



The Shortlist Method for Fast Computation of the Earth Mover's Distance and Finding Optimal Solutions to Transportation Problems

Carsten Gottschlich^{1,2*}, Dominic Schuhmacher¹

¹ Institute for Mathematical Stochastics, University of Göttingen, Göttingen, Germany, ² Felix-Bernstein-Institute for Mathematical Statistics in the Biosciences, University of Göttingen, Göttingen, Germany

Abstract

Finding solutions to the classical transportation problem is of great importance, since this optimization problem arises in many engineering and computer science applications. Especially the Earth Mover's Distance is used in a plethora of applications ranging from content-based image retrieval, shape matching, fingerprint recognition, object tracking and phishing web page detection to computing color differences in linguistics and biology. Our starting point is the well-known revised simplex algorithm, which iteratively improves a feasible solution to optimality. *The Shortlist Method* that we propose substantially reduces the number of candidates inspected for improving the solution, while at the same time balancing the number of pivots required. Tests on simulated benchmarks demonstrate a considerable reduction in computation time for the new method as compared to the usual revised simplex algorithm implemented with state-of-the-art initialization and pivot strategies. As a consequence, the Shortlist Method facilitates the computation of large scale transportation problems in viable time. In addition we describe a novel method for finding an initial feasible solution which we coin *Modified Russell's Method*.

Citation: Gottschlich C, Schuhmacher D (2014) The Shortlist Method for Fast Computation of the Earth Mover's Distance and Finding Optimal Solutions to Transportation Problems. PLoS ONE 9(10): e110214. doi:10.1371/journal.pone.0110214

Editor: Haipeng Peng, Beijing University, China

Received: June 10, 2014; **Accepted:** September 11, 2014; **Published:** October 13, 2014

Copyright: © 2014 Gottschlich, Schuhmacher. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability: The authors confirm that all data underlying the findings are fully available without restriction. Data are available from the Dryad Digital Repository: <http://doi.org/10.5061/dryad.k30sg>.

Funding: CG and DS gratefully acknowledge support by the German Research Foundation and the Open Access Publication Funds of the University of Göttingen. CG gratefully acknowledges the support of the Felix-Bernstein-Institute for Mathematical Statistics in the Biosciences and the Volkswagen Foundation. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* Email: gottschlich@math.uni-goettingen.de

Introduction

Finding solutions to the classical transportation problem is of great importance, since this optimization problem arises in various guises in many real world and theoretical situations. They occur as subproblems in larger problems, e.g. the warehouse location problem or the traveling salesperson problem and also in a variety of engineering and computer science applications, such as content based image retrieval [1], automatic scene analysis [2] or for the discrimination between real and artificial fingerprints [3]. A more extensive discussion of such applications is given in Section Applications of the Transportation Problem.

The problem was first described by Monge in 1781 [4] in somewhat different form and has been analyzed by many researchers including Kantorovich, Hitchcock, Koopmans and especially Dantzig [5,6], the father of the simplex algorithm. The solution of this problem is the fundamental ingredient for computing the Earth Mover's Distance [1] in computer science and the Wasserstein distance, also known as Mallows or Kantorovich distance in statistics and physics, see Chapter 6 in [7].

In order to give a quick and intuitive description of the various facets of the transportation problem and the revised simplex algorithm we often use an economic interpretation, which of course will not reduce the scope of the described algorithms and

their applications in any way. The problem can be summarized as follows.

Consider a consortium of m production and n consumption facilities of a certain good. For simplicity these are also referred to as origins and destinations. Suppose that there is a certain supply of $a_i > 0$ available at origin i , and there is a certain demand of b_j at destination j . The cost for transporting a unit of the good from i to j shall be given by arbitrary $c_{ij} \in \mathbb{R}$. Borrowing the illustration from Chapter 3 in [7], the production facilities might be Parisian bakeries cooperating with cafés (consumption facilities), where the good transported are baguettes, and the cost incurred is the actual transportation cost. It is assumed that total supply equals total demand, i.e. $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$. The objective is then to determine a transportation plan $X = (x_{ij})_{1 \leq i \leq m, 1 \leq j \leq n}$ such that all producers and consumers are satisfied and that the total cost is minimized. In other words

$$\text{minimize } \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (1)$$

$$\text{subject to } \sum_{j=1}^n x_{ij} = a_i \quad \text{for } i=1, \dots, m, \quad (2)$$

$$\sum_{i=1}^m x_{ij} = b_j \quad \text{for } j=1, \dots, n, \quad (3)$$

$$x_{ij} \geq 0 \quad \text{for all } i, j. \quad (4)$$

A dual formulation can be obtained as follows. Suppose that a carrying company offers to take over the good from the consortium for a price of $u_i \in \mathbb{R}$ per unit at origin i and to hand it back at destination j for a price of $v_j \in \mathbb{R}$ (any prices may be negative). In order for the carrier to be competitive, it needs to set prices u_i, v_j so that $u_i + v_j \leq c_{ij}$ for all $1 \leq i \leq m, 1 \leq j \leq n$. Following [8] we refer to the difference $r_{ij} = c_{ij} - u_i - v_j$ as *relative cost* incurred when the consortium takes over the transportation from i to j itself rather than commissioning the carrier. The carrier would like to maximize its profit $\sum_{i=1}^m a_i u_i + \sum_{j=1}^n b_j v_j$ subject to the price constraint. Standard duality theory, e.g. Chapter 4 in [8] relates the solutions of the two problems to one another (provided one of them exists) and shows that the optimal values of the objective functions are the same.

The rest of the paper is organized as follows. In the next section, we first give a non-technical description of the revised simplex algorithm for solving the transportation problem; for a more detailed presentation see [8]. Then we discuss crucial aspects in various subsections, starting with pivot strategies, and passing from cycle finding to treating initialization methods. Next we introduce the new Shortlist Method for solving the transportation problem. Benchmark tests reported in the section simulation results clearly show the advantage of the proposed method over the existing ones. We conclude with a discussion of the results and review relevant application scenarios.

The Transportation Algorithm

Using the simplex approach the transportation algorithm consists of two stages: first, an initial transportation plan X is constructed such that Equations (2–4) are satisfied. Second, the initial plan is iteratively improved until the optimal solution is obtained.

At any time the current feasible plan consists of $m+n-1$ “active” origin/destination pairs (i, j) between which a positive amount x_{ij} is transported (in a degenerate case there might be pairs with zero amount, but we exclude this case in our description). We will refer to them as basis pairs or basis entries.

