A Hierarchical Approach to Optimal Transport

Bernhard Schmitzer and Christoph Schnörr

Image and Pattern Analysis Group, Heidelberg University

Abstract. A significant class of variational models in connection with matching general data structures and comparison of metric measure spaces, lead to computationally intensive dense linear assignment and mass transportation problems. To accelerate the computation we present an extension of the auction algorithm that exploits the regularity of the otherwise arbitrary cost function. The algorithm only takes into account a sparse subset of possible assignment pairs while still guaranteeing global optimality of the solution. These subsets are determined by a multiscale approach together with a hierarchical consistency check in order to solve problems at successively finer scales. While the theoretical worst-case complexity is limited, the average-case complexity observed for a variety of realistic experimental scenarios yields a significant gain in computation time that increases with the problem size.

1 Overview and Contribution

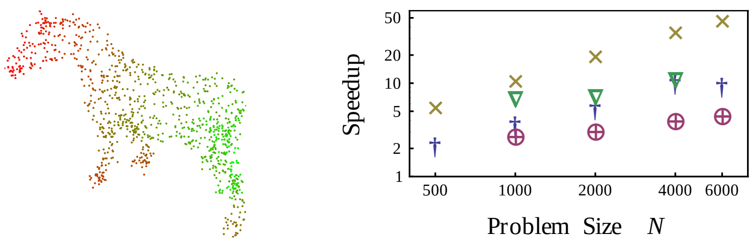
Overview The linear assignment problem (LAP) and, more general, optimal transport (OT) can be considered fundamental tools in computer vision and mathematical image processing and their properties have been thoroughly examined [10, 12]. For optimal transport between smooth distributions on $\mathbb{R}^n$ with convex cost functions, in particular the squared Euclidean distance, specialized solution methods are available [5, 6]. However, this is a rather restricted class of scenarios and the proposed ODE/PDE solutions are very involved numerically. For the LAP there are two classical algorithms: the Hungarian method [7] and the auction algorithm [1], which is apt for parallelization [2] and can be generalized to OT [4]. The evolution of the auction algorithms has also sparked investigation of more general min-cost flow problems [3].

Despite all its merits as a metric on measures [8], optimal transport has the disadvantage of being computationally considerably more expensive than simple comparisons like the L_1 distance. Thus, equivalent, yet more easily computable metrics [11], thresholded cost functions [9] or tangent space approximations [13] have been proposed.

The mentioned classical algorithms do not take into account any particular structure of the cost function, whereas for virtually all practical problems, the cost functions are far from arbitrary, but usually obey some regularity criterion. Secondly, said algorithms become very slow for large, dense problems. However many natural problems are a priori dense, i.e. any conceivable mass assignment is theoretically possible (e.g. linear shape matching relaxations discussed in [8]).

The regularity of the cost function can sometimes be exploited to devise heuristics that aim at ruling out very unlikely (mass) assignments, to reduce the problem size beforehand. Yet, in general it is very hard to come up with a *simple* in-/exclusion rule, that can both rule out a substantial fraction of possible assignments, so as to significantly reduce the problem size, and, at the same time guarantee, that the global optimum of the full problem will not be lost.

Contribution In this paper we present a modification of the auction algorithm that **(a)** can exploit any available heuristic for estimating a relevant sparse subset of assignments. However, it will at the same time be **(b)** guaranteed to find a globally optimal solution of the underlying dense problem by hierarchically checking for violated constraints of the dual problem, which relies on regularity of the cost function. In fact the hierarchical structure will lend itself to **(c)** provide a reasonable sparsity estimate for the problem at hand by a multiscale approach. Although some additional steps are required as compared to the standard auction algorithms, we show that **(d)** the worst case complexity overhead of our proposed method is limited. At the same time **(e)** we demonstrate with realistic examples, that the 'typical' problem complexity for practical setups is significantly reduced. In fact, the gain in computation time grows with problem size. This will enable application of the auction algorithm to problem sizes that were unfeasible so far and which due to their more general structure cannot be solved by PDE methods.



(a) (b)

Fig. 1: (a) Illustration of experimental scenario "mesh": mass distributions on point clouds sampled from manifolds, cost function given by point distance in underlying geodesic metric. (b) Ratio of runtimes of standard auction algorithm and our proposed extension for various scenarios (see Sect. 6) and problem sizes N. †: P2H, $\oplus$: P3H, $\times$: grid, ∇: mesh. P2H-P1, P2I and P2H-LB perform essentially like P2H. N gives the number of points per point cloud or vertices per grid. For $N = 6000$ (i.e. $N^2 = 3.6 \cdot 10^7$ potential assignment pairs) the observed speedup ranges between 4.6 and 48, consistently increasing with problem size.

