 3312–3320, 2015.

[32] R. Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics*, 35(2):876–879, 1964.

[33] J. Solomon, F. de Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and L. Guibas. Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics*, 34(4):66:1–66:11, 2015.

[34] A. Thibault, L. Chizat, C. Dossal, and N. Papadakis. Overrelaxed sinkhorn-knopp algorithm for regularized optimal transport. *arXiv preprint arXiv:1711.01851*, 2017.

[35] D. M. Young. *Iterative solution of large linear systems*. Elsevier, 2014.

[36] S. K. Zavriev and F. V. Kostyuk. Heavy-ball method in nonconvex optimization problems. *Computational Mathematics and Modeling*, 4(4):336–341, 1993.