For each iteration in the second stage a basis entry is replaced by a “better one”. For this we first compute the “dual” prices u_i and v_j . In the context of the simplex method, these are also known as simplex multipliers. Starting with an arbitrary value, e.g. setting $u_1 = 0$, all other prices are determined by solving the equations $u_i + v_j = c_{ij}$, where (i, j) are basis entries. A property well-known as basis triangularity sees to it that every origin and every destination gets a price assigned in this way.

A new basis entry is then selected as a so-called pivot element by finding a non-basis pair (i, j) that has negative relative cost $r_{ij} = c_{ij} - u_i - v_j$, meaning that the consortium can transport goods

more cheaply from i to j by itself than by commissioning the carrier.

Next, a cycle of changes starting in (i, j) is determined by alternately scanning rows and columns for basis entries until a cycle is complete, which again is bound to happen by basis triangularity. Assuming that all amounts $x_{i'j'}$ at basis entries are positive (the non-degenerate case), there is a maximal positive amount θ which we can alternately add and subtract from the values $x_{i'j'}$ when following the cycle, starting with addition for the first value x_{ij} . Since the cycle alternates between following rows and columns, the procedure preserves Equations (2–3).

After this, one of the $x_{i'j'}$ has been reduced to 0 and we remove the corresponding pair (i', j') from the basis (if several values have been reduced to zero, we remove the first such entry, but are then dealing with a degenerate case). The basis still has exactly $n+m-1$ entries, and we proceed with the next iteration, continuing until there are no entries with negative relative cost any more. In this case we have reached an optimum.

Pivot Strategies

When selecting a pivot element to enter the basis, all non-basis entries with relative cost $r_{ij} < 0$ are candidates. According to Dantzig's criterion, the most negative one is chosen. To the best of our knowledge it is an open question whether a better criterion for selecting one of these candidates can be formulated in order to minimize the number of pivot operations until optimality is reached.

If the algorithm is applied to solve real-world transportation problems, the goal of a practical implementation is typically to minimize the runtime on a computer. Our analysis has shown that *two* key factors determine the runtime: the number of pivot operations and the number of elements for which relative costs are computed in order to select pivot elements.

The former can be made small by computing the relative costs for all non-basis entries which in turn maximizes the latter (“matrix most negative” strategy). The other extreme is to perform the pivot operation immediately after discovering the first candidate (“first negative” strategy). In this way, the second factor is minimized at the cost of an increase of the first. A more balanced strategy is to compute the relative costs for all non-basis entries of a row and then choose the most negative among these candidates (“modified row most negative strategy”) or go on with the subsequent row, if no candidate has been discovered. In the next iteration of the algorithm, continue with the first row not considered in the previous one. The latter strategy outperformed the others in our tests, which corroborates earlier findings reported by [9] and by [10].

Finding Cycles

The procedure of finding cycles of changes can be translated into a depth-first search (DFS) [11] on the following directed graph (see also Figure 1): Each basis entry corresponds to two vertices: one vertex with the basis entries in the same row as incoming edges and the basis entries in the same column as outgoing edges, and a second vertex with the basis entries in the same column as incoming edges and the basis entries in the same row as outgoing edges. The graph is weakly connected, acyclic and bipartite. By adding the (two copies of the) pivot element, the graph becomes cyclic and DFS is an efficient method for discovering the (up to mirroring) unique cycle. Since each basis entry is connected to all other basis entries in the same row and the same column, no other data structure is needed to store the graph than a list of basis entries for each row and for each column.

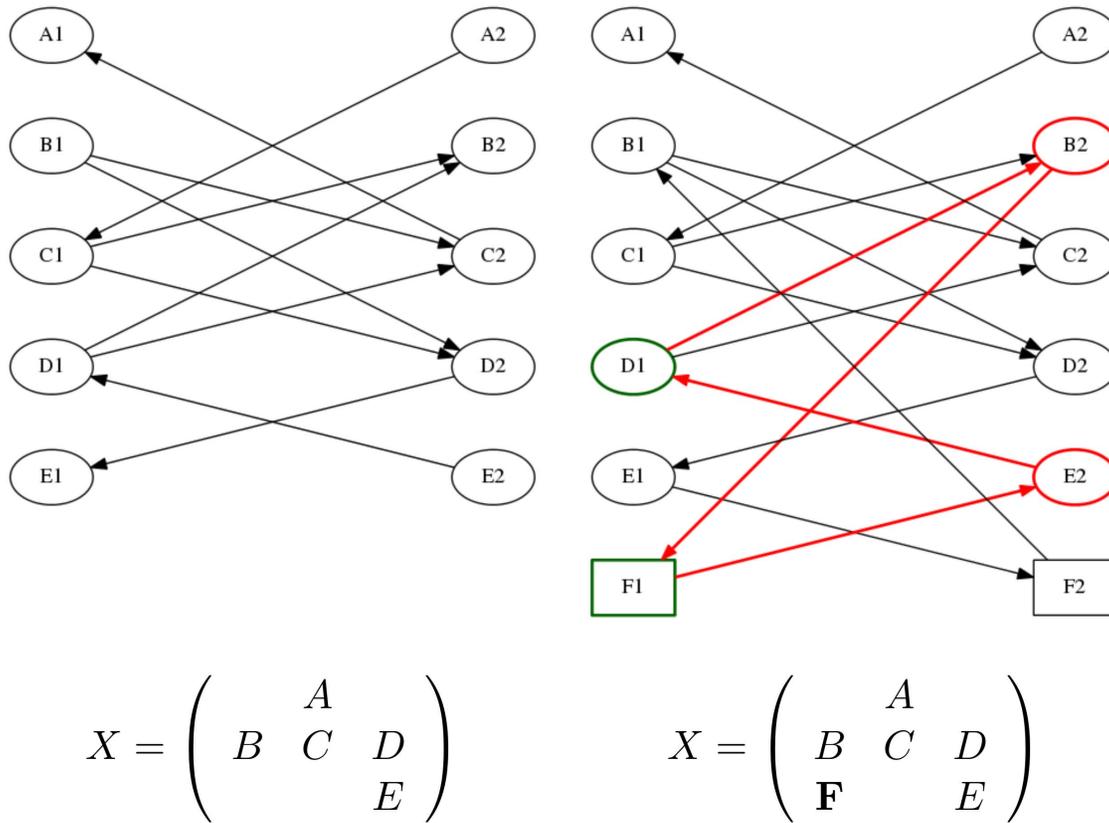


Figure 1. Each graph corresponds to the transport plan shown below. Directed edges are drawn with arrows. The direction from left to right indicates a ‘same row’ relation between basis entries, right to left shows a ‘same column’ relation. The graph on the left becomes cyclic by adding the pivot element F to the basis (right).
doi:10.1371/journal.pone.0110214.g001

Considering the example shown in Figure 1, we begin with the transport plan and graph on the left. Next, we insert F as pivot element (right) and discover the cycle starting in F1 with depth-first search. Along the cycle, the minimum of all nodes on the right side of the graph determines the amount of change θ which is subtracted from B and E (red) and added to D and F (green). One of the two elements B or E will leave the basis. F2 was not required during the pivot operation, but alternatively it would have been possible to use the complementary cycle F2 → B1 → D2 → E1 → F2 instead, leading to the same result. Basis elements A and C remain unchanged during this pivot operation.