In Section 2 we will recall the definitions of LAP and OT. Section 3 reviews the auction algorithm for the LAP and discusses the extension to OT. In Section 4 we present our proposed method. A comparative worst case complexity analysis is given in Sect. 5, before demonstrating with realistic experiments in Sect. 6 the significant benefit of the proposed extensions. The paper concludes in Sect. 7.

2 Linear Assignment Problem and Optimal Transport

The Linear Assignment Problem For two finite sets X, Y and a cost function $c : X \times Y \to \mathbb{R}_+ \cup \{\infty\}$ let $\mathcal{N} = \{(x, y) \in X \times Y : c(x, y) < \infty\}$. We call $\mathcal{N}$ the set of neighbours and write $\mathcal{N}(x) = \{y \in Y : (x, y) \in \mathcal{N}\}$ and similarly $\mathcal{N}(y)$. We will refer to a subset $S \subset X \times Y$ as *assignment* if it satisfies

(a) $S \subseteq \mathcal{N}$,
(b) $|\{(x', y') \in S : y' = y\}| \leq 1 \,\forall\, y \in Y$,
(c) $|\{(x', y') \in S : x' = x\}| \leq 1 \,\forall\, x \in X$.

An assignment is called *complete* if for any $x \in X$ there is a $y \in Y$ such that $(x, y) \in S$ and vice versa.

The LAP is then readily stated as

$$\min \left\{ \sum_{(x,y)\in S} c(x, y) : S \text{ is a complete assignment between } X \text{ and } Y \right\} . \tag{1a}$$

The corresponding dual problem is

$$\max \left\{ \sum_x \alpha(x) + \sum_y \beta(y) : \alpha(x) + \beta(y) \leq c(x, y) \right\} . \tag{1b}$$

Note that for any fixed β the corresponding best choice of α is given by

$$\alpha(x) = \min_y c(x, y) - \beta(y) . \tag{2}$$

It is a well known result that for any optimal assignment S of the primal problem (1a) and optimal (α, β) of the dual problem (1b) one finds

$$(x, y) \in S \Rightarrow \alpha(x) + \beta(y) = c(x, y) . \tag{3}$$

Optimal Transport For two finite sets X, Y let $\mu_X \in \mathbb{R}^{|X|}, \mu_Y \in \mathbb{R}^{|Y|}$ be two vectors with non-negative entries and equal sum of entries $\sum_x \mu_X(x) = \sum_y \mu_Y(y)$, indicating mass distributions on X, Y. Here, $c : X \times Y \to \mathbb{R} \cup \{\infty\}$ is a cost function, giving the cost to transport one unit of mass between elements of the sets.

The optimal transport problem can then be written as

$$\inf \left\{ \sum_{x,y} c(x, y)\, \mu(x, y) : \mu \geq 0, \sum_y \mu(x, y) = \mu_X(x), \sum_x \mu(x, y) = \mu_Y(y) \right\} \tag{4a}$$

where a μ is dubbed a *coupling*. The respective dual is given by

$$\sup\left\{\sum_x \alpha(x)\,\mu_X(x) + \sum_y \beta(y)\,\mu_Y(y) :\ \alpha(x)+\beta(y) \leq c(x,y)\right\}. \tag{4b}$$

Analogous to the primal-dual relation of the LAP (3) one finds for optimal transport: for any optimal μ of primal (4a) and (α,β) of dual (4b) have

$$\mu(x,y) > 0 \Rightarrow \alpha(x)+\beta(y) = c(x,y)\,. \tag{5}$$

3 The Auction Algorithm

The Auction Algorithm for the Assignment Problem We now recall the description of the auction algorithm for the LAP from [4, Sect. 2]. Note that we flipped the signs relative to the original presentation. Thus in the following the comparison to an auction is no longer very intuitive (the lowest bid gets accepted). However this makes the algorithm compatible with the usual notion of optimal transport as presented in Sect. 2.

The main loop of the algorithm is divided into two phases: *bidding* and *assignment*. During the bidding phase elements of X locally determine their most suitable assignment partner in Y and propose a corresponding dual variable change. After that, during the assignment phase, for each $y \in Y$ the best proposed dual variable change is implemented. Different x do not interact during the bidding phase and neither do different y during the assignment phase. Thus both stages can be easily parallelized.

The state of the algorithm is represented by an assignment S and dual variable β. The corresponding α is held implicitly via (2). The algorithm is initialized with the empty assignment $S = \emptyset$ and some arbitrary β. A key property of the auction algorithm is, that condition (3) does not hold strictly throughout the iterations. Instead at any stage during the algorithm, for any $(x,y) \in S$ the weaker condition $\alpha(x)+\beta(y) \geq c(x,y)+\varepsilon$ is satisfied, where ε is some positive parameter. Positivity of ϵ is essential for convergence of the algorithm. However, as long as $\varepsilon < \Delta c/|X|$ the resulting complete S is guaranteed to solve (1a), where Δc is the smallest difference between two non-equal values of c.

