Mathematical and Computational Applications

MDPI

Article

A Method for Solving the Monge–Kantorovich Problem Using an Automaton and Wavelet Analysis

Armando Sánchez-Nungaray [1,*], Marcelo Pérez-Medel [2], Carlos González-Flores [3], Raquiel R. López-Martínez [1] and Martín Solís-Pérez [4]

1 Facultad de Matemáticas, Universidad Veracruzana, Xalapa 91000, Mexico; ralopez@uv.mx
2 Facultad de Estudios Superiores Aragón, UNAM, Mexico City 04510, Mexico; marcelo@unam.mx
3 Escuela Superior de Ingeniería Mecánica y Eléctrica Zacatenco, Instituto Politécnico Nacional, Mexico City 07738, Mexico; cfgonzalez@esimez.mx
4 Instituto Tecnológico de Querétaro, Santiago de Querétaro 76000, Mexico; martin.sp@queretaro.tecnm.mx
* Correspondence: armsanchez@uv.mx

Abstract

This article introduces an automaton designed to improve feasible solutions to the Monge–Kantorovich (MK) problem, particularly effective when the cost function is continuous. To enhance its performance, a good initial solution is obtained using the discrete wavelet transform. Specifically, a transportation problem is solved where the cost matrix is composed of the approximation coefficients of the transform, reducing the number of variables to one quarter of the original discrete problem. The solution to this reduced problem is extended using the detail coefficients, yielding a feasible solution to the original problem. This solution serves as the initial state of the tuning automaton, whose final states provide approximations to the optimal solution of the transportation problem.

Keywords: mass transfer problem; wavelets; multiresolution analysis; automaton

1. Introduction

Recognized as one of the foundational formulations in optimal transport theory, the Monge–Kantorovich (MK) problem provides the theoretical framework for a wide range of applications, including image registration, mass redistribution problems, numerical approximation of measures, and modern machine learning models [1–3]. Efficiently solving this problem requires approximating measures and cost functions defined on high-dimensional spaces, which leads to significant computational challenges, particularly when fine discretizations are used or when progressively refined solutions are sought [4–7].

The Monge–Kantorovich problem is formulated as follows. Let X and Y be two compact subsets of $\mathbb{R}$. We denote by $M^+(X \times Y)$ the family of finite measures on $X \times Y$. Given $\mu \in M^+(X \times Y)$, its marginal measures on X and Y are defined by

$$\Pi_1 \mu(E_1) = \mu(E_1 \times Y), \tag{1}$$

and

$$\Pi_2 \mu(E_2) = \mu(X \times E_2), \tag{2}$$

for each μ-measurable set $E_1 \subset X$ and $E_2 \subset Y$. Let c be a real-valued function defined on $X \times Y$, and let η^1 and η^2 be finite measures on X and Y, respectively. The Monge–Kantorovich mass transfer problem is then given by

Check for updates

Academic Editor: Gianluigi Rozza

Received: 6 February 2026
Revised: 14 March 2026
Accepted: 1 April 2026
Published: 9 April 2026

$$
\begin{aligned}
\text{MK: minimize} \quad & \langle \mu, c \rangle := \int c \, d\mu \\
\text{subject to} \quad & \Pi_1 \mu(E_1) = \eta^1(E_1), \quad \Pi_2 \mu(E_2) = \eta^2(E_2), \quad \mu \in M^+(X \times Y),
\end{aligned} \tag{3}
$$

which constitutes a linear optimization problem over the space of positive measures with prescribed marginals [5,6].

Wavelet-based multiresolution analysis (MRA), particularly Haar MRA, provides a natural framework for constructing hierarchical approximations of the MK problem. In this approach, the involved measures and cost functions are projected onto nested approximation spaces, yielding a sequence of discrete problems at increasing resolution levels. This multiscale structure significantly reduces the computational complexity of the problem while preserving its essential features [8–11]. However, even within this framework, obtaining an optimal solution at a fine resolution may still require solving large-scale transportation problems.

Multiresolution schemes based on wavelet decompositions have already been successfully applied to the Monge–Kantorovich problem. In particular, Haar-based approximations have been used to construct discrete versions of the MK problem and to derive efficient numerical schemes for its solution [12–14]. These approaches exploit the hierarchical nature of the wavelet transform to obtain good initial approximations at coarse levels, which are then refined as the resolution increases.

To further enhance computational efficiency, we propose complementing the wavelet-based approximation with an automaton specifically designed to operate on permutations. This automaton uses local cost information to guide transitions between feasible solutions, ensuring that each transition preserves feasibility while improving or maintaining the value of the objective function. The underlying structure of the automaton is closely related to classical assignment and transportation problems [15–17], and its behavior can be interpreted as a metaheuristic refinement strategy [4].

In our approach, the multiresolution analysis provides an initial feasible solution that is close to optimal at each resolution level. Specifically, we apply the discrete wavelet transform to the cost function and solve the corresponding transportation problem whose cost matrix consists of the approximation coefficients. This reduced problem involves only one quarter of the variables of the original discrete problem, leading to a substantial reduction in computational cost. Similar strategies have been shown to be effective in previous studies [12–14].

The solution obtained for the transportation problem at the approximation level is then extended to a feasible solution of the original problem by incorporating the detail coefficients of the wavelet decomposition. This extended solution serves as the initial state of the tuning automaton. The final states reached by the automaton are taken as approximations of the optimal solution of the Monge–Kantorovich problem. The convergence and stability of such approximation procedures are supported by classical results on approximation schemes and convergence of measures [5,6,18].

The content of this article is as follows: In Section 2, we present a series of preliminary results on multiresolution analysis (MRA) and wavelet transforms that will be applied to obtain approximations of the Monge–Kantorovich problem. In Section 3, we present an adjustment automaton for the permutation group. This tool will be fundamental in the development of this article, as it will allow us to improve the solutions to transportation problems. In Section 4, we present the discretizations of the Monge–Kantorovich problem, noting that the feasible solutions are permutations and defining a function on each state of the automaton (permutation and position) that will allow us to ensure that the next state is better than, or at least equal to, the current one. We then present some examples to illustrate how the automaton works, starting from a random permutation. In

Section 5, we present several classic examples of Monge–Kantorovich problems, giving approximations of the solutions based on a methodology for solving transportation problems at each level of multiresolution analysis using the wavelet transform and the tuning automaton for permutations.

2. Preliminaries

This section presents the theoretical foundations underlying the proposed methodology. In particular, it presents multiresolution analysis (MRA) based on Haar wavelets and its extension to the two-dimensional case, along with the properties of measure projection through multiscale approximations.

A multiresolution analysis (MRA) on the real line $\mathbb{R}$ is defined as a sequence $\{V_j\}_{j\in\mathbb{Z}}$ in the function space of $L^2(\mathbb{R})$. These subspaces satisfy the following properties:

(1). $V_j \subset V_{j+1}$, for every $j \in \mathbb{Z}$.
(2). $L^2(\mathbb{R}) = \overline{\bigcup_{j\in\mathbb{Z}} V_j}$.
(3). $\bigcap_{j\in\mathbb{Z}} V_j = \{0\}$.
(4). $V_j = D_{2^j} V_0$.
(5). There exists a function $\varphi \in L^2(\mathbb{R})$, called the scaling function, such that the collection $\{T_j\varphi\}_{j\in\mathbb{Z}}$ forms an orthonormal system of translates and

$$V_0 = \overline{\text{span}}\{T_j\varphi\}_{j\in\mathbb{Z}}, \tag{4}$$

where D_a and T_b denote the dilation and translation operators by the elements a and b, respectively.

Let $\varphi(x) = \chi_{[0,1)}(x)$ and $\psi(x) = \chi_{[0,1/2)}(x) - \chi_{[1/2,1)}(x)$, where χ_A denotes the characteristic function of the set A. For each pair $(j,k) \in \mathbb{Z}\times\mathbb{Z}$, we define the functions:

$$\begin{aligned}\varphi_{j,k}(x) &= (D_{2^j}T_k\varphi)(x) = 2^{j/2}\chi_{I_{j,k}}(x),\\ \psi_{j,k}(x) &= (D_{2^j}T_k\psi)(x) = 2^{j/2}\big(\chi_{I_{j+1,2k}}(x) - \chi_{I_{j+1,2k+1}}(x)\big),\end{aligned}$$

where $I_{j,k} = \big[2^{-j}k, 2^{-j}(k+1)\big)$. Thus, the Haar multiresolution analysis (Haar-MRA) on $\mathbb{R}$ is the sequence of subspaces $\{V_j\}_{j\in\mathbb{Z}}$ generated by the scaling function $\varphi(x)$. Moreover, the set

$$\bigcup_{j\in\mathbb{Z}}\{\psi_{j,k}\}_{k\in\mathbb{Z}}$$

forms a basis of $L^2(\mathbb{R})$ and for each $j \in \mathbb{Z}$, the space V_{j+1} is decomposed as the direct sum $V_j \oplus W_j$, where

$$W_j = \overline{\text{span}}\{\psi_{j,k}\}_{k\in\mathbb{Z}}.$$

The Haar multiresolution analysis on $\mathbb{R}$ can be extended to $\mathbb{R}^2$ by defining the scaling function as $\Phi(x,y) = \varphi(x)\varphi(y)$, along with the Haar wavelet functions on $\mathbb{R}^2$ defined by

$$\Psi^{(1,2)}(x,y) = \varphi(x)\psi(y), \quad \Psi^{(2,1)}(x,y) = \psi(x)\varphi(y), \quad \Psi^{(2,2)}(x,y) = \psi(x)\psi(y).$$

Using these functions, we define the spaces

$$\mathbf{V}_0 = \overline{\text{span}}\{\mathbf{T}_{j,k}\Phi\}_{(j,k)\in\mathbb{Z}^2} \quad \text{and} \quad \mathbf{V}_j = \mathbf{D}_{2^j}\mathbf{V}_0,$$

where $\mathbf{D}_a$ and $\mathbf{T}_{j,k}$ represent the dilatation and translation operators a and (j,k), respectively. These operators are explicitly defined as:

$$\mathbf{D}_a = D_a \otimes D_a \quad \text{and} \quad \mathbf{T}_{j,k} = T_j \otimes T_k.$$

Hence, the sequence $\{\mathbf{V}_j\}_{j\in\mathbb{Z}}$ of subspaces of $L^2(\mathbb{R}^2)$ constitutes the Haar-MRA on $\mathbb{R}^2$. Consequently, we obtain the following decomposition:

$$\mathbf{V}_{j+1} = \mathbf{V}_j \oplus \mathbf{W}_j^{(1,2)} \oplus \mathbf{W}_j^{(2,1)} \oplus \mathbf{W}_j^{(2,2)}, \tag{5}$$

where

$$\mathbf{W}_j^{(1,2)} = V_j \otimes W_j, \quad \mathbf{W}_j^{(2,1)} = W_j \otimes V_j \quad \text{and} \quad \mathbf{W}_j^{(2,2)} = W_j \otimes W_j.$$

The Haar scaling system consists of all functions of the form

$$\Phi_{j,k_1,k_2}(x,y) = \varphi_{j,k_1}(x)\varphi_{j,k_2}(y), \quad j,k_1,k_2 \in \mathbb{Z},$$

while the Haar scaling system is the collection of all functions

$$\Psi^{(1,2)}_{j,k_1,k_2}(x,y) = \varphi_{j,k_1}(x)\psi_{j,k_2}(y), \quad \Psi^{(2,1)}_{j,k_1,k_2}(x,y) = \psi_{j,k_1}(x)\varphi_{j,k_2}(y),$$
$$\Psi^{(2,2)}_{j,k_1,k_2}(x,y) = \psi_{j,k_1}(x)\psi_{j,k_2}(y),$$

for all $j,k_1,k_2 \in \mathbb{Z}$.

In the two-dimensional Haar decomposition, the wavelet system $\Psi^{(1,2)}$, $\Psi^{(2,1)}$, and $\Psi^{(2,2)}$ corresponds to the horizontal, vertical, and diagonal detail components, respectively. Thus, each resolution level produces a low-pass approximation together with exactly three high-pass components.