Initialization Methods

In the subsequent comparison of methods for constructing an initial feasible solution (stage one in the transportation algorithm), we take the following established procedures into account. If a method generates fewer than $m+n-1$ basis entries (degenerate case), we complement them by adding the right number of entries (i, j) in such a way that all basis entries are connected, i.e. there are other basis entries in the same row or the same column, but no cycles are formed and their values x_{ij} remain zero.

Northwest Corner Rule. Suppose we list all origins from $i=1$ to $i=m$ as rows and all destinations from $j=1$ to $j=n$ as columns. This rather naive rule starts in the top left corner and allocates the maximum possible amount to x_{11} , i.e. the minimum of a_1 and b_1 . If there remains supply at origin 1, we move to the right and assign to x_{12} maximum possible amount. Otherwise if the demand at destination 1 was larger than the supply, we move

one cell down and continue with assignment x_{21} . And in case that a_1 is equal to b_1 , we move directly to x_{22} . In this way, we iterate over all origins and destinations, and we obtain a solution satisfying Equations (2–4).

Least Cost Rule or Matrix Minimum Rule. This simple rule determines in each iteration the minimum cost entry c_{ij} among all origins with remaining supply and among all destinations with remaining demand, and assigns the maximum possible amount to x_{ij} until all requirements are met.

Houthakker’s Method of Mutually Preferred Flows. The idea of Houthakker’s mutually preferred flows [12] is somewhat similar to the least cost rule. For all origins that have any supply left, the minimum cost c_{ij} of the corresponding row is determined, and likewise for all destinations that have any demand left, the minimum cost c_{ij} of the corresponding column is detected. If an entry (i, j) is both row and column minimum, the maximum feasible amount is assigned to x_{ij} . A difference to the least cost method is that more than one entry can enter the basis in each iteration.

Vogel’s Approximation Method. The basic idea of Vogel’s approximation method [13] is to compute the opportunity costs: for each not yet exhausted origin and for each remaining destination, take the difference between its smallest cost and its second smallest cost. This idea is also the key ingredient for computing bids and raising prices in the auction algorithm [14]. In each iteration of Vogel’s approximation method, the row or column with the maximum opportunity cost is selected and for the

minimum c_{ij} in that row or column, the maximum possible value x_{ij} is allocated.

Russell's Method. Russell [15] proposed an approach to approximate Dantzig's criterion. In each iteration denote by I the set of origins i that have any supply left and by J the set of destinations j that have any demand left. Then determine $w_i = \max_{j \in J} c_{ij}$ for every $i \in I$ and $y_j = \max_{i \in I} c_{ij}$ for every $j \in J$. The quantities w_i and y_j are supposed to approximate the simplex multipliers u_i and v_j (see Section 0). Using these estimates, Russell computes in each iteration $(i, j) = \arg \min_{i \in I, j \in J} (c_{ij} - w_i - y_j)$ and allocates the maximum possible amount to x_{ij} .

Modified Russell Method. In this paper, we propose a modification of Russell's method which outperforms the original version on our benchmarks: instead of updating w_i and y_j , we compute these values once at the start. Next, we compute a cost matrix D with $d_{ij} = c_{ij} - w_i - y_j$ and then, we apply the least cost rule to this matrix D . The proposed modification saves a lot of computational time in each iteration by not updating w_i and y_j and performs much better in comparison to the original Russell method.

Weighted Frequency Method. Eight years before Russell, Habr [16] proposed a related method which he called weighted frequency method. Let mr_i be the mean cost of row i and mc_j the mean cost of column j . According to Habr's method, we define a matrix F with cost entries $f_{ij} = c_{ij} - mr_i - mc_j$. The transportation plan is established by choosing x_{ij} in each iteration pursuing the matrix minimum rule applied to F and assigning the maximum possible amount to x_{ij} . Habr provides a nice theoretical justification for his method: suppose for each possible entry (i, j) we consider each possible combination (r, s) with $r \neq i$ and $s \neq j$. The question whether it is beneficial to include x_{ij} in the transportation plan is answered by comparing the costs $c_{ij} + c_{rs}$ with the costs $c_{is} + c_{rj}$ for all combinations (r, s) . Habr showed that summing up the differences $c_{ij} + c_{rs} - c_{is} - c_{rj}$ over all possible combinations is equivalent (up to a constant) to computing the matrix F .

Row Minimum Rule and Modified Row Minimum Rule. These two rules [10] iterate over the rows (origins) and determine for each row i the column (destination) with positive unassigned demand b_j which has the minimum transportation cost c_{ij} . The difference between both rules is that modified row minimum rule assigns at most one entry x_{ij} per row and then resumes with the next row. The row minimum rule in contrast repeatedly determines the minimum for row i until the supply of origin i is completely distributed and only then it continues with the next row.

Column Minimum Rule and Modified Column Minimum Rule. These two rules work exactly as the two previous described methods with rows and columns exchanged.

Alternating Row Column Minimum Rule. This initialization method combines the modified row minimum rule and the modified column minimum rule by alternating between rows and columns.

Two Smallest in Row Rule. The two smallest in row rule [9] can be regarded as a variant of the modified row minimum rule that assigns two instead of one entries per row and iteration.

The Shortlist Method

As described in the previous section, the simplex-based transportation algorithm consists of two stages: an initialization phase to find a feasible solution and a convergence phase in which the current solution is iteratively improved to optimality. The *Shortlist Method* introduces an additional phase in between these

two. The main steps of the Shortlist Method can be outlined as follows:

1. A shortlist is created for each origin containing only a small fraction of all possible destinations.
2. An initial feasible transportation plan is derived from these shortlists (for an example see Figure 2, left).
3. The transportation plan is improved towards optimality based on the shortlists.
4. The transportation plan is improved to global optimality based on the complete matrix (for an example see Figure 2, right).