Bidding Phase For every $x \in X$ that is unassigned under S:

Compute the corresponding value of $\alpha(x)$ as given by (2):

$$\alpha(x) = \min_{y \in \mathcal{N}(x)} c(x,y) - \beta(y) \tag{6}$$

and find a minimizer y^*. Determine also the slack of the second 'nearest' constraint:

$$\alpha'(x) = \min_{y \in \mathcal{N}(x)\setminus\{y^*\}} c(x,y) - \beta(y) \tag{7}$$

Then element $x \in X$ bids for element $y^* \in Y$ with value

$$b_{xy^*} = c(x,y^*) - \alpha'(x) - \varepsilon\,. \tag{8}$$

Assignment Phase For each $y \in Y$ let $P(y)$ be the set of $x \in X$ from which y received a bid in the bidding phase of the iteration. If $P(y)$ is nonempty, decrease $\beta(y)$ to the lowest bid

$$\beta(y) := \min_{x \in P(y)} b_{xy} \tag{9}$$

remove from the assignment S any pair (x, y) (if one exists), and add to S the pair (x^*, y) where x^* is some element in $P(y)$ attaining the minimum in (9). If $P(y)$ is empty, $\beta(y)$ is left unchanged.

Repeat the two stages until S is complete.

The Auction Algorithm for Optimal Transport In principle any optimal transport problem with integer mass distributions can be translated into an LAP by introducing a 'mass-atom' and splitting up each node $x \in X$, $y \in Y$ into multiple copies, depending on how many atoms fit into $\mu_X(x), \mu_Y(y)$. By applying suitable data structures this splitting can be made implicit and the auction algorithm does not actually need to handle each mass atom separately. For example, *assignments* S will be replaced by *couplings* μ. Also, some modifications in the bidding process are advisable to prevent inefficient competition between atoms originating from the same elements of X.

Such a reformulation is given in [4, Sect. 4], which we cannot repeat here, due to space limitations. Instead we will briefly comment on the modifications which are relevant for our proposed extensions to be discussed in the next section.

In the generalized algorithm, due to the splitting, the dual variable β need not be constant 'within' every y. Thus, there is a dual variable $\tilde{\beta}$ for every pair (x, y) and one variable $\tilde{\beta}(\Diamond, y)$ for mass atoms in y which have not yet received a bid. A dual variable β can be obtained by

$$\beta(y) = \begin{cases} \max_{x' \in X\,:\,\mu(x',y)>0} \beta(x', y) & \text{if } \sum_{x'} \mu(x', y) = \mu_Y(y) \\ \beta(\Diamond, y) & \text{else} \end{cases}.$$

In the bidding phase, any x with $\sum_y \mu(x, y) < \mu_X(x)$ can submit bids to multiple y simultaneously. To determine the bid recipients, consider the set

$$\begin{aligned} \Pi(x) = \; & \{c(x, y) - \beta(x', y) \,|\, y \in \mathcal{N}(x),\, x' \neq x \text{ and } x' \in \mathcal{N}(y),\, \mu(x', y) > 0\} \\ & \cup \{c(x, y) - \beta(\Diamond, y) \,|\, y \in \mathcal{N}(x),\, \textstyle\sum_{x'} \mu(x', y) < \mu_Y(y)\} \end{aligned} \tag{10}$$

and assume that the entries are arranged in ascending order, i.e. we have

$$\Pi(x) = \Big\{c(x, y_1) - \beta(x'_1, y_1), \ldots, c(x, y_{|\Pi(x)|}) - \beta(x'_{|\Pi(x)|}, y_{|\Pi(x)|})\Big\} \tag{11}$$

with $c(x, y_i) - \beta(x'_i, y_i) \leq c(x, y_{i+1}) - \beta(x'_{i+1}, y_{i+1})$, for all $i = 1, \ldots, |\Pi(x)| - 1$, where by abuse of notation we allow $x'_i = \Diamond$ for some i.

Values (6) and (7) are the first two entries of this list in the LAP case, for determining the bids in a general OT problem, more than two entries might be relevant. Depending on the mass distributions μ_X, μ_Y, one will determine an integer $m > 1$ such that the equivalent of (7) is given by

$$\alpha'(x) = c(x, y_m) - \beta(x'_m, y_m)\,. \tag{12}$$

For a complete description of the algorithm we refer the reader to [2].