It is well known that the collection of functions

$$\bigcup_{j_0\in\mathbb{Z}}\bigcup_{j\geq j_0}\left\{\Phi_{j_0,k_1,k_2}, \Psi^{(1,2)}_{j,k_1,k_2}, \Psi^{(2,1)}_{j,k_1,k_2}, \Psi^{(2,2)}_{j,k_1,k_2}\right\}_{k_1,k_2\in\mathbb{Z}}$$

forms an orthonormal basis for $L^2(\mathbb{R}^2)$. Furthermore, the collection

$$\bigcup_{j\in\mathbb{Z}}\left\{\Psi^{(1,2)}_{j,k_1,k_2}, \Psi^{(2,1)}_{j,k_1,k_2}, \Psi^{(2,2)}_{j,k_1,k_2}\right\}_{k_1,k_2\in\mathbb{Z}} \tag{6}$$

also constitutes an orthonormal basis for $L^2(\mathbb{R}^2)$.

Thus, the approximation operator at the level $j \in \mathbb{Z}$ is defined as

$$(\mathbf{P}_j f)(x,y) = \sum_{k_1,k_2\in\mathbb{Z}} \langle f, \Phi_{j,k_1,k_2}\rangle \Phi_{j,k_1,k_2}(x,y), \tag{7}$$

for all $f \in L^2(\mathbb{R}^2)$. The detail operators at this level are defined as

$$(\mathbf{Q}^i_j f)(x,y) = \sum_{k_1,k_2\in\mathbb{Z}} \langle f, \Psi^i_{j,k_1,k_2}\rangle \Psi^i_{j,k_1,k_2}(x,y), \tag{8}$$

with $i \in \{(1,2),(2,1),(2,2)\}$. Note that the projection satisfies the following decomposition

$$\mathbf{P}_{j+1} = \mathbf{P}_j + \mathbf{Q}_j^{(1,2)} + \mathbf{Q}_j^{(2,1)} + \mathbf{Q}_j^{(2,2)}.$$

We now describe the approximation operator $\mathbf{P}_j$ and detail operators $\mathbf{Q}_j^{(1,2)}$, $\mathbf{Q}_j^{(2,1)}$ and $\mathbf{Q}_j^{(2,2)}$ from the geometric point of view. Given that $j,k_1,k_2 \in \mathbb{Z}$ fixed, we consider the square defined by

$$S(j,k_1,k_2) = I_{j,k_1} \times I_{j,k_2} = \left[\frac{k_1}{2^j}, \frac{k_1+1}{2^j}\right) \times \left[\frac{k_2}{2^j}, \frac{k_2+1}{2^j}\right). \tag{9}$$

This square can be decomposed as a disjoint union of the following squares:

$$\begin{aligned} S_{11}(j,k_1,k_2) &= S(j+1,2k_1,2k_2), \\ S_{12}(j,k_1,k_2) &= S(j+1,2k_1,2k_2+1), \\ S_{21}(j,k_1,k_2) &= S(j+1,2k_1+1,2k_2), \\ S_{22}(j,k_1,k_2) &= S(j+1,2k_1+1,2k_2+1). \end{aligned}$$

Thus, we have that

$$\begin{aligned} \Phi_{j,k_1,k_2} &= 2^j \chi_{S_{11}}, & \Psi^{(1,2)}_{j,k_1,k_2} &= 2^j \left(\chi_{S_{11}\cup S_{21}} - \chi_{S_{12}\cup S_{22}}\right), \\ \Psi^{(2,1)}_{j,k_1,k_2} &= 2^j \left(\chi_{S_{11}\cup S_{12}} - \chi_{S_{21}\cup S_{22}}\right), & \Psi^{(2,2)}_{j,k_1,k_2} &= 2^j \left(\chi_{S_{11}\cup S_{22}} - \chi_{S_{12}\cup S_{21}}\right), \end{aligned}$$

where the sets S_{ij}, with $i,i = 1,2$, are illustred in Figure 1.

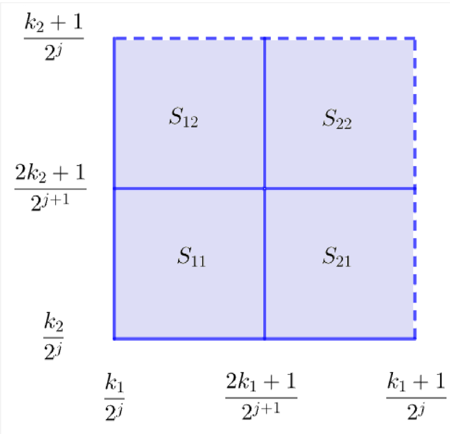


Figure 1. The square S is expressed as the disjoint union of the four sub-squares S_{11}, S_{12}, S_{21} and S_{22}.

Consequently, for $f \in L^2(\mathbb{R}^2)$ the operator $\mathbf{P}_j f$ acts as a discretization of f that is constant over the disjoint squares $S(j,k_1,k_2)$. For more details on the fundamental results in multiresolution analysis, see the following works: [9–11,13].

Additionally, let X be a compact subset of $\mathbb{R}^2$. We assume that μ is a measure absolutely continuous with respect to the Lebesgue measure λ, whose Radon-Nikodym derivative f_μ satisfies $f_\mu \in L^1(X) \cap L^2(X)$.

It is well known that $\|\mathbf{P}_j f_\mu - f_\mu\|_2 \to 0$ as $j \to \infty$. Moreover, by the compactness of X, we have the estimate

$$\|\mathbf{P}_j f_\mu - f_\mu\|_1 \leq \lambda(X)^{1/2} \|\mathbf{P}_j f_\mu - f_\mu\|_2.$$

Therefore, the approximation μ_j of the measure μ in the level j defined by $d\mu_j = \mathbf{P}_j f_\mu \, d\lambda$ is absolutely continuous with respect to the Lebesgue measure λ. If we further assume that the measure μ has support contained in X, then the approximations μ_j converge to μ in both the L^1 and L^2 sense.

Note that for each μ_j, there exists an integrable function $\hat{\mu}_j \in \mathbf{P}_j L^2(\mathbb{R}^2)$ such that $\mu_j(A) = \mathbb{E}[\hat{\mu}_j \chi_A]$ for every Lebesgue measurable set $A \subset \mathbb{R}^2$, where $\mathbb{E}[g]$ denotes the expectation of g with respect to the Lebesgue measure.

Finally, the fact that each the measure μ_j has compact support implies that the sequence $\{\mu_j\}_{j\in\mathbb{Z}}$ converges weakly to the measure μ. For further details on the approximation of a measure using multiresolution analysis, see [12,13].

3. Adjustment Automaton over Permutations

This section introduces a finite automaton defined on the permutation group S_n, designed to refine discretized feasible solutions to the Monge–Kantorovich (MK) problem. The automaton serves as a central tool in the subsequent section, where it enables systematic local adjustments that reduce the associated cost and thereby improves solution quality.

An automaton is a theoretical model of computation used to describe and analyze systems that process inputs and produce outputs according to a prescribed set of states and transition rules. Formally, an automaton is defined as follows:

Definition 1. *An automaton is defined as a 5-tuple:*

$$A = (Q, \Sigma, \delta, q_0, F),$$

where:

- *Q is a finite set of **states**,*
- *Σ is a finite set of input symbols called the **alphabet**,*
- *$\delta : Q \times \Sigma \to Q$ is the **transition function**,*
- *$q_0 \in Q$ is the **initial state**,*
- *$F \subseteq Q$ is the set of **accepting (or final) states**.*

Explanation of the Components:

1. **States (Q).** At any given time, the automaton occupies one of a finite number of states. These states encode the internal configuration of the system and may be interpreted as its "memory."
2. **Alphabet (Σ).** The input alphabet is a finite set of symbols that the automaton is capable of reading. For instance, in the case of binary inputs, one has $\Sigma = \{0, 1\}$.
3. **Transition Function (δ).** The transition function specifies how the automaton evolves from one state to another in response to an input symbol. Formally, it is a mapping
$$\delta : Q \times \Sigma \to Q.$$
Given a state $q \in Q$ and an input symbol $a \in \Sigma$, the value $\delta(q, a)$ determines the subsequent state of the automaton.
4. **Initial State (q_0).** Computation begins at the distinguished initial state $q_0 \in Q$, from which the automaton starts processing the input.
5. **Accepting States (F).** A subset $F \subseteq Q$ is designated as the set of accepting (or final) states. An input string is said to be **accepted** by the automaton if, after the entire input has been processed, the automaton terminates in a state belonging to F.

Further information on the automaton is provided in [19,20]. We now recall the following concept:

Definition 2. *Let $n \in \mathbb{N}$ with $n \neq 0$, and let $\mathbf{n} = \{1, \ldots, n\}$. A **permutation** on n is an invertible function $\sigma : \mathbf{n} \to \mathbf{n}$. The set of all permutations on n, together with function composition, forms a group called the **symmetric group** of degree n, which is denoted by S_n.*

A permutation $\sigma \in S_n$ can be represented as an array (or list) of length n, where the i-th entry specifies the image of i under σ. Formally, a permutation σ is represented as:

$$\sigma = \begin{bmatrix} \sigma(1) & \sigma(2) & \cdots & \sigma(n) \end{bmatrix}.$$

The inverse of a permutation $\sigma = \begin{bmatrix} \sigma(1) & \sigma(2) & \cdots & \sigma(n) \end{bmatrix}$ is the permutation σ^{-1} such that:

$$\sigma^{-1}(\sigma(i)) = i, \quad \text{for all } i \in \{1, 2, \ldots, n\}.$$

To compute σ^{-1} from the array representation of σ, one swaps the indices and the corresponding values. A detailed discussion of permutations can be found in [21].

Example 1. *Let* $\sigma = \begin{bmatrix} 4 & 3 & 1 & 2 \end{bmatrix} \in S_4$. *Then its inverse permutation is given by*

$$\sigma^{-1} = \begin{bmatrix} 3 & 4 & 2 & 1 \end{bmatrix}.$$

We can represent the permutation σ by the matrix $A_\sigma = (a^\sigma_{i,j})^n_{i,j=1}$, defined by

$$a^\sigma_{i,j} = \begin{cases} 1, & \text{if } j = \sigma(i), \quad i = 1, \ldots, n, \\ 0, & \text{otherwise}. \end{cases} \tag{10}$$

Example 2. *Let* $\sigma = \begin{bmatrix} 3 & 1 & 2 \end{bmatrix} \in S_3$. *Its matrix representation is given by*

$$A_\sigma = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Observe that for each fixed permutation $\sigma \in S_n$ and fixed index k, the components of σ associated with the position k take the following form.

$$\begin{array}{cccccccc} \text{values} & \sigma(k)-1 & \cdots & \sigma(k-1) & \sigma(k) & \sigma(k+1) & \cdots & \sigma(k)+1 \\ \text{positions} & \sigma^{-1}(\sigma(k)-1) & \cdots & k-1 & k & k+1 & \cdots & \sigma^{-1}(\sigma(k)+1) \end{array}$$

The second row indicates the position in the array associated with the permutation σ, while k denotes the position at which the automaton acts.

Example 3. *To exemplify the above paragraph, we consider the following permutation with respect to position* $k = 5$.

$$\begin{array}{cc} \sigma = & (4,6,8,7,2,5,3,1) \\ \textit{positions} & 1,2,3,4,5,6,7,8 \end{array}$$

The inverse permutation is given by

$$\begin{array}{cc} \sigma^{-1} = & (8,5,7,1,6,2,4,3) \\ \textit{positions} & 1,2,3,4,5,6,7,8 \end{array}$$

Now, calculate the distinguished position with respect to $k = 5$, *which are* $4, 5, 6, 7, 8$ *since*

$$\sigma^{-1}(\sigma(k) + 1) = \sigma^{-1}(\sigma(5) - 1) = \sigma^{-1}(2 - 1) = 8,$$

and

$$\sigma^{-1}(\sigma(k) + 1) = \sigma^{-1}(2 + 1) = 7.$$

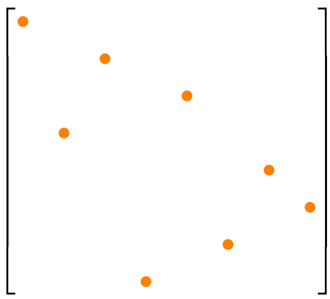
Finnally, the matrix representation is given by

$$A_\sigma = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Remark 1. *In the figures presented in Section 5, the graphs representing the permutations (feasible solutions) are numbered from bottom to top and from left to right. Moreover, the positions of the* 1*'s are represented by colored points. In contrast, in the previous matrix representation, the numbering is carried out from top to bottom and from right to left. For example, considering the row reordering used in Section 5, the previous matrix corresponds to the following arrangement.*

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Finally, the representation of the permutation using colored points is as follows:



The neighbors' interpretation is as follows. The neighbors of $(k, \sigma(k))$ in the matrix representation are those positions that contain the value 1 in the rows and columns adjacent to position $(k, \sigma(k))$, since σ is fixed, we will refer to the position $(k, \sigma(k))$ simply as position k. Explicitly, these positions are:

$$\begin{aligned} \left(\sigma^{-1}(\sigma(k)-1), \sigma(k)-1\right), \quad & \left(k-1, \sigma(k-1)\right), \\ \left(k+1, \sigma(k+1)\right), \quad & \left(\sigma^{-1}(\sigma(k)+1), \sigma(k)+1\right) \end{aligned}$$

For simplicity, we will write

$$\sigma^{-1}(\sigma(k)-1), k-1, k+1, \sigma^{-1}(\sigma(k)+1)$$

to denote the neighbors of position k, which are illustrated in Figure 2.