The crucial part is the third step in which the shortlist search for a new basis entry balances the computational burden between the number of elements for which relative costs are calculated and the number of pivot operations performed.

More precisely the Shortlist Method uses as parameters the length s of the shortlists and two decision criteria k and p . The four steps are carried out as follows.

At the beginning, for each origin i , a list of s destinations with the lowest transportation costs is created, containing the index j of the destination and the corresponding costs. This shortlist is sorted in ascending order according to costs by QuickSort [11].

Next, we iterate over all not yet exhausted origins i and assign the maximum feasible amount to x_{ij} with the smallest costs c_{ij} among all destinations j in the shortlist of i . If no such destination is available any more, the minimum over the remaining j is chosen. The latter is usually only necessary for very small shortlist lengths.

In the third phase, we improve the transportation plan X iteratively considering batches of consecutive shortlists. Starting from the first shortlist not considered in the previous iteration, we compute relative costs $r_{ij} = c_{ij} - u_i - v_j$ for non-basis entries until k candidate entries with negative r_{ij} have been discovered or p percent of all shortlists have been searched. Then the batch ends. We choose the entry with the most negative relative cost for performing a pivot operation, i.e. we add the entry to the basis, compute a cycle of changes and remove another entry from the basis as detailed in Section. Then we go the next iteration. Whenever the last shortlist has been used, we continue by reusing the first one. If at any point no more candidates are discovered, phase three is terminated.

In the final phase, complete rows are searched instead of shortlists and if a row contains at least one candidate, the most negative one is chosen; i.e. we perform the simplex-based transportation algorithm as described in Section with the 'modified row most negative' pivot strategy until the optimum is reached.

Simulation Results

In order to evaluate the performance of the described initialization methods as a function of the number of origins and destinations, a benchmark was generated in the following way: On an empty grid of size 512×512 , the x - and y -coordinates of locations for n origins and n destinations were chosen independently and uniformly at random while avoiding double allocations. Amounts a_i and b_j were chosen independently and uniformly at random between 0 and 255. A final adjustment step ensures the equality of the sum over all a_i and the sum over all b_j . The cost matrix C contains as entry c_{ij} the Euclidean distance between origin i and destination j . 100 examples are generated for each number n of origins and destinations from 100 to 3000 in steps of 100.

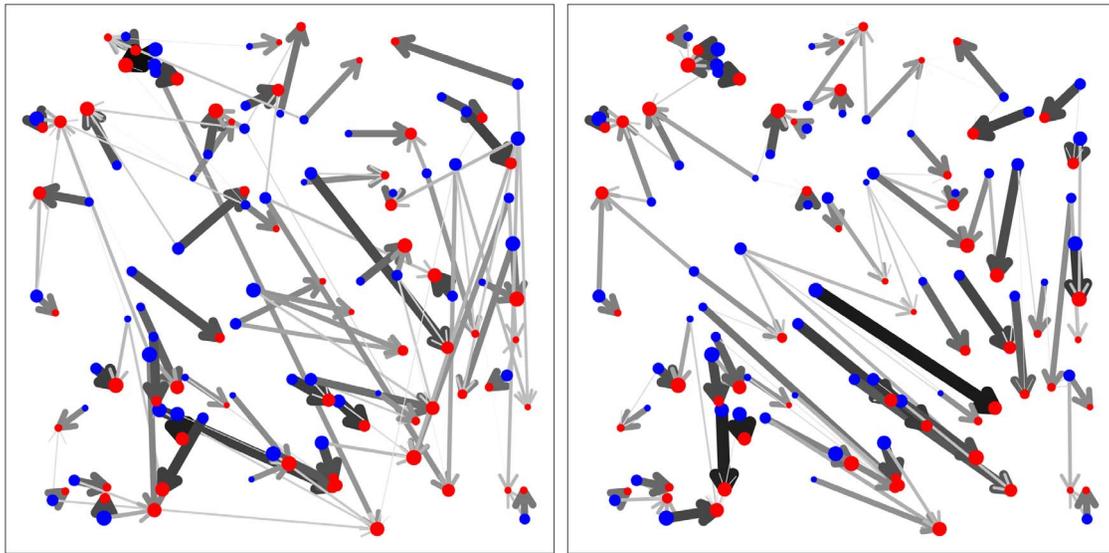


Figure 2. Visualization of a transportation plan for a very small example with 60 origins (blue) and 60 destinations (red) at the end of phase two (left; initial feasible solution derived from shortlists) and at the end of phase four (right; global optimal solution). The diameters of the circles correspond to the mass at these origins and destinations. A greater width and darker color of arrows indicates a larger amount of mass being transported. doi:10.1371/journal.pone.0110214.g002

We make the generated transportation problem examples available for download, so that other researchers can reproduce the results and test other methods on the same benchmark. Data are available from the Dryad Digital Repository: <http://doi.org/10.5061/dryad.k30sg>

An implementation of the shortlist method is provided as part of the R package `transport`: <http://cran.r-project.org/web/packages/transport/>

All initialization methods were implemented to the best of our knowledge and optimal solutions were computed using the same revised simplex implementation for all methods. In Figure 3, we report total runtimes including the runtime for finding an initial basis and the runtime for the simplex iterations. The total runtimes are averaged over the 100 examples for each n . The implementations are written in Java and were tested using one core of an Intel Core i7 CPU with 3.20 GHz.

We observe that the Shortlist Method outperforms the other methods by a rather large margin. While for other initialization methods it is clearly preferable to use the “modified row most negative pivot strategy” (compare the remark in Subsection), this makes hardly a difference for the Shortlist Method. We may attribute this to the fact that this choice of the pivot only enters in step 4 of the Shortlist Method. However, by the end of step 3 the solution is already so close to optimality that step 4 does not have much influence on the total computation time.

The aforementioned parameters of the Shortlist Method were chosen in the following way. An additional set of examples was created with 30 examples for each n . For each parameter, a set of a few possible choices were defined, and in total about thirty of their combinations were used for computing initial bases and optimal solutions on this training set. In this way, we obtained the following rough rule of thumb:

Shortlist length: $s := 15$ for $n \leq 200$, then an increase of s by another 15 for each doubling of n . More precisely, $s := 15 + \lfloor 15 \cdot \log_2(n/200) \rfloor$ for $n > 200$.

Stop criteria: (i) $k := s$ candidates. (ii) $p := 5\%$ of shortlists are searched at most in one iteration.