4 A Hierarchical Multiscale Approach to Optimal Transport

Motivation Obviously both algorithms will perform faster on sparse problems, where the set of neighbours $\mathcal{N}$ is small. For example, the creation of the list (10) will require much fewer queries. In practice however, many problems are dense and a priori any assignment (x, y) could be possible. For some applications one might be able to devise good heuristics to exclude certain pairs, which are unlikely part of an optimal solution. But due to the combinatorial structure of the underlying LAP it is in general hard to rule out a significant amount of potential assignments and yet guarantee that the global optimum of the full problem will be attained.

In most practical problems the sets X and Y are equipped with some additional structure and notion of *closeness or similarity* which is also represented in the cost function. If x and y are close to x' and y' respectively, then we expect $|c(x,y) - c(x', y')|$ to be somehow bounded. The details of this boundedness condition (e.g. Lipschitz continuity) may depend on the problem at hand and are not crucial for the applicability of the scheme to be discussed.

We will now present a sparse/dense hybrid variant of the auction algorithm, that can be initialized with a good heuristic guess for the subset of relevant assignment pairs and will benefit from the sparsity of this set and the additional available structure of X, Y and c. Yet it will be guaranteed to find a globally optimal assignment or coupling measure (Proposition 1). This hybrid variant can then be used in a multiscale scheme, that successively generates optimal couplings at finer and finer scales of the problem, using the results from the coarser scales for efficiently solving the finer scales. A central concept of this algorithm are hierarchical partitions, to be introduced next.

Hierarchical Partitions Let $\mathcal{A}_1 \subset 2^{|X|}$ be a partition of X, such that any two elements x, x' of one partition cell are considered to be 'close' in the aforementioned sense. Then let $\mathcal{A}_2$ be another (coarser) partition that is compatible with $\mathcal{A}_1$ in the sense that any element $a \in \mathcal{A}_2$ can be written as the union of some cells of $\mathcal{A}_1$. This coarsening can be repeated multiple times, each time ensuring that elements in the same cell satisfy some (scale-adjusted) closeness criterion. The resulting structure implies a directed tree graph with vertex set $\mathcal{A} = \bigcup_{i=0}^{g-1} \mathcal{A}_i$ where $\mathcal{A}_0 = \{\{x\} :\ x \in X\}$ is the set of singletons of X and g is the depth of the hierarchy. For $0 \leq i < g$ we say $a' \in \mathcal{A}_i$ is a child of $a \in \mathcal{A}_{i+1}$ (and a is parent of a') and write $a' \in \mathrm{ch}(a), a = \mathrm{pa}(a')$ if $a' \subset a$. We call this a *hierarchical partition* of X.

Analogous we let $\mathcal{B}$ be a hierarchical partition of Y and w.l.o.g. assume that $\mathcal{A}$ and $\mathcal{B}$ have the same depth.

Now for a given dual variable α define the extension $\hat{\alpha}$ onto the whole hierarchical partition by

$$\hat{\alpha}(a) = \max_{x \in a} \alpha(x) = \begin{cases} \alpha(x) & \text{if } a = \{x\} \in \mathcal{A}_0 \text{ for some } x \\ \max_{a' \in \mathrm{ch}(a)} \hat{\alpha}(a') & \text{if } a \in \mathcal{A}_i \text{ for some } i > 0 \end{cases} \tag{13}$$

and analogous for β and $\hat{\beta}$.

Similarly define an extension $\hat{c}$ of c onto $\mathcal{A} \times \mathcal{B}$ via

$$\hat{c}(a, b) = \min_{x \in a, y \in b} c(x, y)\,. \tag{14}$$

We now define an extension of the dual constraints of (1b,4b) to coarser scales: we will refer to the following set of inequalities as *dual constraints of generation n*:

$$\hat{\alpha}(a) + \hat{\alpha}(b) \leq \hat{c}(a, b) \,\forall\, (a, b) \in \mathcal{A}_n \times \mathcal{B}_n \tag{15}$$

Obviously if the dual constraints of generation n hold for some extended $\hat{\alpha}, \hat{\beta}$ and $\hat{c}$, then so will the constraints at all generations $n' < n$. For $n = 0$ these constraints are those of the original optimal transport problem. The requirement that elements within the same partition cell of any generation should be close, will ensure, that the dual constraints of generation n will not be a lot tighter than those of generation $n - 1$.