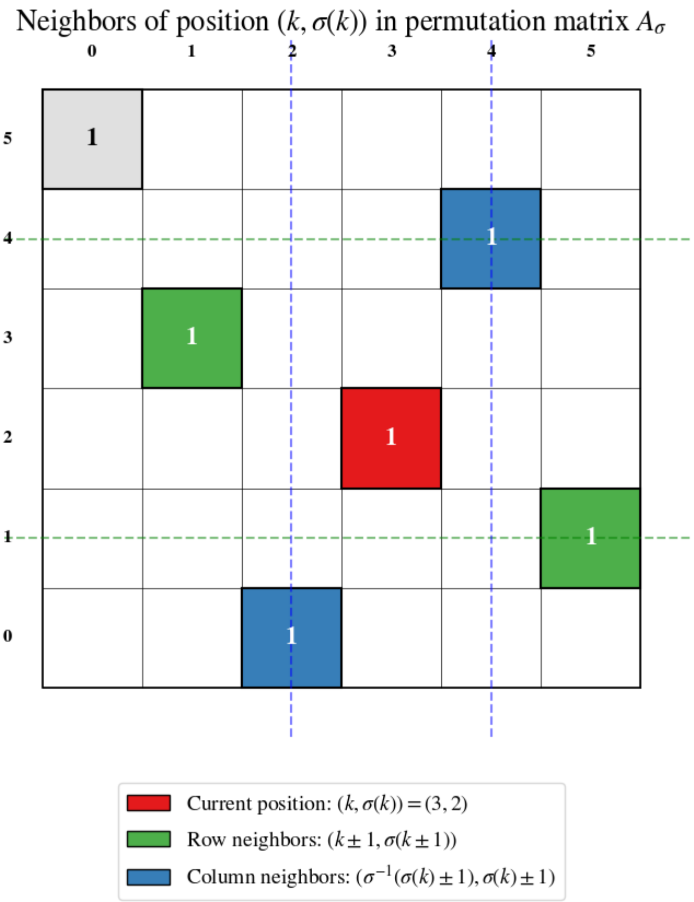


Figure 2. Neighbors of position $(k, \sigma(k))$ in the permutation matrix σ.

Definition 3. *We define an adjusted automaton constructed from the elements of S_n in the following manner: we consider the* 5*-tuple*

$$A = (Q, \Sigma, \delta, q_0, F), \tag{11}$$

where:

- $Q = S_n \times \{1, \ldots, n+1\}$ *is a finite set of* ***states****,*
- $\Sigma = \{0, 1, 2, 3, 4\}$ *is a finite set of input symbols called the* ***alphabet****,*
- $\delta : Q \times \Sigma \to Q$ *is the* ***transition function****, which is defined by*

$$\begin{aligned}
\delta((\sigma, k), 0) &= (\sigma, k+1) \\
\delta((\sigma, k), 1) &= (\sigma_1, k+1) \\
\delta((\sigma, k), 2) &= (\sigma_2, k+1) \\
\delta((\sigma, k), 3) &= (\sigma_3, k+1) \\
\delta((\sigma, k), 4) &= (\sigma_4, k+1)
\end{aligned}$$

where

$$\begin{aligned}
\sigma_1 &= [\sigma(1), \ldots \sigma(k) - 1, \ldots, \sigma(k), \sigma(k-1), \sigma(k+1), \ldots, \sigma(k) + 1, \ldots, \sigma(n)] \\
\sigma_2 &= [\sigma(1), \ldots, \sigma(k) - 1, \ldots, \sigma(k-1), \sigma(k+1), \sigma(k) \ldots, \sigma(k) + 1, \ldots, \sigma(n)] \\
\sigma_3 &= [\sigma(1), \ldots, \sigma(k), \ldots \sigma(k-1), \sigma(k) - 1, \sigma(k+1), \ldots, \sigma(k) + 1, \ldots, \sigma(n)] \\
\sigma_4 &= [\sigma(1), \ldots, \sigma(k) - 1, \ldots \sigma(k-1), \sigma(k) + 1, \sigma(k+1), \ldots, \sigma(k), \ldots, \sigma(n)]
\end{aligned}$$

- $q_0 = (\sigma_0, 1) \in Q$ *is the* ***initial state****, where* σ_0 *is any element of* S_n.
- $F = S_n \times \{n+1\} \subseteq Q$ *is the set of* ***accepting (or final) states****.*

4. Improving Solutions of the Monge–Kantorovich Problem via an Adjustment Automaton on the Permutation Group

In this section, we present a hybrid methodology that integrates Haar wavelet-based multiresolution analysis (MRA) with a finite automaton acting on permutations, with the aim of obtaining optimal solutions to the Monge–Kantorovich (MK) problem.

This approach exploits the hierarchical structure of MRA to construct an initial approximate solution, which is subsequently refined through local adjustments performed by the automaton described in the previous section.

M-K Problem and MRA

A measure $\mu \in M^+(X \times Y)$ is said to be a feasible solution to the MK problem if it satisfies (3) and the pairing $\langle \mu, c \rangle$ is finite. The MK problem is said to be solvable if there exists a feasible measure μ^* that attains the optimal value of the objective functional. In this case, μ^* is referred to as an optimal solution of (3).

Moreover, assuming that μ is absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}^2$ and that η_1 and η_2 are absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}$, the MK problem admits a natural discretization via Haar multiresolution analysis on $\mathbb{R}^2$.

Recall that, according to Equation (3), the Monge–Kantorovich problem at level j is given by:

$$\begin{aligned} \text{MK}_j\text{: minimize} \quad & \langle \mu_j, c \rangle := \int c d\mu_j \\ \text{subject to:} \quad & \Pi_1 \mu_j(E_1) = \eta_j^1(E_1), \quad \Pi_2 \mu_j(E_2) = \eta_j^2(E_2), \quad \mu \in M^+(X \times Y), \end{aligned}$$

for each μ-measurable set $E_1 \subset X$ and $E_2 \subset Y$, where μ_j, η_j^1 and η_j^2 are the projections to level j of the measures μ, η^1 and η^2 to the respective Haar MRA.

In particular, we consider the MK problem with cost function $c = c(x, y)$, base sets $X = Y = [0, 1]$, and marginal measures $\eta^1 = \eta^2 = \lambda|_{[0,1]}$. Since practical applications typically involve discretized formulations, the application of multiresolution analysis on $\mathbb{R}^2$ leads us to the following objective problem:

$$\begin{aligned} \text{MK}_j\text{: minimize} \quad & \sum_{i,k} \mu_{i,k}^j c_{i,k}^j \\ \text{subject to:} \quad & \sum_i \mu_{i,k}^j = \frac{1}{2^j}, \text{ for each} \quad k = 1, \ldots, 2^j. \\ & \sum_k \mu_{i,k}^j = \frac{1}{2^j}, \text{ for each} \quad i = 1, \ldots, 2^j. \\ & \sum_{i,k} \mu_{i,k}^j = 1. \\ & \mu_{i,k}^j \geq 0, \text{ for each } i, k = 1, \ldots, 2^j. \end{aligned} \tag{12}$$

Here, $\mu_{i,k}^j$ denotes the portion of the initial mass $\frac{1}{2^j}$ located at the interval $I_{j,i}$ on the x-axis that is allocated to the interval $I_{j,k}$ on the y-axis. We refer to the j-discrete unit square as the grid formed by the squares $S(j, k_1, k_2)$ (see (9)), which partitions the set $[0, 1] \times [0, 1]$ into $2^j \times 2^j$ blocks, each naturally identified with the point (k_1, k_2).

The above formulation is an assignment problem, which is a particular case of the transport problem. The problem instance has a number of agents and a number of tasks.

Any agent can be assigned to perform any task, incurring some cost that may vary depending on the agent-task assignment. The Hungarian method is a combinatorial optimization algorithm that solves the assignment problem in polynomial time and which anticipated later primal–dual methods. It was developed and published in 1955 by Harold Kuhn, who gave it the name "Hungarian method" because the algorithm was largely based on the earlier works of two Hungarian mathematicians, Dénes Kőnig and Jenő Egerváry. For more details, see [15–17].

We assume the existence of a simple solution μ to (12). That is, μ is a feasible solution such that, for any $i_0, k_0 \in \{1, \ldots, 2^j\}$ with $\mu_{i_0,k_0} \neq 0$, it necessarily holds that $\mu_{i_0,k} = 0$ for all $k \neq k_0$ and $\mu_{i,k_0} = 0$ for all $i \neq i_0$.

All feasible solutions of (12) correspond to permutations $\sigma \in S_{2^j}$. Consequently, we employ the automaton introduced in Section 3, with $n = 2^j$ in this setting.

Our objective is to improve a given solution of (12), where $c^j_{i,k}$ denotes the associated cost matrix. Based on this cost matrix and the current state of the automaton, we introduce the following quantities.

$$\begin{aligned}
a_1 &= c^j_{k-1,\sigma(k-1)} + c^j_{k,\sigma(k)} - c^j_{k,\sigma(k-1)} - c^j_{k-1,\sigma(k)} \\
a_2 &= c^j_{k+1,\sigma(k+1)} + c^j_{k,\sigma(k)} - c^j_{k,\sigma(k+1)} - c^j_{k+1,\sigma(k)} \\
a_3 &= c^j_{\sigma^{-1}(\sigma(k)-1),\sigma(k)-1} + c^j_{k,\sigma(k)} - c^j_{k,\sigma(k)-1} - c^j_{\sigma^{-1}(\sigma(k)-1),\sigma(k)} \\
a_4 &= c^j_{\sigma^{-1}(\sigma(k)+1),\sigma(k)+1} + c^j_{k,\sigma(k)} - c^j_{k,\sigma(k)+1} - c^j_{\sigma^{-1}(\sigma(k)+1),\sigma(k)}
\end{aligned} \tag{13}$$

Definition 4. *We define the evaluation function of the state associated with the matrix $c^j_{i,k}$ as follows:*

$$f\big((\sigma,k)\big) = \begin{cases} 0 & \textit{If } a_1 \leq 0, a_2 \leq 0, a_3 \leq 0, a_4 \leq 0, \\ 1 & \textit{If } a_1 \geq 0, a_1 \geq a_2, a_1 \geq a_3, a_1 \geq a_4, \\ 2 & \textit{If } a_2 \geq 0, a_2 \geq a_1, a_2 \geq a_3, a_2 \geq a_4, \\ 3 & \textit{If } a_3 \geq 0, a_3 \geq a_1, a_3 \geq a_2, a_3 \geq a_4, \\ 4 & \textit{If } a_4 \geq 0, a_4 \geq a_1, a_4 \geq a_2, a_4 \geq a_3. \end{cases} \tag{14}$$

where the values a_j for $j = 0, \ldots, 5$ are given by (13).

The interpretation of the evaluation function is the following:

- If $f\big((\sigma,k)\big) = 0$ then any change improves the feasible solution.
- If $f\big((\sigma,k)\big) = 1$ then the change in the positions $k-1$ and k (in the array representation of σ) is the better local change that improves the feasible solution.
- If $f\big((\sigma,k)\big) = 2$ then the change in the positions $k+1$ and k is the better local change that improves the feasible solution.
- If $f\big((\sigma,k)\big) = 3$ then the change in the positions $\sigma^{-1}\big(\sigma(k)-1\big)$ and k is the better local change that improves the feasible solution.
- If $f\big((\sigma,k)\big) = 4$ then the change in the positions $\sigma^{-1}\big(\sigma(k)+1\big)$ and k is the better local change that improves the feasible solution.