Although these parameter values have been trained, we consider them to be rather ad hoc, as they were chosen informally and by considering a few choices only. We understand this as a proof of concept of the Shortlist Method and as a first step towards determining good universal parameters that only depend on the problem size. There are clearly situations, where one has the opportunity to train the method to more specific features of the problem at hand, e.g. when comparing images to a larger database. Then we expect our method to perform even considerably better than suggested by the above simulations.

Table 1 gives a comparison of our implementation of the shortlist method with two other programs: the original C code by Rubner used via the R package `emd` [17] and `lp_solve` [18] by Berkelaar and others, a general purpose mixed integer linear programming solver (which accounts to some extent for its long runtime).

Last but not least, let us note that we have also compared the different approaches on various collections of real and randomly generated images, and the respective performances were largely confirmed.

Discussion

The results for various problem sizes presented in the previous section demonstrate the potential of the novel approach. The Shortlist Method outperforms all the other methods on the considered benchmark and the curves in Figure 3 and 4 suggest that the performance difference increases with increasing problem size.

To substantiate this conjecture we have simulated additional sets of 10 examples for each of the six best performing methods in the lower panel of Figure 3 in combination with each of the problem sizes $n = 400, 800, 1600, 3200, 6400,$ and 12800 . Based on the literature we have expected polynomial growth of the time complexity of the problem with an exponent that is somewhat below 3, i.e. a runtime of roughly the form $r = cn^q$ for some $q \in [2,3]$. Since this implies that $\log(r) = \log(c) + q \log(n)$ one

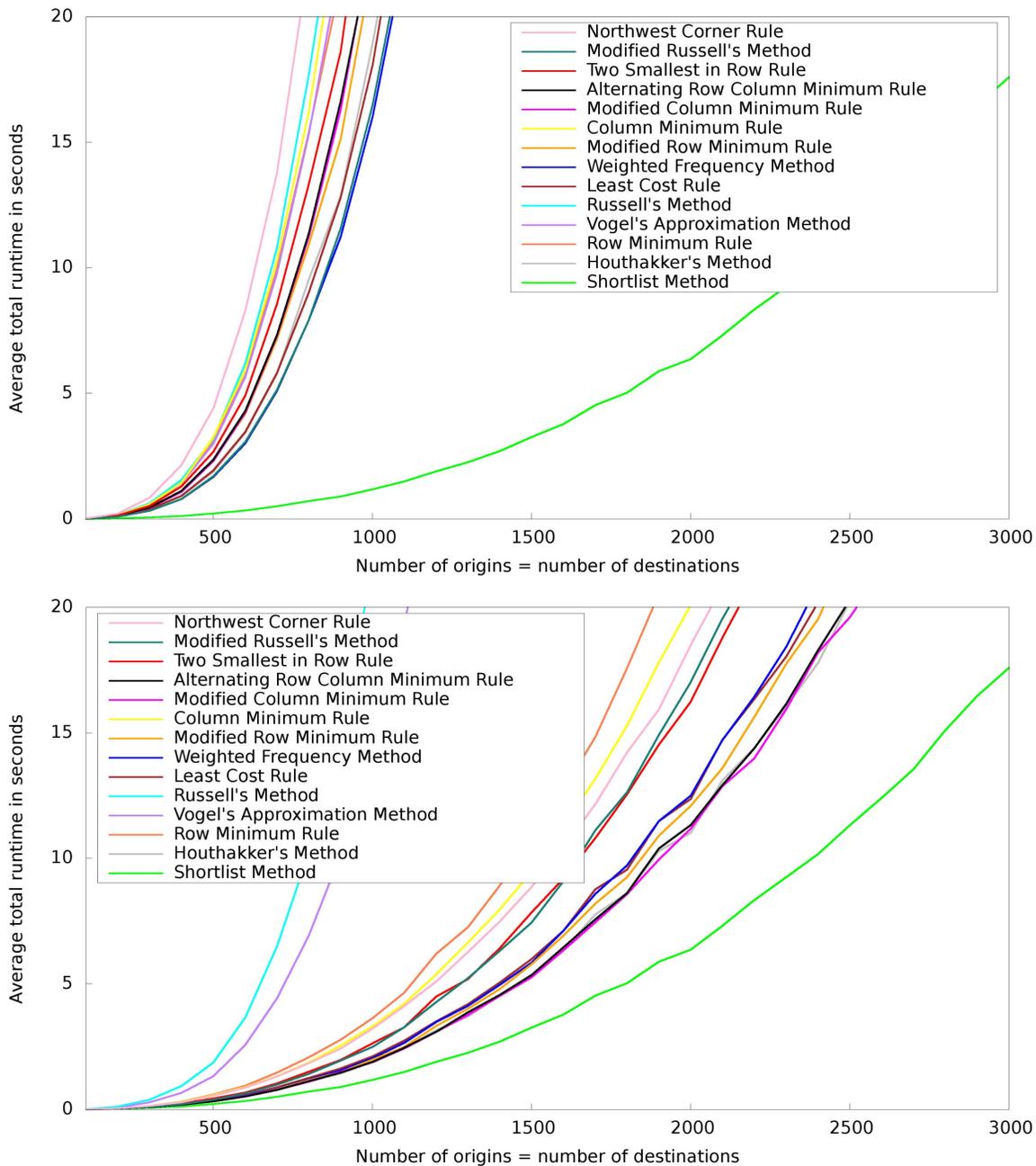


Figure 3. Comparison of the Shortlist Method to other methods. Depicted are total runtimes in seconds (for each method and each number of origins averaged over 100 solved transportation problems) for various initialization methods from the literature combined with one of two pivot strategies: matrix most negative (top) and modified row most negative (bottom). The total runtime encompasses the runtime for finding an initial basis and the runtime for the simplex iterations.
doi:10.1371/journal.pone.0110214.g003

can expect a roughly linear relation, when drawing the logarithm of the runtime as a function of the log problem size.

As we can see from Figure 4 this idea works out quite well. We only plot the results for the Shortlist Method and two competing methods as the other four competitors would overlap large parts of the two that are given. The circles indicate the results from our simulations, whereas the lines have been fitted by least-squares regression. Note that the lines fit the simulation data very well. The slopes of the lines provide estimates for the exponents q . These are given numerically in Table 2 for all seven methods, together with p-values for testing whether the slope is different

from the q obtained for the Shortlist Method. The p-values are based on statistical tests for comparing slopes in an ANCOVA model, see [19, Chapter 13]. Since they are so small, it seems highly likely that the Shortlist Method has in fact a better time complexity than the other methods.