A Sparse/Dense Hybrid Variant of the Auction Algorithm Consider a feasible optimal transport problem between (X, μ_X) and (Y, μ_Y) with cost function c. Let $\hat{\mathcal{N}} \subset X \times Y$ such that $(x, y) \in \hat{\mathcal{N}} \Rightarrow c(x, y) < \infty$. However not necessarily $c(x, y) < \infty \Rightarrow (x, y) \in \hat{\mathcal{N}}$, i.e. we might start with a set of neighbours which is smaller than the maximally possible one. We now give an algorithm that will run on a given submaximal neighbour set $\hat{\mathcal{N}}$, but detect if some $(x, y) \in \mathcal{N}$ might have to be considered as part of an assignment and extend $\hat{\mathcal{N}}$ accordingly if necessary. The *bidding* and *assignment* phases will work just as in the standard auction algorithms, Sect. 3, with $\hat{\mathcal{N}}$ in place of $\mathcal{N}$. But there will be an additional *consistency check* step in between:

Consistency Check Phase Let $\hat{\alpha}'$ be the hierarchical extension of α' as defined in (7,12) and $\hat{\beta}$ the hierarchical extension of $\beta(\cdot)$. Then start with checking whether $\hat{c}(a, b) - \hat{\beta}(b) \geq \hat{\alpha}'(a)$ for all $a \in \mathcal{A}_n, b \in \mathcal{B}_n$ at some generation $n > 0$.

If a checked inequality holds, then certainly $c(x, y) - \beta(x', y) \geq \alpha'(x)$ for all $x \in a, y \in b, x' \in X$ and thus no $y \in b$ could lead to a different bid for $x \in a$ if $(x, y) \in \hat{\mathcal{N}}$ during the *bidding phase*, since these potential candidates would appear further behind in the ordered list $\Pi(x)$, (11).

If a checked inequality $\hat{c}(a, b) - \hat{\beta}(b) \geq \hat{\alpha}'(a)$ is found to be violated, check on a finer level: $\hat{c}(a', b') - \hat{\beta}(b') \geq \hat{\alpha}'(a')$ for $a' \in \mathrm{ch}(a), b' \in \mathrm{ch}(b)$. Recursively continue this process until either all inequalities hold, or at generation 0 a candidate $c(x,y) - \beta(y) < \alpha'(x)$ is found. If for such a candidate $(x,y) \notin \hat{\mathcal{N}}$, then update $\hat{\mathcal{N}} := \hat{\mathcal{N}} \cup \{(x,y)\}$ and list x for rebidding.

After the consistency check, reevaluate the bidding phase for all listed x.

Proposition 1. *The sparse/dense hybrid auction algorithm, initialized with some non-maximal neighbourhood set $\hat{\mathcal{N}}$, such that the problem constrained to $\hat{\mathcal{N}}$ is still feasible, will converge to a globally optimal coupling μ under the same conditions as the dense algorithm variant.*

The proof is rather simple and thus for lack of space will be postponed to a more thorough article on the subject. It hinges on the fact, that elements in the list $\Pi(x)$, Eq. (11), that appear beyond position m (which determines the value of α', see (12)), do not influence the process of the algorithm.

It should be noted, that this modification preserves the parallel structure of the algorithm. Bidding and assignment work as before and the tree structure of the successive hierarchical consistency checks allows for distribution of the consistency evaluation onto multiple processors.

A Hierarchical Multiscale Approach to Optimal Transport The hybrid variant will give a globally optimal coupling μ for valid initializations of $\hat{\mathcal{N}}$ and usually require far less queries than a naïve dense algorithm, if the initial $\hat{\mathcal{N}}$ is chosen well and c is 'sufficiently regular' within the partition cells. For specific problems one may devise good heuristics for such an initial guess. Now we want to propose a *generic scheme, that works in principle for any problem.* Its practicality will be evaluated in Sect. 6. Again, to save space, we can only so much as give a sketch and must omit proofs for now.

For an optimal transport problem the coarsened problem at generation n is defined by

$$\begin{aligned} &\inf \sum_{(a,b)\in\mathcal{A}_n\times\mathcal{B}_n} \hat{c}(a,b)\,\hat{\mu}(a,b) \text{ subject to} \\ &\hat{\mu} \geq 0,\ \sum_b \hat{\mu}(a,b) = \sum_{x\in a} \mu_X(x),\ \sum_a \hat{\mu}(a,b) = \sum_{y\in b} \mu_Y(y)\,. \end{aligned} \tag{16}$$

Denote by D_n its optimal value.

Let Δc_n be an upper bound on the variation of c within one partition cell of $\mathcal{A}_n \times \mathcal{B}_n$, i.e. $\hat{c}(a,b) \leq c(x,y) \leq \hat{c}(a,b) + \Delta c_n$ for $(a,b) \in \mathcal{A}_n \times \mathcal{B}_n$, $(x,y) \in a \times b$. In addition, any feasible $\hat{\mu}$ of the coarsened problem at some generation n does induce feasible couplings on lower generations. Let $\hat{\mu}'$ be some feasible coupling of generation $n-1$ induced by an optimizer $\hat{\mu}$ of generation n, then one can easily proof that

$$D_n \leq D_{n-1} \leq \sum_{(a,b)\in\mathcal{A}_{n-1}\times\mathcal{B}_{n-1}} \hat{c}(a,b)\,\hat{\mu}'(a,b) \leq D_n + \Delta c_n \cdot M\,,$$

where $M = \sum_x \mu_X(x)$. Thus, solving the problem of generation n not only provides a bounded interval for D_{n-1} but also gives a feasible candidate for the problem of generation $n-1$ which is at most suboptimal by a margin $\Delta c_n \cdot M$.