By construction, if we consider a state (σ_k, k) and apply the automaton with the alphabet symbol $f(\sigma_k, k)$, we obtain the following.

$$\delta\big((\sigma_k,k), f(\sigma_k,k)\big) = (\sigma_{k+1}, k+1).$$

It is clear that both σ_k and σ_{k+1} are feasible solutions of (12). Moreover, by construction of function f defined in (14), we have

$$\sum_{i,l=1}^{2^{-j}} 2^{-j} a_{i,l}^{\sigma_{k+1}} c_{i,l}^{j} \leq \sum_{i,l=1}^{2^{-j}} 2^{-j} a_{i,l}^{\sigma_k} c_{i,l}^{j},$$

equivalently

$$\sum_{l=1}^{2^{-j}} 2^{-j} c_{l,\sigma_{k+1}(l)}^{j} \leq \sum_{l=1}^{2^{-j}} 2^{-j} c_{l,\sigma_k(l)}^{j},$$

where $a_{i,l}^{\sigma_k}$ and $a_{i,l}^{\sigma_{k+1}}$ denote the entries of the matrices associated with σ_k and σ_{k+1}, respectively, as defined in (10).

In summary, we have the following result.

Theorem 1. *Consider the transport problem given by* (12) *and consider a state* (σ_k, k) *where* σ_k *is a permutation and a feasible solution of the transport problem. Let* f *be a function that evaluates a state with respect to a cost matrix given by* (14)*. If* $\delta((\sigma, k), f(\sigma_k, k)) = (\sigma_{k+1}, k+1)$*, then*

$$\sum_{i,k}^{2^{-j}} 2^{-j} a_{i,k}^{\sigma_{k+1}} c_{i,k}^{j} \leq \sum_{i,k}^{2^{-j}} 2^{-j} a_{i,k}^{\sigma_k} c_{i,k}^{j}.$$

equivalently

$$\sum_{l}^{2^{-j}} 2^{-j} c_{l,\sigma_{k+1}(l)}^{j} \leq \sum_{l}^{2^{-j}} 2^{-j} c_{l,\sigma_k(l)}^{j}.$$

Given an initial state of the automata $(\sigma_0, 1)$, we applied the automata to improve the feasible solution σ_0 as follows.

$$\begin{aligned} \delta((\sigma_0, 1), f(\sigma_0, 1)) &= (\sigma_1, 2) \\ \delta((\sigma_1, 2), f(\sigma_1, 2)) &= (\sigma_2, 3) \\ &\vdots \\ \delta((\sigma_{2^j-1}, 2^j), f(\sigma_{2^j-1}, 2^j)) &= (\sigma_{2^j}, 2^j + 1). \end{aligned}$$

Thus, we find that the final state of the automaton is given by $(\sigma_{2^j}, 2^j + 1)$. Therefore, the final state σ_{2^j} of the automaton provides an improved solution compared to the initial state σ_0.

Example 4. *Figure 3 illustrates the application of the tuning automaton to the Monge–Kantorovich problem with cost function* $(2y - x - 1)^2(2y - x)^2$*, discretized at level 3, which produces a cost matrix of size* 8×8*.*

The first image corresponds to a randomly generated feasible solution, obtained from a random permutation. The remaining images show the successive adjustments performed by the automaton as it evaluates possible swaps between neighboring elements.

In each step, the colored markers indicate the candidate solution under evaluation. The green dots represent the elements involved in the swap currently being tested, while the yellow dots correspond to assignments that remain fixed during that evaluation step.

The evaluation process proceeds sequentially by rows. Starting from row 0, the algorithm evaluates possible exchanges with neighboring elements; the process then continues with row 1, and so on until all rows have been examined.

The last image corresponds to the optimal solution obtained using the Hungarian algorithm. In this example, the solution produced by the proposed automaton coincides with the optimal solution.

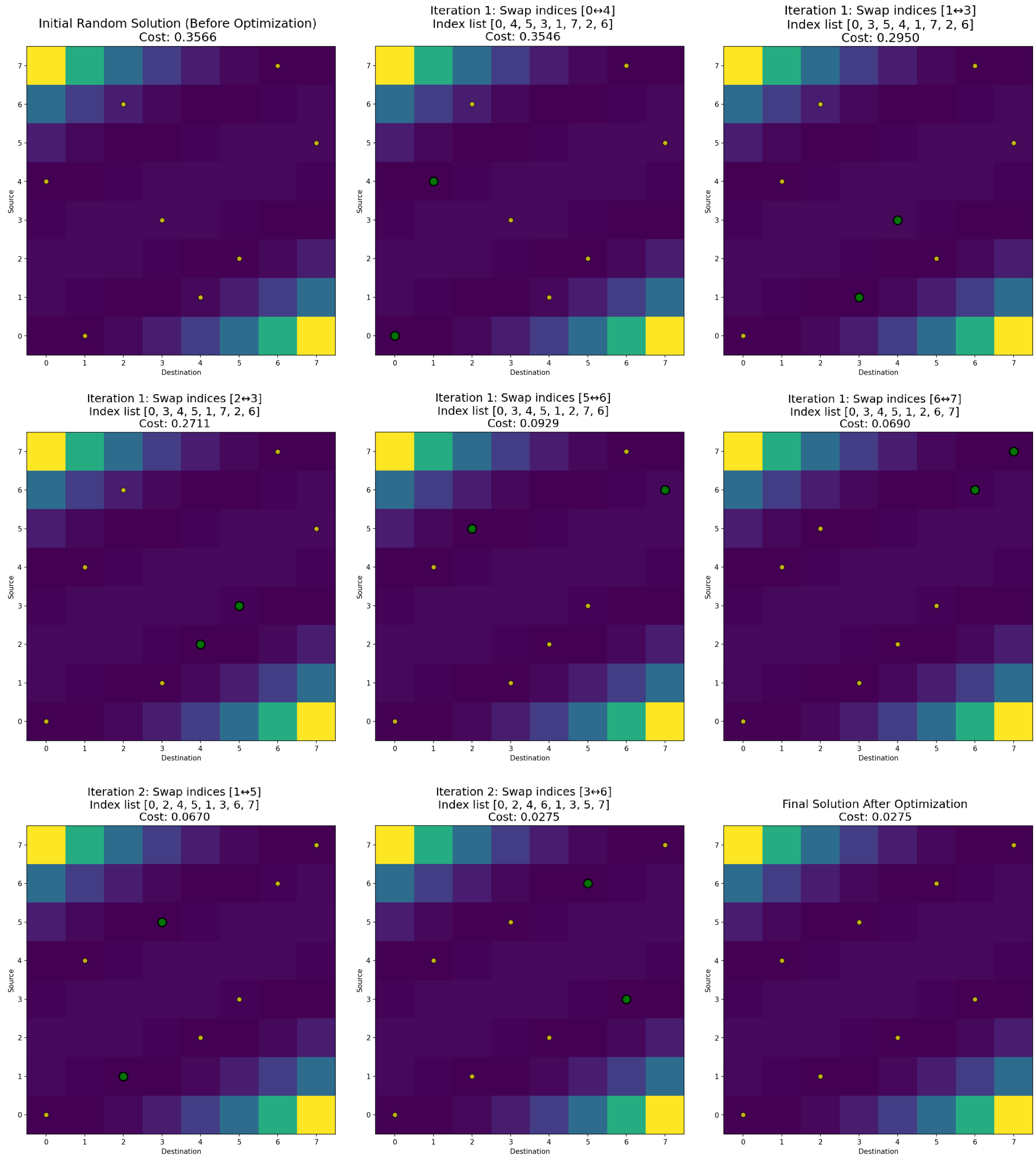


Figure 3. Evolution of the candidate solution during the optimization process. The background colors represent the values of the cost matrix. The green dots indicate the assignments involved in the swap currently being evaluated, while the yellow dots correspond to assignments that remain fixed during that step. The sequence of images shows how the automaton progressively improves the candidate solution starting from an initial random feasible assignment until reaching the optimal solution computed with the Hungarian algorithm.

5. Examples of the Proposed Methodology Based on Wavelets and an Automaton

The automaton constitutes a viable alternative for improving feasible solutions. Nevertheless, it is desirable for its initial state to be a feasible solution to the Monge–Kantorovich problem that already provides a close approximation to the optimal solution. To construct such an initial solution, we adopt the first stage of the scheme proposed in [13], which is then used as the input for the automaton.

Our approximation scheme consists of the following steps:

1. **Discretization of the cost function:** The cost function is discretized at level j, and the resulting discretized cost is denoted by c_j.
2. **Application of the wavelet transform:** Applying the wavelet transform to c_j, we obtain:
   - a low-pass component, denoted by c_{j-1};
   - three high-pass components, denoted by Ψ_1, Ψ_2, and Ψ_3.
3. **Solution of the MK^{j-1} problem:** Using the cost function c_{j-1} and the methodology proposed in [12], we compute an optimal solution μ^*_{j-1} to the MK^{j-1} problem.
4. **Feasible solution construction for MK^j problem**: Using μ^*_{j-1} together with the high-pass components Ψ_1, Ψ_2, and Ψ_3, we construct a feasible solution $\hat{\mu}_j$ at level j with the following properties:
   (a) $\mathbf{P}_{j-1}(\hat{\mu}_j) = \mu^*_{j-1}$
   (b) $\mathbf{Q}^{(1,2)}_{j-1}(\hat{\mu}_j) = 0$
   (c) $\mathbf{Q}^{(2,1)}_{j-1}(\hat{\mu}_j) = 0$
   (d) $\mathbf{Q}^{(2,2)}_{j-1}(\hat{\mu}_j) = \mathrm{sign}(\Psi_3)\mu^*_{j-1}$.

   Here, $\mathrm{sign}(\cdot)$ denotes the sign function. In other words, the support of $\hat{\mu}_j$ is contained in the support of μ^*_{j-1}. The components Ψ_1 and Ψ_2 do not participate in the construction, as they affect the boundary conditions of the MK^j problem. Finally, the sign of Ψ_3 determines the direction in which the solution is scaled.
5. **Implementation of the automaton**: We use the feasible solution $\hat{\mu}_j$ as the input to the automaton defined in (11). The automaton's output yields the final approximation.

5.1. Example 1

In this example, we consider the cost function $4x^2y - xy^2$. In Figure 4, the left side represents the discretization for level 5, where the cost function turns out on a cost matrix of 32×32, and the right side represents 3-dimensional graphics of the cost function.

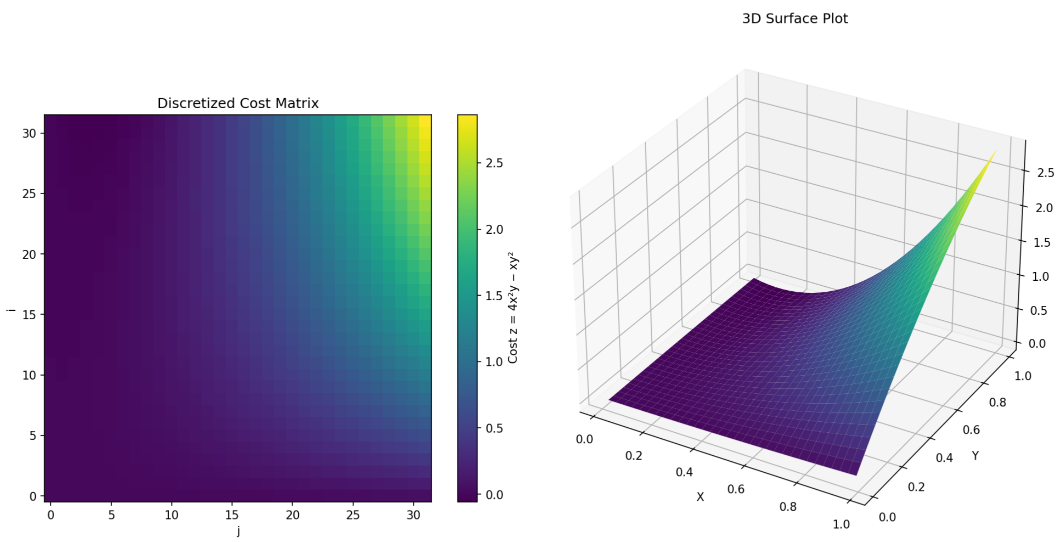


Figure 4. Cost function $4x^2y - xy^2$ and its discretization.