Let us also compare the performances of the best six competitors for our original benchmark to earlier performance studies from the literature. Based on the results considered in the lower panel of Figure 3, i.e. based on problem sizes up to 3000, several initialization methods performed similarly: the modified column minimum rule, Houthakker's method and the alternating

Table 1. Comparison of the shortlist method with `lp_solve` [18] and `emdlist` [17]. Runtimes in seconds averaged over 100 solved transportation problems.

| Problem size | Method | | |
|--------------|----------------------------|---------------------------|-----------|
| | <code>lp_solve</code> [18] | <code>emdlist</code> [17] | Shortlist |
| 100 × 100 | 0.1360 | 0.0616 | 0.0054 |
| 200 × 200 | 1.1839 | 0.1507 | 0.0246 |
| 300 × 300 | 4.3854 | 0.5705 | 0.0634 |
| 400 × 400 | 10.6491 | 1.8974 | 0.1245 |
| 500 × 500 | 22.4806 | 4.8668 | 0.2254 |
| 600 × 600 | 40.4955 | 9.0441 | 0.3525 |
| 700 × 700 | 67.5250 | 17.0948 | 0.5269 |
| 800 × 800 | 104.1458 | 28.5478 | 0.7411 |
| 900 × 900 | 145.6244 | 42.1987 | 0.9436 |
| 1000 × 1000 | 203.5568 | 62.3756 | 1.2314 |

doi:10.1371/journal.pone.0110214.t001

row column minimum rule, followed by the modified row minimum rule, the least cost rule and the weighted frequency method.

These results confirm earlier findings reported in [9] and in [10] on other benchmarks, with one exception: the least cost rule (also known as matrix minimum rule) performed among the best competitors in our test and finished among the slowest methods in [9]. A possible explanation is our implementation which sorts all matrix entries once in ascending order by transportation costs and

then iterates over the list until the initial solution is obtained. This procedure is more efficient than determining the matrix cost minimum in each iteration by scanning all remaining origins and destinations. Analogously, we can explain the advantage of the proposed modified Russell's method over the original Russell's method. The speedup gained by the avoidance of scanning large parts of the complete matrix in each iteration clearly outweighs a possible quality loss of the initial solution by not updating the quantities w_i and y_j which are supposed to approximate the

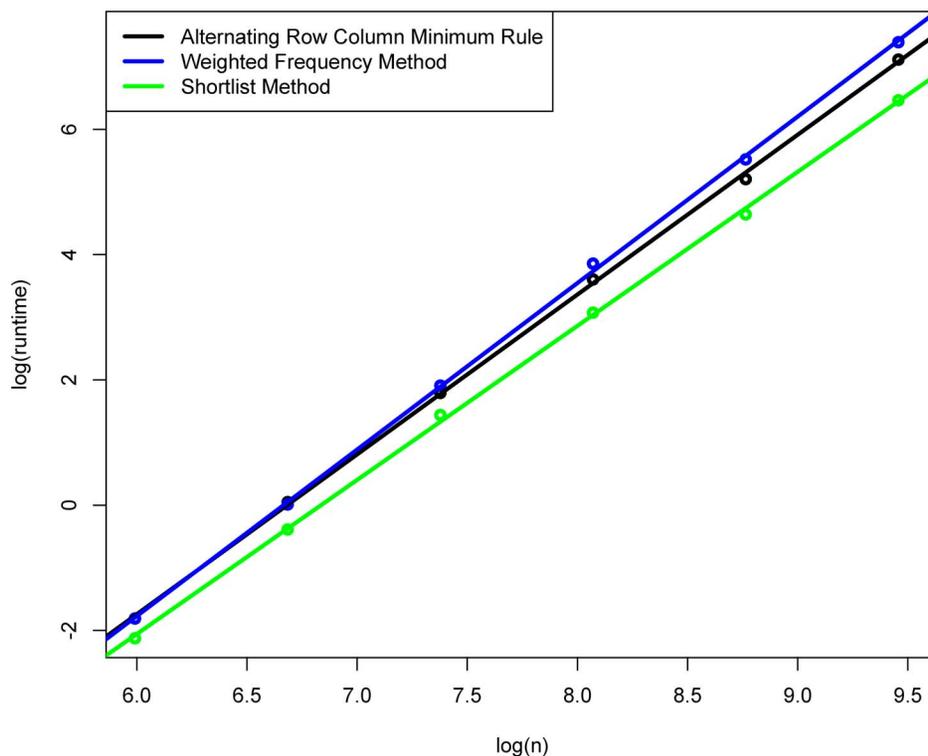


Figure 4. Comparison of the Shortlist Method to two main competitors. Depicted as circles are the logarithms of total runtimes in seconds (for each method and each number of origins averaged over 10 solved transportation problems) depending on the logarithm of the problem size (circles). The lines have been fitted by least-squares regression.

doi:10.1371/journal.pone.0110214.g004

Table 2. We assume a relation of $r = cn^q$ between computation time r and problem size n .

| Method | factor [$\cdot 10^{-8}$] | exponent | p-value | signif. |
|-----------------------------------|----------------------------|---------------|-------------|---------|
| Shortlist Method | 5.0026 | 2.4591 | — | |
| Alternating Row Col. Minimum Rule | 3.9526 | 2.5510 | 0.009090 | ** |
| Modified Column Minimum Rule | 3.4778 | 2.5667 | 0.002624 | ** |
| Modified Row Minimum Rule | 3.1825 | 2.5915 | 0.000312 | *** |
| Least Cost Rule | 2.3954 | 2.6362 | 0.000005 | *** |
| Weighted Frequency Method | 2.0282 | 2.6574 | $< 10^{-6}$ | *** |
| Houthakker's Method | 1.7119 | 2.6594 | $< 10^{-6}$ | *** |

Shown are estimates of the factor c (to be multiplied by 10^{-8}) and the exponent q , together with p-values for the comparison of exponents for the Shortlist versus other methods. Significance levels correspond to the usual classification: $0 < *** < 0.001 < ** < 0.01 < * < 0.05 < \cdot < 0.1$.

doi:10.1371/journal.pone.0110214.t002

simplex multipliers u_i and v_j (see The Transportation Algorithm Section).

Further research includes a systematic large-scale simulation study to determine good universal parameter settings depending only on easy-to-determine features of the problem such as problem size. Also we would like to investigate to what extent computation times and orders of complexity can be improved when comparing images within a homogeneous database, where one has the possibility to train the parameters to the expected type of transportation problem.

In either case we believe that there is still much room for improvement of the results obtained above. We expect these findings to prepare the ground for applications in pattern recognition, computer vision and image processing, where solving the transportation problem has so far been considered as intractable due to the problem size and the runtimes of existing methods when applied to (smaller) raw gray scale images or features like curved Gabor filter bank responses [20] or histograms of invariant gradients (HIG) descriptors [21]. A selection of further applications is contained in the next section.