Since c is supposed to be regular in some sense and partitions are to be chosen according to the closeness structure on X and Y, we can assume, that Δc_n is usually small compared to the fluctuations of c throughout the whole coupling space and that, thus, this bound is of actual practical value.

Also, it seems natural, to pick the support of $\hat{\mu}'$ as initial guess for $\hat{\mathcal{N}}$, when solving the refined problem with the hybrid algorithm. Obviously the restriction to $\hat{\mathcal{N}}$ keeps the problem feasible, since it allows $\hat{\mu}'$.

Thus, in short, instead of directly solving the problem at generation 0, we start at some coarser scale n, where the problem is small enough for direct dense solution. Then we use the obtained minimizers to recursively solve the problem at finer scales, each time producing an initial guess for the sparse support subset.

5 Complexity Analysis

We will first give the *worst case complexity analysis of the auction algorithm* for the dense LAP with $\mathcal{N} = X \times Y$, $|X| = |Y|$. It can be considered a special case of a class of min-cost flow algorithms presented in [3]. From [3, Lemma 5] we can see that the number of bids submitted per source is $\mathcal{O}(|X| \cdot C)$ where

$$C = \max_{x,y} c(x,y) - \min_{x,y} c(x,y)\,.$$

From the description in Sect. 3 we can see that cost of one bid for a given source is of order $\mathcal{O}(|\mathcal{X}|)$, i.e. scanning every possible assignment partner once. This already incorporates the costs of bid acceptance at one sink, since at most one bid is accepted per submitted bid. Hence the total worst case complexity of the algorithm is $\mathcal{O}(|X|^3 \cdot C)$.

The *extension to the sparse/dense hybrid variant* requires several additional steps, of which we must estimate the worst case costs. In a worst case scenario any possible link will be added to $\hat{\mathcal{N}}$, i.e. $\hat{\mathcal{N}} = X \times Y$, as in the full problem. Let p be an upper bound on the number of elements in one partition cell at any generation of the hierarchical partitions and let g be the number of generations. Then per bid submission at most $\mathcal{O}(g)$ steps are required to compute the extension $\hat{\alpha}'$ and at most $\mathcal{O}(p \cdot g)$ per reception to update $\hat{\beta}$. There will be of the order $\mathcal{O}(|\mathcal{B}|)$ hierarchical constraints to be tested per bid. Thus for one bid we get costs of the order $\mathcal{O}(|X| + g \cdot (p+1) + |\mathcal{B}|)$, resulting in a total worst case complexity of $\mathcal{O}\big(|X|^2 \cdot C \cdot (|X| + g \cdot (p+1) + g \cdot |\mathcal{B}|)\big)$. In the worst case, after the consistency phase, the bidding phase needs to be rerun completely. However, this only amounts to a constant factor 2 in the number of steps.

If the *hierarchical partitions* satisfy a relation like $|\mathcal{B}_{n+1}| \leq |\mathcal{B}_n| \cdot q$ for some $q \in]0,1[$ then $|\mathcal{B}| \leq \sum_{k=0}^{g-1} |X| q^k < |X|/(1-q)$. For octrees one has for example $q = 1/8$. Also, usually $g, p \ll |X|$, for example $p \approx |\mathcal{A}_{n+1}|/|\mathcal{A}_n| \approx 1/q$ ($= 8$ for octrees) and $g = \mathcal{O}\big(\log(|X|/|\mathcal{A}_{g-1}|)/\log(1/q)\big)$ where $\mathcal{A}_{g-1}$ would be the coarsest generation of the hierarchical partition. Thus, the complexity of the hybrid variant is usually dominated by the last term, which yields $\mathcal{O}\big(|X|^3 \cdot C \cdot g/(1-q)\big)$. Hence, the overhead scales with a constant factor $(1-q)^{-1}$, depending on the hierarchy structure, and a term logarithmic in $|X|$ which accounts for the hierarchy resolution.

In principle the algorithm presented in [3] can also be used to solve the general optimal transport problem, resulting in a similar complexity bound. The variant referred to in Sect. 3 has a much higher worst case complexity but tends to perform faster in practice due to increased resistance to a phenomenon dubbed *price haggling* [3]. This means that the additional steps required by our hybrid variant are of little significance in the worst case, yet are very useful in the 'typical' case, as demonstrated in the next section.

