A linear time algorithm for the bottleneck transportation problem with a fixed number of sources

DORIT S. HOCHBAUM *   GERHARD J. WOEGINGER †

Abstract

We investigate a special case of the bottleneck transportation problem where the number s of sources is bounded by a constant and not part of the input. For the subcase $s = 2$, a best possible linear time algorithm has been derived by Varadarajan [Operations Research Letters 10 (1991), 525–529].

In this note we show that for any fixed number $s \geq 1$, the bottleneck transportation problem with s sources can be solved in linear time. The algorithm is conceptually simple and easy to implement.

Keywords. Transportation problem, bottleneck cost, linear time algorithm.

1 Introduction

This note discusses the $s \times n$ *bottleneck transportation problem* (BTP) that can be formulated as follows:

$$
\begin{array}{llll}
\text{(BTP)} & \min & \max_{\{(i,j):\ x_{ij}>0\}} \ c_{ij} & \\
& \text{s.t.} & \sum_{j=1}^{n} x_{ij} \ = \ a_i & \text{for } 1 \leq i \leq s \\
& & \sum_{i=1}^{s} x_{ij} \ = \ b_j & \text{for } 1 \leq j \leq n \\
& & x_{ij} \geq 0 & \text{for } 1 \leq i \leq s,\ 1 \leq j \leq n.
\end{array}
$$

Here c_{ij} is the cost associated with the connection between *source* i and *destination* j, x_{ij} is the quantity shipped from i to j, a_i is the *supply* at source i, and b_j is the *demand* at destination j. All numbers c_{ij}, a_i, and b_j are non-negative integers; it is customarily assumed that the total supply equals the total demand, $\sum_{i=1}^{s} a_i = \sum_{j=1}^{n} b_j$. It is well-known that under these conditions, the transportation problem BTP always possesses an optimal solution in which all the numbers x_{ij} are integer.

*`dorit@hochbaum.berkeley.edu`. Department of IEOR, University of California, Berkeley, CA 94720, USA.
Supported in part by NSF award No. DMI-9713482, and by SUN Microsystems.

†`gwoegi@opt.math.tu-graz.ac.at`. Institut für Mathematik B, TU Graz, Steyrergasse 30, A-8010 Graz, Austria. Supported by the START program Y43-MAT of the Austrian Ministry of Science.

The *linear transportation problem* (LTP) has the same constraint system as the BTP, but a different objective function:

$$\text{(LTP)} \qquad \min \sum_{i=1}^{s} \sum_{j=1}^{n} c_{ij} x_{ij}$$

We refer the reader e.g. to Nemhauser & Wolsey [13] for more information on the LTP, and to Garfinkel & Rao [8] and Derigs & Zimmermann [4] for more information on the BTP and for some of its applications.

There are several papers that investigate the special cases of LTP and BTP where the number of sources is bounded by a constant. Adolphson & Thomas [1] showed that that the LTP with $s = 2$ sources is solvable in linear time $O(n)$. Hochbaum [10] and Matsui [11] presented algorithms with $O(n)$ running time for the LTP for any fixed number s of sources. However, the LTP with a fixed number of sources is just a special case of the d-dimensional linear multiple choice knapsack problem, for which linear time algorithms based on the basic multi-dimensional search of Megiddo [12] were constructed by Dyer [5] and by Zemel [17]. The approach of [5, 17] yields the optimal objective value for the dual of LTP in $O(2^{2^s} n)$ time. The bipartite flow algorithm of Gusfield, Martel & Fernandez-Baca [9] then can be applied to explicitly compute the corresponding primal solution in $O(s^2 n)$ time. All in all, this results in an $O(n)$ algorithm for the LTP with a fixed number of sources.

Let us turn to the BTP. For the special case of the BTP with $s = 2$ sources, Varadarajan [16] gave an optimal $O(n)$ time solution algorithm, and Swarc [14] described a very simple $O(n \log n)$ time algorithm.