Applications of the Transportation Problem

Solving transportation problems efficiently is of great importance in many different fields of application. We would like to give an idea of the relevance of fast algorithms by discussing a selection of specific examples.

Detection of Phishing Web Pages

The earth mover's distance (EMD) has been applied for the detection of phishing web pages by Fu *et al.* [22]. Screenshots are taken from banking websites and potential phishing sites and the visual similarity is measured using the EMD. If an anti-phishing system automatically compares thousands or millions of websites, the speed of each comparison is an important factor and can become the bottleneck of the system. In this application scenario, the speedup by the shortlist method can make a huge difference. E.g. if web sites are compared at a resolution of 100×100 pixels, this corresponds to a problem with an approximate dimension of 5000 origins and 5000 destinations.

Linguistics

The EMD has been applied as a measure of dissimilarity when comparing the distribution of color names among 110 different languages [23]. Notably, computation of EMDs for 2300 language vectors took the authors about one week using an industrial

strength LP solver [24]. Due to the computational complexity, they refrained from evaluating the 23,982 speaker response vectors.

Content-based Image Retrieval

Since the early days of retrieving images from large databases, the EMD has been applied for comparing histograms and signatures [1]. Pele and Werman proposed a thresholded ground distance which is an EMD variant [25]. For content-based image retrieval, thresholding the ground distance has a positive effect on the retrieval accuracy [26].

Fingerprint Recognition

In the area of fingerprint recognition, the EMD has been applied for discriminating between real and synthetic fingerprint images based on minutiae histograms [3]. These 2-dimensional minutiae histograms capture the minutiae distribution as a fixed-length feature vector which is invariant to rotation, translation and the variations in the number of minutiae. Scale invariance can be achieved by scaling input fingerprint images or minutiae templates to the size of adult fingerprints at a fixed resolution, e.g. 500 DPI. Fingerprints of adolescents can be enlarged using an age-dependent scaling factor as described in [27].

Performance Evaluation of Multi-Object Filters

In [28] and [29] the transport idea was used to evaluate the performance of multi-object filtering and control algorithms. Using a simulated ground truth of a varying number of objects moving through space, the online predictions by an algorithm that had only a cluttered version of the ground truth available was judged by performance curves over time. These curves at any one time were defined as the cost of the optimal transport between predicted configuration and ground truth.

Perceived Plant Color

The EMD was applied for computing color differences between images of different plant species by Kendal *et al.* [30]. Comparisons showed that these results were largely consistent with qualitative assessments by human experts.

Shape Matching

A fast approximation of the EMD for shape matching was introduced by Grauman and Darrell in [31]. Similar shapes are retrieved by embedding the minimum weight matching of the contour features of a query contour and performing an approximate nearest neighbors search with locality-sensitive

hashing. Ling and Okada proposed a method [32] that reduces the computational complexity for computing the EMD between histograms and they show its usefulness for shape matching and histogram feature matching. However, the method is restricted to the taxicab metric (ℓ_1 distance).

Cell Classification

Qiu [33] considered the two-class problem of classifying cells represented by multi-dimensional flow cytometry data into cells from healthy donors and cells from patients with acute myeloid leukemia. The EMD was used by Qiu to compare cell distributions and to derive features for classification.

Complex Scene Analysis

Ricci *et al.* apply the EMD idea for analyzing complex scenes such as frames from videos which change dynamically and they propose to learn a sparse set of prototypes with EMD [2].

Visual Object Tracking

Zhao *et al.* address the problem of visual object tracking [34]. They argue that the EMD is suited for capturing the perceptual differences between images, however, its computational complexity is too large for many potential applications. They propose a differential EMD for tracking which has a reduced computational complexity.

Squared Euclidean Distances and the Interpolation of Shapes and Images

In the last two decades, numerical schemes were proposed for the special situation that the ground distance is the square of the Euclidean distance between origins and destinations. Aurenhammer *et al.* [35] proposed an algorithm which uses power diagrams to transform the transportation problem into an unconstrained convex minimization problem. Recently, Mériçot [36] improved this algorithm by solving this optimization problem via a multiscale approach and applied it to the interpolation of images. Further methods for solving transportation problems with a squared Euclidean ground distance were proposed by Benamou and Brenier [37], by Angenent *et al.* [38], by Loeper and Rapetti [39] and by Benamou *et al.* [40].

Assignment Problems

An important special case of the transportation problem is the assignment problem, where the numbers of origins and destinations are the same and the mass at each origin and destination is equal to one.

References

- Rubner Y, Tomasi C, Guibas IJ (2000) The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision* 40: 99–121.
- Ricci E, Zen G, Sebe N, Messelodi S (2013) A prototype learning framework using EMD: Application to complex scenes analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35: 513–526.
- Gottschlich C, Huckemann S (2015) Separating the real from the synthetic: Minutiae histograms as fingerprints of fingerprints. *IET Biometrics*. In press. <http://dx.doi.org/10.1049/iet-bmt.2013.0065>
- Monge G (1781) *Mémoire sur la théorie des déblais et des remblais*. De l'Imprimerie Royale.
- Dantzig GB (1963) *Linear Programming and Extensions*. Princeton, NJ: Princeton Univ. Press.
- Dantzig GB (1990) The diet problem. *Interfaces* 20: 43–47.
- Villani C (2008) *Optimal Transport, Old and New*. Berlin, Germany: Springer.
- Luenberger DG, Ye Y (2008) *Linear and Nonlinear Programming*. New York, NY: Springer.
- Srinivasan V, Thompson GL (1973) Benefit-cost analysis of coding techniques for the primal transportation algorithm. *Journal of the ACM* 20: 194–213.
- Glover F, Karney D, Klingman D, Napier A (1974) A computation study on start procedures, basis change criteria, and solution algorithms for transportation problems. *Management Science* 20: 793–813.
- Sedgewick R (2003) *Algorithms in Java*. Boston, MA: Addison-Wesley.
- Houthakker HS (1955) On the numerical solution of the transportation problem. *Operations Research* 3: 210–214.
- Reinfeld NV, Vogel WR (1958) *Mathematical Programming*. Englewood Cliffs, NJ: Prentice-Hall.
- Bertsekas DP, Castanon DA (1989) The auction algorithm for the transportation problem. *Annals of Operations Research* 20: 67–96.
- Russell EJ (1969) Extension of Dantzig's algorithm to finding an initial near-optimal basis for the transportation problem. *Operations Research* 17: 187–191.
- Habr J (1961) Die Frequenzmethode zur Lösung der Transportprobleme und verwandter linearer Programmierungsprobleme. *Wissenschaftliche Zeitung der Universität Dresden* 10: 1069–1071.
- Urbanek S, Rubner Y (2012). *emdlist*. R package version 0.3-1. Available: <http://CRAN.R-project.org/package=emdlist>.
- Berkelaar M (2014). *lp_solve* v5.5. Available: <http://lpsolve.sourceforge.net/5.5/>.