In practice runtime of the auction algorithms does exhibit a strong sensitivity to C. This can be remedied by a method called *ε-scaling* [3] which can be shown to replace the factor C by $\log(|X| \cdot C)$ in the complexity estimates. Also, this method is compatible with our presented additions.

6 Experiments

In the previous section we have considered the *theoretical worst case complexity* of the auction algorithm and its hybrid extension. It is however very hard to obtain a theoretical estimate for the *'typical' complexity*. Thus, for demonstrating the benefit of the augmented algorithm we need to rely on numerical experiments.

Implementation Details For evaluation we implemented the auction algorithm in `c++` with sparse data structures. The hybrid variant is based on the same implementation, extended by the consistency phase, to obtain a meaningful performance comparison. All mass distributions were picked to be integer and the cost functions were truncated to a fine discrete grid of equidistant values. To get practically relevant solving times, we used a very rudimentary form of ε-scaling, in which the problem is repeatedly solved for decreasing values of ε until global optimality can be guaranteed.

Performance Measures Computation time is naturally the measure of performance that matters most in the end. To gain additional insight we also consider the number of queries required to construct the list $\Pi(x)$, (11), the additional number of queries in the hierarchical consistency phase and the degree of sparsity of $\hat{\mathcal{N}}$ in the hybrid method.

Experimental Scenarios We consider a variety of problem scenarios for evaluation: **(a)** P2H: point clouds, each uniformly sampled from the 2D unit square, squared Euclidean distance as cost, **(b)** P3H: same as P2H, but points sampled from 3D unit cube, **(c)** P2H-P1: same as P2H but with non-squared Euclidean distance as cost, **(d)** P2I: same as P2H but with inhomogeneous sampling densities and **(e)** grid: smooth 2D mass distribution, approximated by a discrete grid, cost given by squared Euclidean distance, **(f)** mesh: mass distributions on points sampled from the surface of a 3D mesh, geodesic distance (within mesh surface) as cost function. In all experiments quadtrees (resp. octrees in 3D) were used as hierarchical structures.

Last, we test an additional scenario, **(g)** P2H-LB: same as P2H, but instead of computing $\hat{c}$ by explicit minimization as in (14), we use lower bounds directly obtained from the quadtree structure. This demonstrates that the method can also be applied to avoid explicit computation of all pairwise costs, which for more complicated problems might be a costly task in itself.

Results A summary of the numerical results is given in Table 1. The hybrid variant is *significantly faster* than the regular algorithm *for all* presented scenarios. This is due to a drastic decrease in the number of necessary constraint violation queries. In particular one can see (Fig. 1) that *the gain increases with growing problem size.* For $N = 6000$ (i.e. for $3.6 \cdot 10^7$ possible assignment pairs) the ratio of runtimes ranges from 4.6 to 48. In the hybrid variant, for most scenarios at the finest scale less than one percent of potential assignments was added to $\hat{\mathcal{N}}$. Only for mesh it was slightly more ($\approx 4\%$), owed to the more complicated cost function. Also in the scenario P2D-LB the hybrid variant clearly outperforms the regular algorithm, while at the same time potentially saving explicit assignment cost computation. Thus, for the presented scenarios the multiscale scheme obviously works as intended.

7 Conclusion

As demonstrated in the last Section, the presented extension of the auction algorithm clearly outperforms the regular variant on all presented test scenarios. The observed gain in computation time grows with problem size. Compared to PDE approaches for OT problems our method is much more flexible: X and Y need not be regular grids on $\mathbb{R}^n$ and the cost can be chosen freely, as long as a certain regularity is retained. Due to the very limited space we could only give a very brief sketch on the theoretical properties of the algorithm, i.e. its worst case complexity, the claim that it reliably finds the global optimum and the relation between the different scales of the problem. Proofs for these claims will be presented in a more detailed future publication. It also remains to be examined more carefully how the hierarchical structure we proposed interacts with the ε-scaling scheme or whether under further assumptions on the cost function better theoretical complexity bounds can be obtained. Yet, already at this stage of research the potential of the extension is evident in all tested scenarios.

Acknowledgement This work was supported by the DFG, grant GRK 1653.