According to [12], the optimal solution μ_5 at this level is illustrated in Figure 5, where the red points indicate the optimal assignments of the transportation problem.

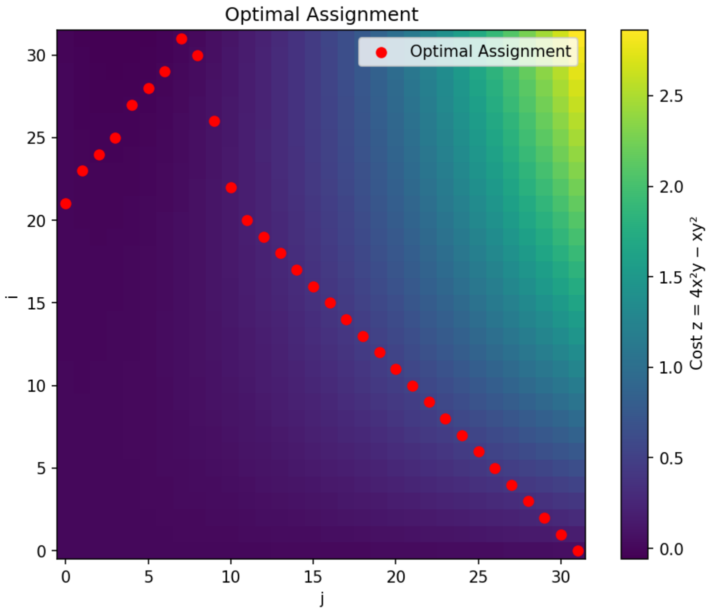


Figure 5. Optima solution using Hungarian algorithm.

Below, we present the implementation of the algorithm described at the beginning of this section.

Applying the wavelet transform to the previous function, we obtain the decomposition in Figure 6.

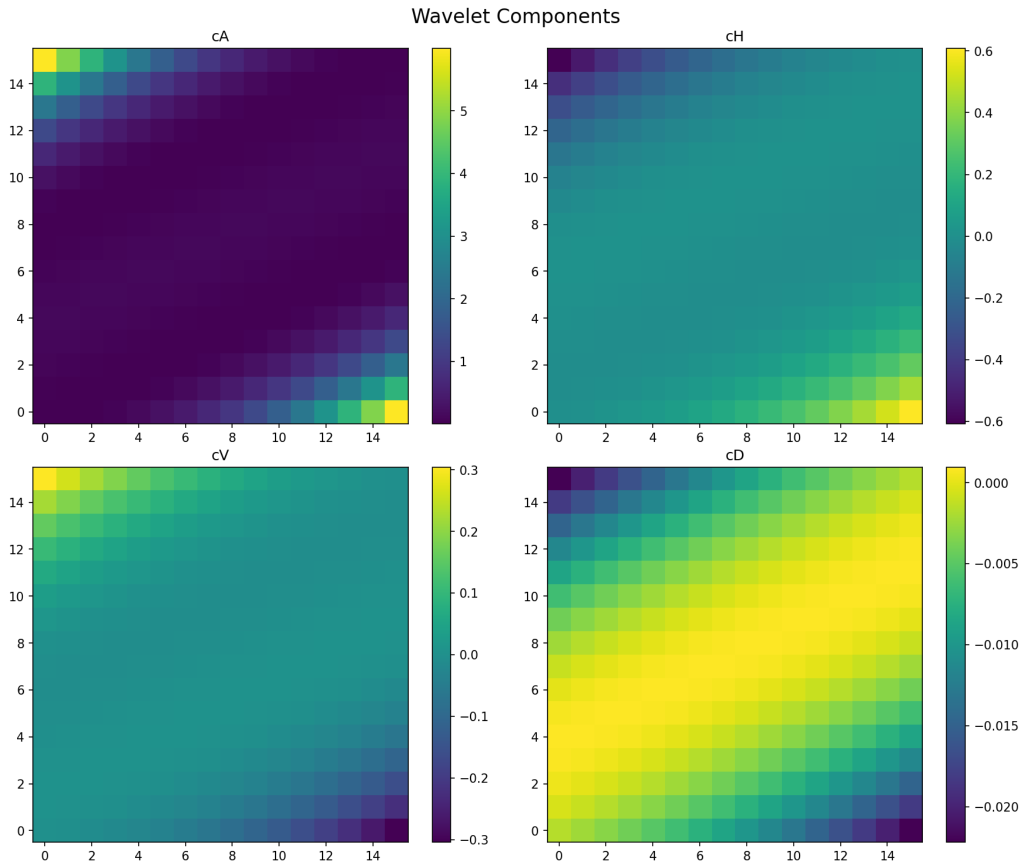


Figure 6. Wavelet Components. Wavelet decomposition components: approximation and details.

Here, the upper-left figure illustrates the lower-level approximation, the upper-right figure depicts the horizontal details, the lower-left figure shows the vertical details, and the lower-right figure corresponds to the diagonal details.

By solving the transportation problem for the cost function given by the approximation at the lower level, we obtain the measure μ_4^*. The goal is to construct a feasible solution $\hat{\mu}_5$ (level 5) from μ_4^* and the components Ψ_1, Ψ_2, and Ψ_3, as illustrated in Figure 7.

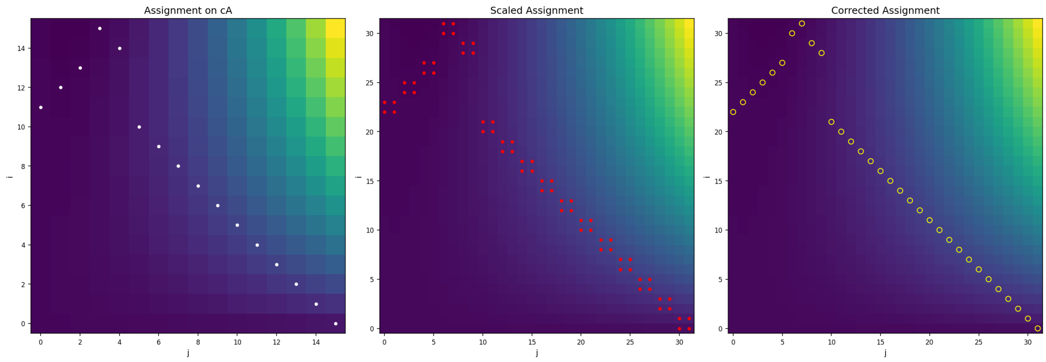


Figure 7. Assignment Comparison. Comparison between the original and approximated assignments.

Description of Figure 7.

- The figure on the left depicts the optimal solution to the transportation problem (μ_4^*), where the cost matrix is obtained from the wavelet-based approximation shown in the upper-left panel of Figure 6.
- The central figure represents the natural extension of this solution to a higher level; however, it does not constitute a feasible solution to the transportation problem.
- The figure on the right shows a feasible solution to the transportation problem at the higher level. This solution is obtained by selecting one diagonal from each 2×2 block in the central figure, with the choice guided by the sign of the diagonal detail matrix (see the lower-right panel of Figure 6).

Below, we implement the automaton defined in (11), as illustrated in Figure 8:

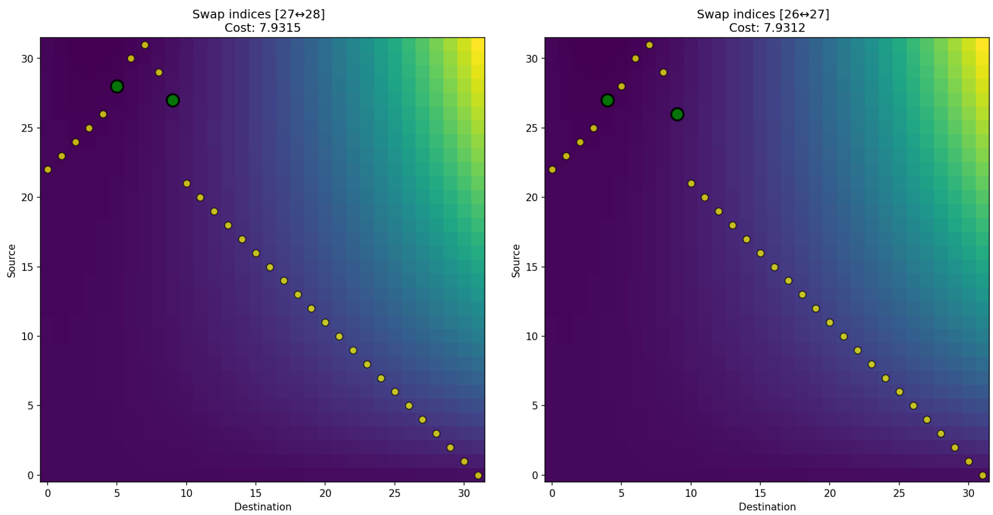


Figure 8. *Cont*.

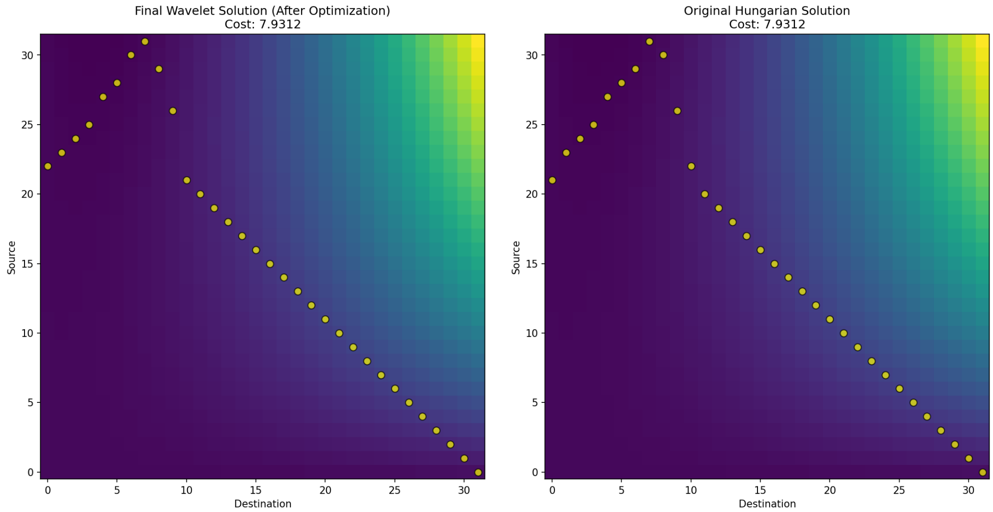


Figure 8. Implementation of the automaton whose initial state is defined by the right-hand diagram in Figure 7.

Description of Figure 8:

- The upper figures illustrate the movements performed by the tuning automaton when the initial state is given by the right-hand panel of Figure 7. The green dots indicate the positions at which the automaton performed local optimizations. For clarity, steps in which no changes occurred are omitted.
- The lower-left figure represents the final solution obtained by our procedure, which consists of applying the wavelet transform, solving the transportation problem with a 16×16 cost matrix (level 4), extending the solution, and subsequently applying the automaton (level 5).
- The lower-right figure shows the solution obtained by directly solving the transportation problem with a 32×32 cost matrix (level 5), as described in [12]. The solution obtained by our algorithm and the one obtained by directly applying the Hungarian algorithm are not identical; however, both produce the same minimum value, and therefore they can be considered equivalent solutions.
- In the numerical experiments presented in this work, the proposed automaton consistently converged to an optimal assignment. However, a complete theoretical characterization of the convergence properties of the method, including the precise conditions under which the optimal solution is unique, remains an open question. Determining such conditions and establishing formal convergence guarantees constitutes an interesting direction for future research.

5.2. Example 2

In this example, we consider the cost function $x^2y - xy^2$. The discretized function at level $j = 5$ is shown in the following Figure 9:

The feasible μ_5 solution at this level, according to [12], is shown in the following Figure 10:

On the other hand, by applying the wavelet transform to the discretization at level $j = 5$, we obtain the decomposition in Figure 11, analogous to that shown in Figure 6.