There exists a multitude of applications in computer science and electrical engineering as well as in operations research: e.g. assigning n persons to n jobs, or n computational tasks to n nodes in a network.

For geographical coordinates obtained at different points in time for objects like airplanes from radar or satellites, target tracking can be viewed as an assignment problem by matching moving targets observed at two points in time. However, if more than two points in time are considered simultaneously, the problem becomes a multi index assignment problem which is a NP-hard problem [41].

Further potential applications can arise in the area of future public transportation systems: in case of a prevalence of electric drive vehicles and autonomous driving, the proposed method can be used to optimally assign cars to recharging locations, using for recharging e.g. a wireless transmission by electromagnetic induction.

Conclusions

In this paper, we have introduced the Shortlist Method, which is a novel approach for solving the classical transportation problem in full generality (with an arbitrary cost matrix) based on the simplex algorithm. We have demonstrated that the new method clearly outperforms previous variants of the simplex algorithm and two freely available modern solvers of transportation problems on a rather general benchmark. In view of the host of specialized transportation problems, we are far from making a claim that the Shortlist Method is universally the best in any way. However, we do believe that it is an appealing addition to the zoo of transportation algorithms that is very versatile and whose full potential has yet to be uncovered.

There are various other promising approaches to fast solving of large-scale transportation problems, many for special cost matrices (e.g. based on *squared* Euclidean distance, as mentioned in Section) or considering only rather coarse approximations to the real problem. Also there are many modern ideas to optimization, such as the growing class of swarm intelligence algorithms. These algorithms imitate aspects of the behavioral patterns of social animals, such as ants (see e.g. [42]) or bees, and have shown remarkable performance for similar problems in combinatorial optimization (see e.g. [43,44]).

Author Contributions

Conceived and designed the experiments: CG DS. Performed the experiments: CG DS. Analyzed the data: CG DS. Wrote the paper: CG DS.

19. Faraway JJ (2004) Linear Models with R. Boca Raton, FL: Chapman & Hall/CRC.
20. Gottschlich C (2012) Curved-region-based ridge frequency estimation and curved Gabor filters for fingerprint image enhancement. *IEEE Transactions on Image Processing* 21: 2220–2227.
21. Gottschlich C, Marasco E, Yang AY, Cukic B (2014) Fingerprint liveness detection based on histograms of invariant gradients. In: *Proc. IJCB*. Clearwater, FL, USA.
22. Fu A, Wenyin L, Deng X (2006) Detecting phishing web pages with visual similarity assessment based on earth mover's distance (EMD). *IEEE Transactions on Dependable and Secure Computing* 3: 301–311.
23. Vejdemo-Johansson M, Vejdemo S, Ek CH (2014) Comparing distributions of color words: Pitfalls and metric choices. *PLOS ONE* 9: e89184.
24. (2010) CPLEX v12.5.1. High-performance mathematical programming engine. International Business Machines corp.
25. Pele O, Werman M (2009) Fast and robust earth mover's distances. In: *Proc. ICCV*. Kyoto, Japan, pp. 460–467.
26. Lv Q, Charikar M, Li K (2004) Image similarity search with compact data structures. In: *Proc. CIKM*. Washington, DC, USA, pp. 208–217.
27. Gottschlich C, Hotz T, Lorenz R, Bernhardt S, Hantschel M, et al. (2011) Modeling the growth of fingerprints improves matching for adolescents. *IEEE Transactions on Information Forensics and Security* 6: 1165–1169.
28. Hoffman J, Mahler R (2004) Multitarget miss distance via optimal assignment. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 34: 327–336.
29. Schuhmacher D, Vo BT, Vo BN (2008) A consistent metric for performance evaluation of multi-object filters. *IEEE Transactions on Signal Processing* 56: 3447–3457.
30. Kendal D, Hauser C, Garrard G, Jelinek S, Giljohann K, et al. (2013) Quantifying plant colour and colour difference as perceived by humans using digital images. *PLOS ONE* 8: e72296.
31. Grauman K, Darrell T (2004) Fast contour matching using approximate earth mover's distance. In: *Proc. CVPR*. Washington, DC, USA, pp. 220–227.
32. Ling H, Okada K (2007) An efficient earth mover's distance algorithm for robust histogram comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29: 840–853.
33. Qiu P (2013) Inferring phenotypic properties from single-cell characteristics. *PLOS ONE* 7: e37038.
34. Zhao Q, Yang Z, Tao H (2010) Differential earth mover's distance with its applications to visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32: 274–287.
35. Aurenhammer F, Hoffmann F, Aronov B (1998) Minkowski-type theorems and least-squares clustering. *Algorithmica* 20: 61–76.
36. Mérigot Q (2011) A multiscale approach to optimal transport. *Computer Graphics Forum* 30: 1583–1592.
37. Benamou JD, Brenier Y (2000) A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numerische Mathematik* 84: 375–393.
38. Angenent S, Haker S, Tannenbaum A (2003) Minimizing flows for the Monge-Kantorovich problem. *SIAM Journal on Mathematical Analysis* 35: 61–97.
39. Loeper G, Rapetti F (2005) Numerical solution of the Monge-Ampère equation by a Newton's algorithm. *Comptes Rendus Mathématique* 340: 319–324.
40. Benamou JD, Froese B, Oberman A (2014) Numerical solution of the optimal transportation problem using the Monge-Ampère equation. *Journal of Computational Physics* 260: 107–126.
41. Spieksma F, Woeginger G (1996) Geometric three-dimensional assignment problems. *European Journal of Operational Research* 91: 611–618.
42. Li L, Peng H, Kurths J, Yang Y, Schellnhuber HJ (2014) Chaos-order transition in foraging behavior of ants. *PNAS* 111: 8392–8397.
43. Bonabeau E, Dorigo M, Theraulaz G (2000) Inspiration for optimization from social insect behaviour. *Nature* 406: 39–42.
44. Lučić P, Teodorović D (2002) Transportation modeling: an artificial life approach. In: *Proc. IC'AI*. Washington, DC, USA, pp. 216–223.