References

1. Bertsekas, D.P.: A distributed algorithm for the assignment problem. Tech. rep., Lab. for Information and Decision Systems Report, MIT (May 1979)
2. Bertsekas, D.P.: The auction algorithm: A distributed relaxation method for the assignment problem. Annals of Operations Research 14, 105–123 (1988)
3. Bertsekas, D.P., Eckstein, J.: Dual coordinate step methods for linear network flow problems. Mathematical Programming, Series B 42, 203–243 (1988)
4. Bertsekas, D., Castanon, D.: The auction algorithm for the transportation problem. Annals of Operations Research 20, 67–96 (1989)

N	500	1000	2000	4000	6000	500	1000	2000	4000	6000
	time					queries				
	/s	/10s	/10^2s	/10^3s	/10^3s	/10^7	/10^7	/10^8	/10^9	/10^{10}
	2.81	2.00	1.93	1.98	6.54	2.31	12.00	6.63	4.06	1.10
P2H	1.06	0.53	0.31	0.20	0.73	0.41	1.51	0.40	0.15	0.04
	1.38	0.59	0.38	0.21	0.74	0.57	1.79	0.57	0.18	0.04
		1.16	1.07	0.97	6.54		7.52	3.88	1.93	0.52
P3H		0.31	0.20	0.17	0.99		1.93	0.63	0.20	0.04
		0.42	0.34	0.24	1.41		2.90	1.30	0.34	0.07
	3.04	1.98	1.58	1.00	3.58	2.49	12.30	6.09	2.63	0.67
P2H-P1	1.05	0.49	0.24	0.10	0.31	0.53	2.30	0.61	0.17	0.05
	1.34	0.54	0.29	0.11	0.32	0.69	2.54	0.77	0.19	0.05
	3.64	2.92	2.14	2.35	7.68	2.55	13.20	7.02	4.51	1.21
P2I	1.32	0.67	0.38	0.34	1.14	0.46	2.22	0.43	0.21	0.03
	1.69	0.73	0.45	0.35	1.33	0.62	2.49	0.60	0.24	0.04
	54.70	19.60	19.30	5.92	13.20	56.00	185.00	162.00	46.90	9.91
grid	9.46	1.67	0.94	0.15	0.26	22.90	22.90	26.70	2.00	0.46
	9.53	1.76	0.95	0.16	0.27	23.00	23.70	26.80	2.15	0.47
		21.30	13.10	9.97	N/A		150.00	86.90	49.60	N/A
mesh		2.95	1.55	0.88	10.9		45.40	17.60	9.78	8.52
		3.18	1.84	0.93	11.1		48.40	22.20	10.60	8.70
	2.75	1.82	2.00	1.93	9.04	2.26	11.60	6.74	4.07	1.15
P2H-LB	1.10	0.52	0.35	0.24	1.35	0.41	1.63	0.46	0.20	0.03
	1.57	0.57	0.38	0.25	1.41	0.65	1.97	0.62	0.23	0.04

Table 1: Summary of numerical experiments for the scenarios introduced in Sect. 6 and various problem sizes. N gives the (in all experiments equal) cardinality of X and Y. For each scenario the **first row** gives the results of the dense algorithm, where 'queries' gives the number of pairs checked for creating the lists $\Pi(x)$, (11), throughout the algorithm. The **second row** gives the results of the hybrid algorithm, only at the finest scale. Here 'queries' gives the number of checks for creating all $\Pi(x)$ plus the number of hierarchical consistency checks. The **third row** gives the results for the hybrid algorithm summed over all scales, i.e. for solving the whole problem from scratch. All results are averaged over multiple instances.

In all scenarios the number of queries is reduced significantly by the hybrid variant, resulting in a corresponding runtime decrease. For some scenarios the runtime ratio full/hybrid slightly decreased from $N = 4000$ to $N = 6000$. We attribute this to the changing relation of problem size to hierarchy depth, the effects of which have yet to be more carefully examined. We expect the ratio to increase again for $N > 6000$.

5. Carlier, G., Galichon, A., Santambrogio, F.: From Knothe's transport to Brenier's map and a continuation method for optimal transport. SIAM J. Math. Anal. 41, 2554–2576 (2010)
6. Haker, S., Zhu, L., Tannenbaum, A., Angenent, S.: Optimal mass transport for registration and warping. Int. J. Comput. Vision 60, 225–240 (December 2004)
7. Kuhn, H.W.: The hungarian method for the assignment problem. Naval Research Logistics 2, 83–97 (1955)
8. Mémoli, F.: Gromov-Wasserstein distances and the metric approach to object matching. Found. Comp. Math. 11, 417–487 (2011)
9. Pele, O., Werman, W.: Fast and Robust Earth Mover's Distances. In: Proc. Int. Conf. Comp. Vision (ICCV) (2009)
10. Schrijver, A.: Combinatorial Optimization: Polyhedra and Efficiency, Algorithms and Combinatorics, vol. 24. Springer (2003)
11. Shirdhonkar, S., Jacobs, D.: Approximate earth mover's distance in linear time. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on (2008)
12. Villani, C.: Optimal Transport: Old and New. Springer (2009)
13. Wang, W., Slepčev, D., Basu, S., Ozolek, J.A., Rohde, G.K.: A linear optimal transportation framework for quantifying and visualizing variations in sets of images. International Journal of Computer Vision 101, 254–269 (2012)