By solving the transportation problem for the cost function given by the approximation at the lower level, we obtain the measure μ_4^*. From this solution, we obtain the feasible solution $\hat{\mu}_5^*$ (Figure 12) in a form analogous to that presented in Figure 7.

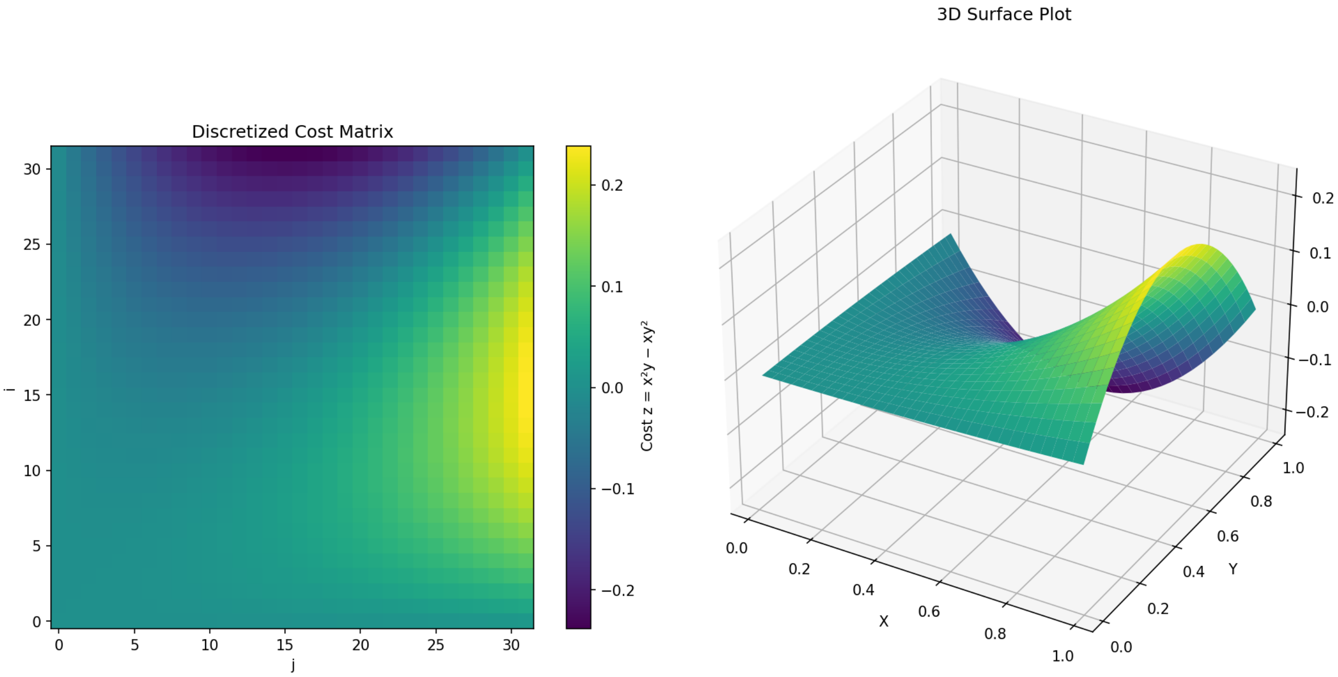


Figure 9. Cost function $x^2y - xy^2$ and its discretization.

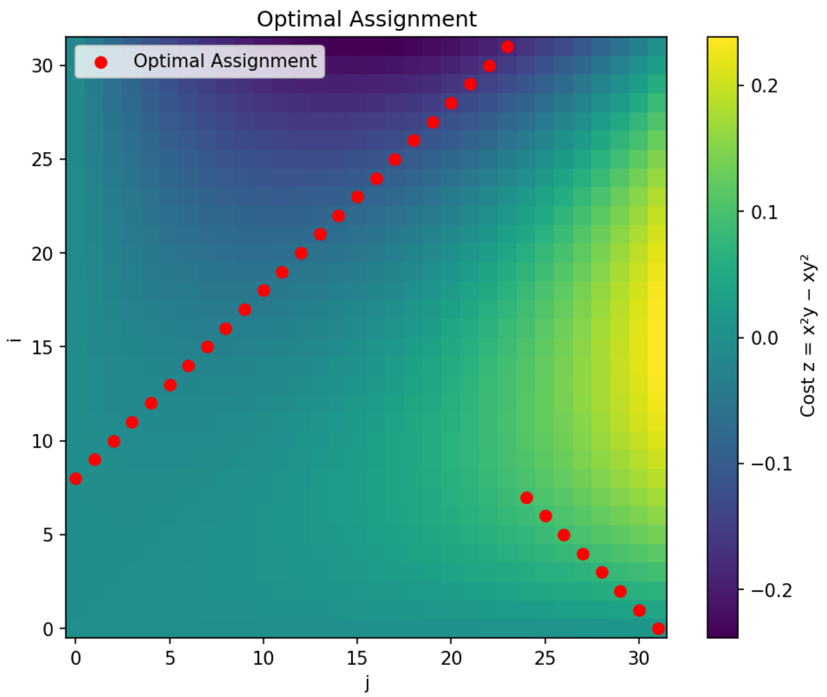


Figure 10. Optimal Assignment. Optimal assignment over the original cost matrix using the Hungarian algorithm.

In this case, the measure $\hat{\mu}_5$ constructed from μ_4^* coincides with the feasible measure μ_5. Consequently, the automaton performs no modifications, as shown in Figure 13, which are analogous to those presented in the comparison carried out in the previous example.

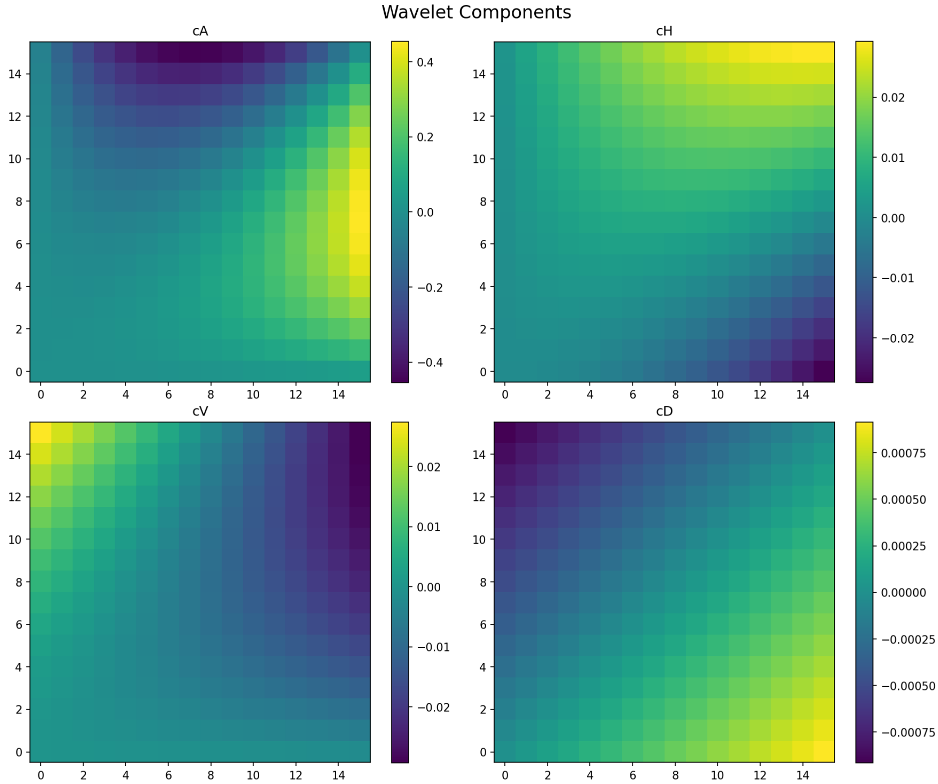


Figure 11. Wavelet Components. Composite figure of wavelet decomposition components: approximation and details.

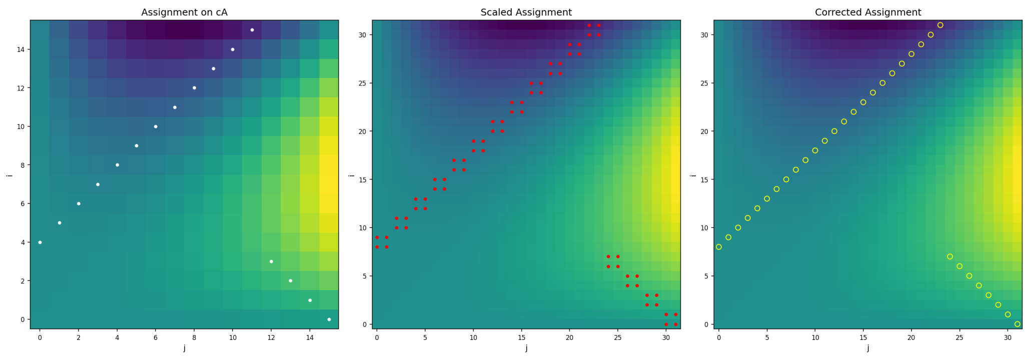


Figure 12. Assignment Comparison. Comparison between the original and approximated assignments.

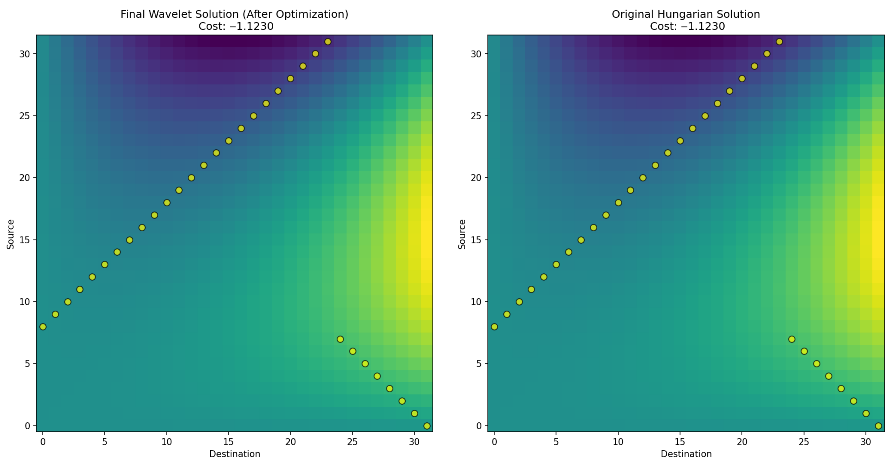


Figure 13. In this example, our methodology coincides with the Hungarian algorithm.

5.3. Example 3

In this example, we consider the cost function $(2y - x - 1)^2(2y - x)^2$. The discretized function at level $j = 5$ is shown in the following Figure 14:

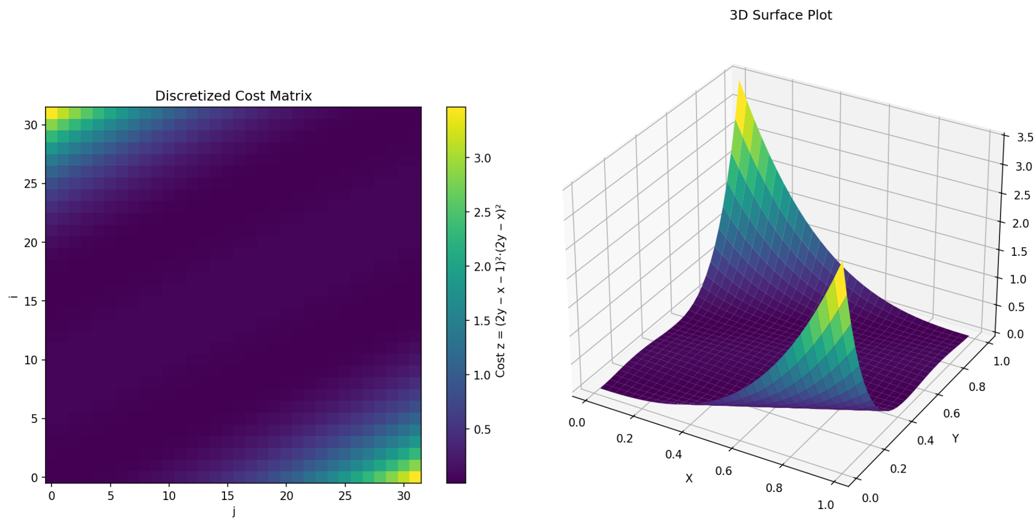


Figure 14. Cost function $(2y - x - 1)^2(2y - x)^2$ and its discretization.

