A. Optimality and Effectiveness

Alg. 2 computes an optimum flow $F^*$, whose components are determined by the quantities $r$ in step 4. Namely, the components of the $i$-th row of $F^*$, are given recursively as $F^*_{i,s[1]} = \min(p_i, q_{s[1]})$ and $F^*_{i,s[l]} = \min(p_i - \sum_{u=1}^{l-1} F^*_{i,s[u]}, q_{s[l]})$ for $l = 2, \ldots, h_q$.

**Lemma 1.** Each row $i$ of the flow $F^*$ of Algorithm 2 has a certain number $k_i$, $1 \leq k_i \leq h_q$ of nonzero components, which are given by $F^*_{i,s[l]} = q_{s[l]}$ for $l = 1, \ldots, k_i$ and $F^*_{i,s[k_i]} = p_i - \sum_{l=1}^{k_i-1} q_{s[l]}$.

The Lemma follows by keeping track of the values of the term $r$ in step 4 in Alg. 2. An immediate implication is that the flow $F^*$ satisfies the constraints (2) and (4). One can also show that $F^*$ is a minimal solution of (1) under the constraints (2) and (4), and this leads to the following theorem.

**Theorem 1.** (i) The flow $F^*$ of Algorithm 2 is an optimal solution of the relaxed minimization problem given by (1), (2) and (4). (ii) ICT provides a lower bound on EMD.

**Proof.** Proof of part (i): It has already been shown that the flow $F^*$ satisfies constraints (2) and (4), and it remains to show that $F^*$ achieves the minimum in (1). To this end, let $\mathbf{F}$ be any nonnegative flow, which satisfies (2) and (4). To show that $F^*$ achieves the minimum in (4), it is enough to show that for every row $i$, one has $\sum_j F_{i,j} C_{i,j} \geq \sum_j F_{i,j}^* C_{i,j}$, which then implies $\sum_{i,j} F_{i,j} C_{i,j} \geq \sum_{i,j} F_{i,j}^* C_{i,j}$.

By Alg. 2, there is a reordering given by the list $s$ such that

$$C_{i,s[1]} \leq C_{i,s[2]} \leq \ldots \leq C_{i,s[n_q]}.$$  \hspace{1cm} (10)

By Lemma 1, there is a $k_i \leq n_q$ such that $\sum_{l=1}^{k_i} F^*_{i,s[l]} = p_i$ and $F^*_{i,s[l]} = 0$ for $l > k_i$. Furthermore by Lemma 1 and by constraint (4) on $\mathbf{F}$, it follows that

$$F_{i,s[l]} \leq q_{s[l]} = F^*_{i,s[l]} \quad \text{for} \quad l = 1, \ldots, k_i - 1.$$  \hspace{1cm} (11)

The outflow-constraint (2) implies $\sum_j F_{i,j} = p_i = \sum_j F_{i,j}^*$ or, equivalently,

$$\sum_{l=k_i}^{n_q} F_{i,s[l]} = F_{i,s[k_i]} + \sum_{l=1}^{k_i-1} (F_{i,s[l]} - F_{i,s[l]}).$$  \hspace{1cm} (12)

In the following chain of inequalities, the first inequality follows from (10), and (12) implies the equality in the second step.

$$\sum_{l=k_i}^{n_q} C_{i,s[l]} F_{i,s[l]} \geq C_{i,s[k_i]} \sum_{l=k_i}^{n_q} F_{i,s[l]}$$

$$= C_{i,s[k_i]} (F_{i,s[k_i]} + \sum_{l=1}^{k_i-1} (F_{i,s[l]} - F_{i,s[l]}))$$

$$= C_{i,s[k_i]} F_{i,s[k_i]} + \sum_{l=1}^{k_i-1} C_{i,s[l]}(F_{i,s[l]} - F_{i,s[l]})$$

$$\geq C_{i,s[k_i]} F_{i,s[k_i]} + \sum_{l=1}^{k_i-1} C_{i,s[l]}(F_{i,s[l]} - F_{i,s[l]}).$$

The inequality in the last step follows from (10) and the fact that the terms $F_{i,s[l]} - F_{i,s[l]}$ are nonnegative by (11). By
rewriting the last inequality, one obtains the desired inequality
\[
\sum_j F_{i,j}^* C_{i,j} = \sum_{l=1}^{n_0} F_{i,s[l]}^* C_{i,s[l]} \\
\geq \sum_{i=1}^{k_i} F_{i,s[l]}^* C_{i,s[l]} \\
= \sum_j F_{i,j}^* C_{i,j},
\]
where in the last equation \( F_{i,s[l]}^* = 0 \) for \( l > k_i \) is used.