According to [12], the feasible measure μ_5 at this level is illustrated in Figure 15.

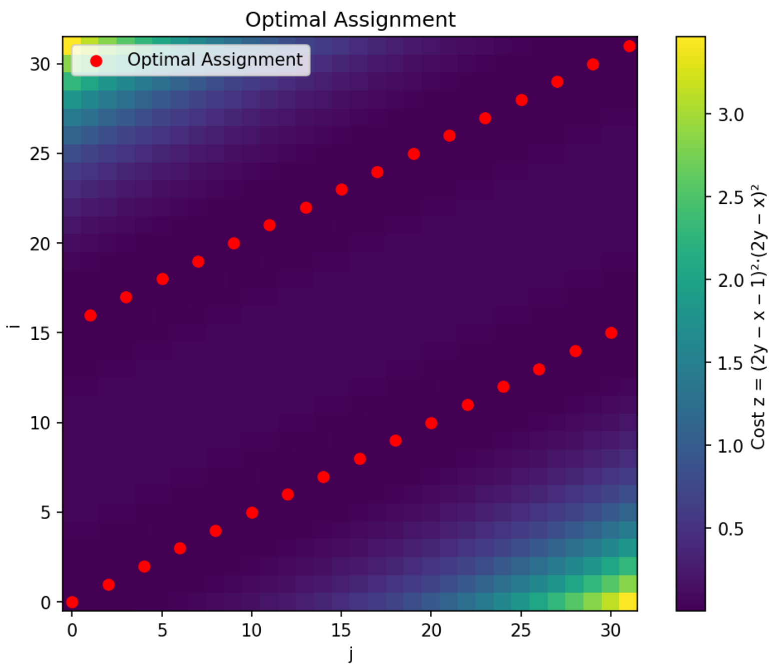


Figure 15. Optimal Assignment. Optimal assignment over the original cost matrix using the Hungarian algorithm.

By applying the wavelet transform to the discretization at level $j = 5$, we obtain the decomposition in Figure 16, which is analogous to that shown in Figure 6.

The solution of the transportation problem corresponding to the cost function given by the lower-level approximation produces the measure μ_4. From this result, we obtain a feasible solution $\hat{\mu}_5$ (Figure 17), presented in a form analogous to that depicted in Figure 7.

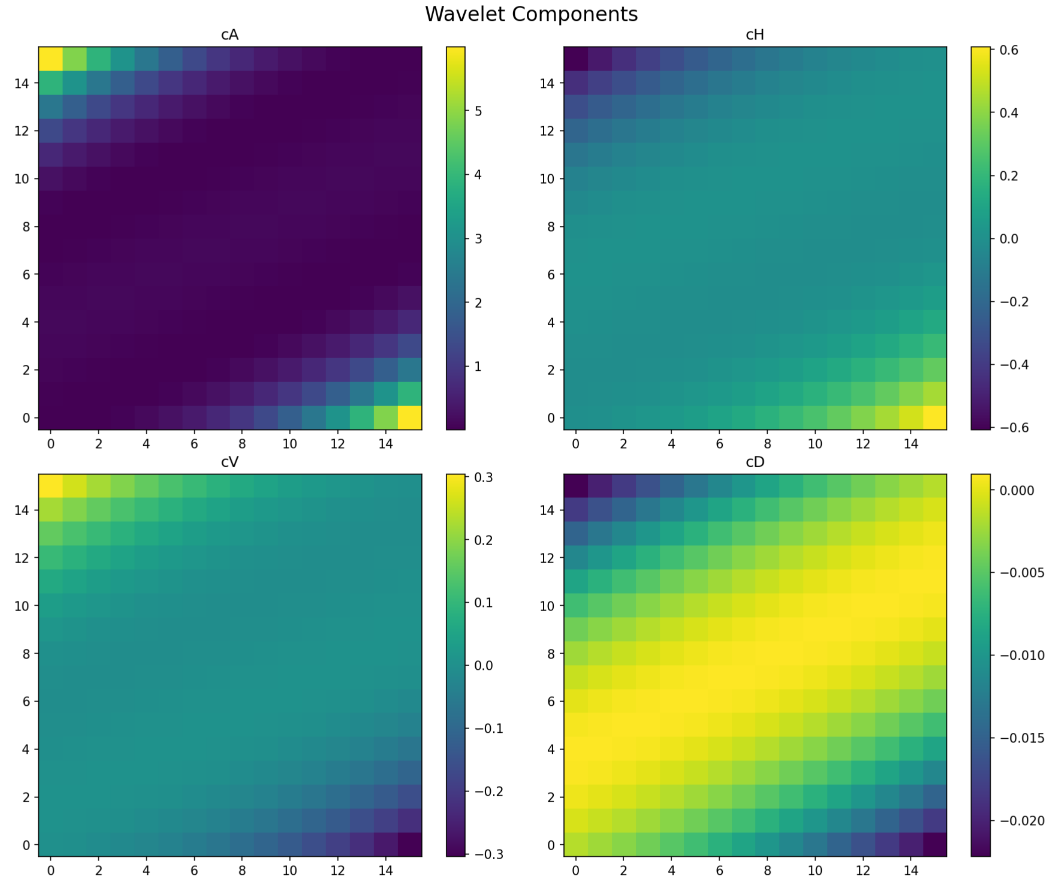


Figure 16. Wavelet Components. Wavelet decomposition components: approximation and details.

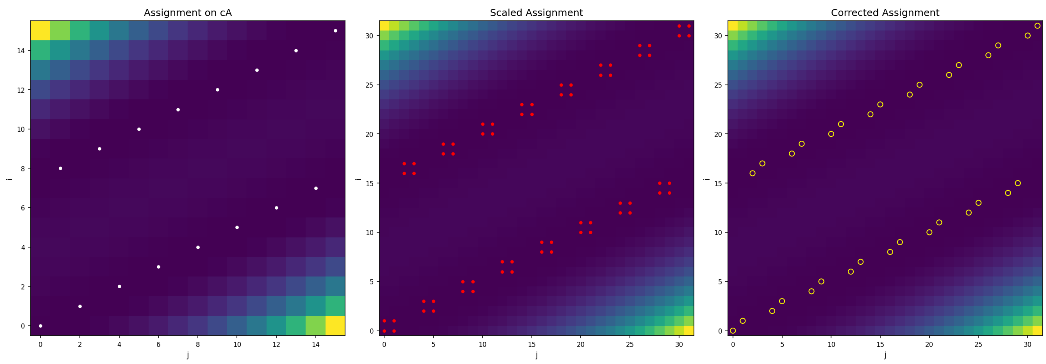
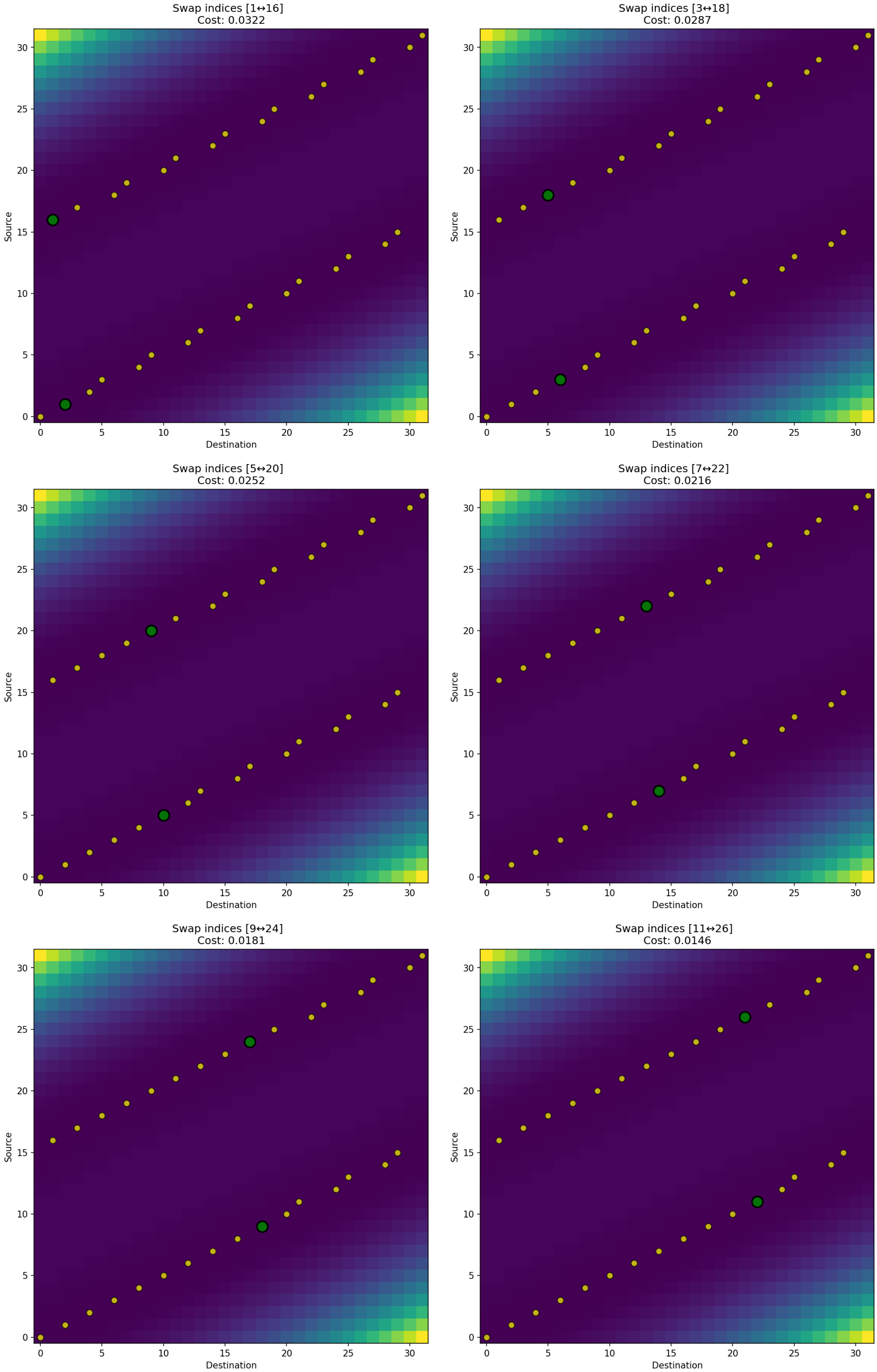


Figure 17. Assignment Comparison. Comparison between the original and approximated assignments.

Figure 18 illustrates both the changes produced by the implementation of the automaton on the feasible measure $\hat{\mu} * 5$, resulting in the feasible measure $\mu * 5$, and the sequence of candidate solutions generated by the tuning automaton after applying the scaling procedure. The process begins with the approximation obtained after scaling and performing an initial adjustment, and each subsequent image corresponds to a new approximation produced while evaluating possible swaps. The penultimate image represents the final solution obtained by the automaton, whereas the last image shows the optimal solution computed using the Hungarian algorithm for comparison.


Swap indices [1↔16]
Cost: 0.0322
Swap indices [3↔18]
Cost: 0.0287
Swap indices [5↔20]
Cost: 0.0252
Swap indices [7↔22]
Cost: 0.0216
Swap indices [9↔24]
Cost: 0.0181
Swap indices [11↔26]
Cost: 0.0146
Source
Destination

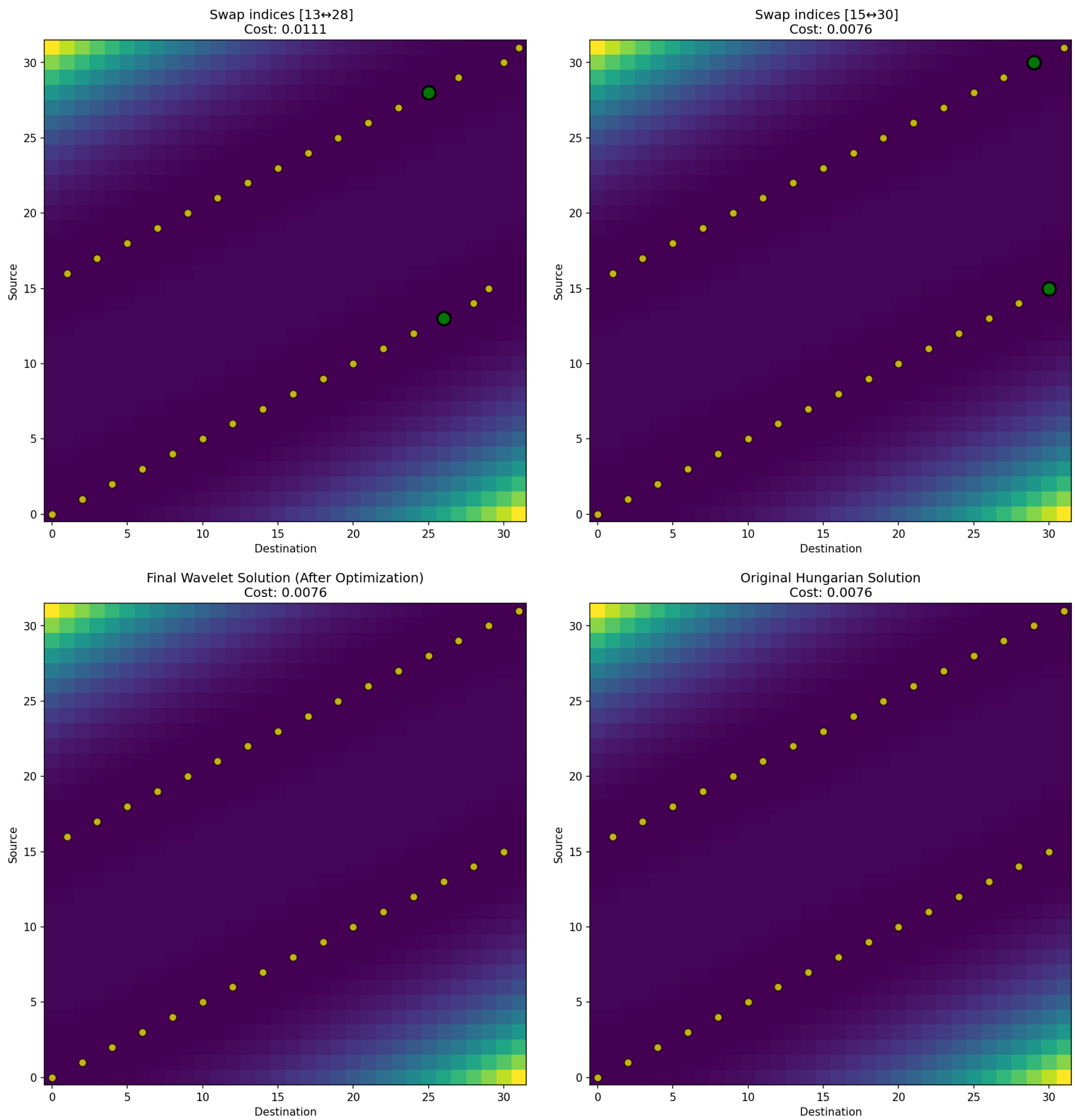


Figure 18. Evolution of the candidate solution after applying the scaling procedure. The first image corresponds to the approximation obtained after scaling and performing an initial adjustment. Each subsequent image represents a new approximation generated by the automaton as it evaluates possible swaps between neighboring assignments. The penultimate image corresponds to the final solution obtained by the automaton, while the last image shows the optimal solution computed with the Hungarian algorithm for comparison. The background colors represent the values of the cost matrix, and the colored markers indicate the assignments belonging to the candidate solution under evaluation.

6. Conclusions

An important part of this work was the introduction of an automaton on the permutation group, which proved to be a useful tool to improve feasible solutions to the transportation problem associated with the MK problem. This represents a significant contribution to automata theory applied to optimization problems.

In all examples of transportation problems (assignment problem) associated with classic Monge–Kantorovich problems with a continuous cost function, these problems have the number of agents 2^j. Therefore, solving this type of problem with the Hungarian algorithm has a computational cost of $O(2^{3j})$.

Our methodology for solving the resulting transportation problems consists of the following steps:

1. Apply the wavelet transform to a cost matrix of size $2^j \times 2^j$, which has a computational cost of $O(2^{2j})$.
2. Solve the transportation problem (assignment problem) over the approximation coefficients of the wavelet transform. At this coarser resolution level the problem involves 2^{j-1} agents, which is simpler than the original assignment problem. This reduced assignment problem is solved using the Hungarian algorithm in order to obtain an optimal assignment for the smaller cost matrix.
3. Extending the solution from level $j-1$ to level j has a computational cost of $O(2^j)$, since it is only a fixed number of steps for each element of an array of size 2^j.
4. The computational cost of the linear automaton over the size of the array 2^j, which is $O(2^j)$.

Therefore, the computational cost of this methodology for solving the transportation problem associated with the MK problem is significantly lower than that of applying the Hungarian algorithm directly. Since steps 1, 3, and 4 of our algorithm are of order $O(2^{2j})$, the main computational cost of the methodology is concentrated in step 2. In this work, this step is solved by applying the Hungarian algorithm to a reduced problem involving only half of the agents.

Although the order of the Hungarian algorithm for this reduced problem is $O(2^{3j-3}) = O(2^{3j})$, which in the worst case has the same asymptotic order, in practice fewer operations are required. Furthermore, to solve the problem with 2^{j-1} agents, our algorithm can be applied again, reducing the problem to one with 2^{j-2} agents, and so on. The study of the limits and scope of these techniques constitutes a direction for future research.

Furthermore, this methodology produced results equivalent to those obtained with the Hungarian algorithm for transport problems associated with MK problems. To illustrate this, three examples were presented at the level $j = 5$, clearly demonstrating that equivalent results are obtained.

In the transportation problems considered in this article, the cost matrices are obtained from discretizations of a continuous function, which provides a certain degree of local stability to the matrix coefficients. Future work will aim to explore the limits of this technique in settings where the cost matrix exhibits higher entropy.

In Figure 3, the process begins with a random initial solution, to which the automaton is directly applied in order to solve the problem under consideration. In this case, the automaton successfully finds the optimal solution at level 3. However, the application of the automaton at higher levels of resolution was not investigated. As a direction for future research, it would be of interest to analyze the scope and limitations of applying the automaton to more general assignment problems, particularly in cases where the cost matrix is not derived from the discretization of a continuous function.

In addition, the technique will be evaluated by applying a greater number of multiresolution levels to the cost function in order to analyze its performance and robustness.

The automaton-based refinement strategy proposed in this work could also be applied to the optimal partial transport problem. In that setting, only a portion of the total mass is transported, which introduces additional constraints in the assignment structure. The local adjustment mechanism of the automaton, which operates through sequential swaps of assignments, could be adapted to handle partial matching by allowing unassigned elements or slack variables to represent unused mass. Although such an extension would require modifications to the state representation and the evaluation rules of the automaton, the underlying idea of iterative local refinement remains applicable. Investigating this extension and analyzing its numerical behavior constitutes an interesting direction for future research.

Author Contributions: Conceptualization, A.S.-N., M.P.-M. and C.G.-F.; methodology, A.S.-N., M.P.-M., R.R.L.-M. and C.G.-F.; software, M.P.-M. and M.S.-P.; research, A.S.-N. and M.S.-P. writing—original draft preparation, A.S.-N., M.P.-M. and C.G.-F.; writing—review and editing A.S.-N. and R.R.L.-M.; project administration A.S.-N. All authors have read and agreed to the published version of the manuscript.

Funding: This publication was funded by resources from the 2026 Institutional Program for Academic Strengthening and Excellence, of the General Directorate for Academic Development and Educational Innovation of the Universidad Veracruzana.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhu, L.; Yang, Y.; Haker, S.; Tannenbaum, A. Optimal Mass Transport for Registration and Warping. *Int. J. Comput. Vis.* **2004**, *60*, 225–240. [CrossRef]
2. Haker, S.; Tannenbaum, A.; Kikinis, R. Mass Preserving Mappings and Surface Registration. In *Proceedings of the Medical Image Computing and Computer-Assisted Intervention (MICCAI 2001), Utrecht, The Netherlands, 14–17 October 2001*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2001; Volume 2208, pp. 120–127.
3. Haber, E.; Rehman, T.; Tannenbaum, A. An Efficient Numerical Method for the Solution of the L_2 Optimal Mass Transfer Problem. *SIAM J. Sci. Comput.* **2010**, *32*, 197–211. [CrossRef] [PubMed]
4. Avendaño-Garrido, M.L.; Gabriel-Argüelles, J.R.; Quintana-Torres, L.; Mezura-Montes, E. A Metaheuristic for a Numerical Approximation to the Mass Transfer Problem. *Int. J. Appl. Math. Comput. Sci.* **2016**, *26*, 757–766. [CrossRef]
5. Hernández-Lerma, O.; Lasserre, J.B. Approximation Schemes for Infinite Linear Programs. *SIAM J. Optim.* **1998**, *8*, 973–988. [CrossRef]
6. González-Hernández, J.; Gabriel-Argüelles, J.R.; Hernández-Lerma, O. On Solutions to the Mass Transfer Problem. *SIAM J. Optim.* **2006**, *17*, 485–499. [CrossRef]
7. Gabriel-Argüelles, J.R.; González-Hernández, J.; López-Martínez, R.R. Numerical Approximations to the Mass Transfer Problem on Compact Spaces. *IMA J. Numer. Anal.* **2010**, *30*, 1121–1136. [CrossRef]
8. Gröchenig, K.; Madych, W.R. Multiresolution Analysis, Haar Bases, and Self-Similar Tilings of $\mathbb{R}^n$. *IEEE Trans. Inf. Theory* **1992**, *38*, 556–568. [CrossRef]
9. Walnut, D.F. *An Introduction to Wavelet Analysis*; Birkhäuser: Boston, MA, USA, 2004; pp. 115–138.
10. Guo, K.; Labate, D.; Lim, W.Q.; Weiss, G.; Wilson, E. Wavelets with Composite Dilations and Their MRA Properties. *Appl. Comput. Harmon. Anal.* **2006**, *20*, 202–236. [CrossRef]
11. Krishtal, I.A.; Robinson, B.D.; Weiss, G.L.; Wilson, E.N. Some Simple Haar-Type Wavelets in Higher Dimensions. *J. Geom. Anal.* **2007**, *17*, 87–96. [CrossRef]
12. Sánchez-Nungaray, A.; González-Flores, C.; López-Martínez, R.R. Multiresolution Analysis Applied to the Monge–Kantorovich Problem. *Abstr. Appl. Anal.* **2018**, *2018*, 1764175. [CrossRef]
13. Acosta-Portilla, J.R.; López-Martínez, R.R.; Sánchez-Nungaray, A.; González-Flores, C. Efficient Method to Solve the Monge–Kantarovich Problem Using Wavelet Analysis. *Axioms* **2023**, *12*, 555. [CrossRef]
14. Avendaño-Garrido, M.L.; Gabriel-Argüelles, J.R.; Quintana-Torres, L.; Mezura-Montes, E. An Efficient Numerical Approximation for the Monge–Kantorovich Mass Transfer Problem. In *Machine Learning, Optimization, and Big Data*; Pardalos, P., Pavone, M., Farinella, G., Cutello, V., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2016; Volume 9432, pp. 233–239. [CrossRef]
15. Kuhn, H.W. The Hungarian Method for the Assignment Problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97. [CrossRef]
16. Kuhn, H.W. Variants of the Hungarian Method for Assignment Problems. *Nav. Res. Logist. Q.* **1956**, *3*, 253–258. [CrossRef]
17. Bazaraa, M.S.; John, J.; Hanif, D. *Linear Programming and Network Flows*; John Wiley & Sons: Hoboken, NJ, USA, 2010; pp. 513–528.
18. Billingsley, P. *Convergence of Probability Measures*; John Wiley & Sons: New York, NY, USA, 1999; pp. 27–29.
19. Rodrigues, N. A Logical Treatment of Finite Automata. In *Tools and Algorithms for the Construction and Analysis of Systems*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2024; pp. 350–369. [CrossRef]

20. Pettorossi, A. *Automata Theory and Formal Languages: Fundamental Notions, Theorems, and Techniques*; Springer: Cham, Switzerland, 2022. [CrossRef]
21. Sagan, B. *The Symmetric Group: Representations, Combinatorial Algorithms, and Symmetric Functions*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2001; Volume 203.