Proof of part (ii): Since ICT is a relaxation of the constrained minimization problem of the EMD, ICT provides a lower bound on EMD given by the output of Alg. 2, namely, \( \sum_{i,j} F_{i,j}^* C_{i,j} = \text{ICT}(p, q) \leq \text{EMD}(p, q) \).

Similar to Alg. 2, Alg. 3 also determines an optimum flow \( F^* \), which now depends on the number of iterations \( k \).

**Lemma 2.** Each row \( i \) of the flow \( F^* \) of Algorithm 3 has a certain number \( k_i \), \( 1 \leq k_i \leq k \) of nonzero components, which are given by \( F_{i,s[l]}^* = q_{a[l]} \) for \( l = 1, \ldots, k_i - 1 \) and \( F_{i,s[k_i]}^* = p_i - \sum_{l=1}^{k_i-1} q_{a[l]} \).

Based on this Lemma, one can show that the flow \( F^* \) from Algorithm 3 is an optimum solution to the minimization problem given by (1), (2) and (4), in which the constraint (4) is further relaxed in function of the predetermined parameter \( k \). Since the constrained minimization problems for ICT, ACT, OMR, RWMD form a chain of increased relaxations of EMD, one obtains the following result.

**Theorem 2.** For two normalized histograms \( p \) and \( q \): \( \text{RWMD}(p, q) \leq \text{OMR}(p, q) \leq \text{ACT}(p, q) \leq \text{ICT}(p, q) \leq \text{EMD}(p, q) \).

We call a nonnegative cost function \( C \) effective, if for any indices \( i, j \), the equality \( C_{i,j} = 0 \) implies \( i = j \). For a topological space, this condition is related to the Hausdorff property. For an effective cost function \( C \), one has \( C_{i,j} > 0 \) for all \( i \neq j \), and, in this case, \( \text{OMR}(p, q) = \sum_{i,j} C_{i,j} F_{i,j}^* = 0 \) implies \( F_{i,j}^* = 0 \) for \( i \neq j \) and, thus, \( k_i = 1 \) in Lemma 2 and, thus, \( F^* \) is diagonal with \( F_{i,i}^* = p_i \). This implies \( p_i \leq q_i \) for all \( i \) and, since both histograms are normalized, one must have \( p = q \).

**Theorem 3.** If the cost function \( C \) is effective, then \( \text{OMR}(p, q) = 0 \) implies \( p = q \), i.e., \( \text{OMR} \) is effective.

**Remark 1.** If \( \text{OMR} \) is effective, then, a fortiori, ACT and ICT are also effective. However, RWMD does not share this property.

**B. Complexity Analysis**

The algorithms presented in Section 3 assume that the cost matrix \( C \) is given, yet they still have a quadratic time complexity in the size of the histograms. Assume that the histograms size is \( h \). Then, the size of \( C \) is \( h^2 \). The complexity is determined by the row-wise reduction operations on \( C \). In case of the OMR method, the top-2 smallest values are computed in each row of \( C \) and a maximum of two updates are performed on each bin of \( p \). Therefore, the complexity is \( O(h^2) \). In case of the ACT method, the top-\( k \) smallest values are computed in each row, and up to \( k \) updates are performed on each histogram bin. Therefore, the complexity is \( O(h^2 \log k + kh) \). The ICT method is the most expensive one because 1) it fully sorts the rows of \( C \), and 2) it requires \( O(h) \) iterations in the worst case. Its complexity is given by \( O(h^2 \log h) \).

In Section 5, the complexity of Phase 1 of the LC-ACT algorithm is \( O(vhm + nh \log k) \) because the complexity of the matrix multiplication that computes \( D \) is \( O(vhm) \), and the complexity of computing top-\( k \) smallest distances in each row of \( D \) is \( O(nh \log k) \). The complexity of performing (6), (7), (8), and (9) are \( O(nh) \) each. When \( k-1 \) iterations of Phase 2 is applied, the overall time complexity of the LC-ACT algorithm is \( O(vhm + knh) \). Note that when the number of iterations \( k \) performed by LC-ACT is a constant, LC-ACT and LC-RWMD have the same time complexity. When the number of iterations are in the order of the dimensionality of the coordinates (i.e., \( O(k) = O(m) \)) and the database is sufficiently large (i.e., \( O(n) = O(v) \)), LC-ACT and LC-RWMD again have the same time complexity, which increases linearly in the size of the histograms \( h \). In addition, the sizes of the matrices \( X, V, D, \) and \( Z \) are \( nh, vm, vh, \) and \( vk \), respectively. Therefore, the overall space complexity of the LC-ACT algorithm is \( O(nh + vm + vh + vk) \).