It can be observed that any instance of BTP can be formulated as an instance of LTP with the same number of sources: Consider an arbitrary instance I^B of BTP, and denote by $A = \sum_{i=1}^{s} a_i$ the total supply and demand. Construct a corresponding instance I^L of LTP, where every cost coefficient c_{ij} in BTP is replaced by the new cost coefficient $c_{ij}^* := A^{c_{ij}}$. If $c_{ij} > c_{k\ell}$ holds in instance I^B, then in instance I^L it is cheaper to transport *all* units via the connections with cost bounded by $c_{k\ell}^*$, than to send a *single* unit via a connection of cost c_{ij}^*. Hence, any optimal solution to I^L induces an optimal solution to I^B.

An obvious drawback of this formulation is that the computer word length becomes very large: Let $\overline{c} := \max_{i,j} c_{ij}$ denote the largest cost coefficient. Then in instance I^B, the length of the binary encoding of every cost coefficient is in $O(\log \overline{c})$. In instance I^L, however, the binary encodings of the cost coefficients may be as large as $\Theta(\overline{c} \log A)$ which is exponentially larger than the length of I^B. Such a huge word length is not compatible with the unit cost model of computation, which is our standard model for dealing with arithmetic operations (cf. e.g. Aho, Hopcroft & Ullman [2]). One way of getting around this word length would be to replace the cost coefficients in I^B by the smaller numbers $1, 2, \ldots, sn$ without changing their relative ordering. This would be feasible, since in a bottleneck problem the actual size of the cost coefficients does not matter, but only their ordering relative to each other does. However, replacing the coefficients by the smaller number would amount to sorting them, and we are not aware of *any* reasonable model of computation in which this could be done in $O(n)$ time.

Contribution of this note. We present an algorithm with $O(n)$ running time for the BTP for the case where the number s of sources is constant and not part of the input. Our algorithm is based on an 'inefficient' formulation of the BTP that involves $O(2^s)$ constraints. Combining this formulation with the linear-time median finding technique of Blum, Floyd, Pratt, Rivest & Tarjan [3] yields a fast $O(2^s n) = O(n)$ time algorithm for the BTP with a constant number of sources. The computer word length in this algorithm is logarithmically bounded in the length of the input, i.e., it is the usual word length in the usual unit cost model of computation (cf. [2]).

Organization of the note. Section 2 provides some technical tools that are then applied in Section 3 to derive a linear time algorithm for the BTP with a bounded number s of sources.

2 Technical Preliminaries

This section presents two technical lemmas that are to be used in Section 3.

Lemma 1 *Let there be given n pairs (q_j, r_j), $1 \leq j \leq n$, of non-negative integers and a bound R. Then the smallest number Q that fulfills*

$$\sum_{j: q_j > Q} r_j \ \leq \ R$$

can be computed in $O(n)$ time.

Proof. Observe that if $\sum_{j=1}^{n} r_j \leq R$ holds, the smallest possible Q equals $-\infty$. Otherwise if $\sum_{j=1}^{n} r_j > R$, the smallest possible Q must be one of the numbers $q_1, \ldots, q_n$. The algorithm starts with determining the median q^* of $q_1, \ldots, q_n$. Let $R^>$ denote the sum of all numbers r_j that fulfill $q_j > q^*$ and let $R^=$ denote the sum of all numbers r_j with $q_j = q^*$ (empty sums have value zero by definition). We distinguish three cases. (Case 1) If $R^> \leq R < R^> + R^=$, then we set $Q = q^*$ and stop. (Case 2) If $R < R^>$, then we remove all pairs (q_j, r_j) with $q_j \leq q^*$, and recursively solve the problem for the at most $\lfloor \frac{n}{2} \rfloor$ remaining pairs with the same bound R. (Case 3) In case $R \geq R^> + R^=$ holds, we remove all pairs (q_j, r_j) with $q^* \leq q_j$, and recursively solve the problem for the at most $\lfloor \frac{n}{2} \rfloor$ remaining pairs, but this time with the new bound $R - (R^> + R^=)$.

The procedure terminates when the condition in (Case 1) is fulfilled. The correctness of this procedure is straightforward to verify. For the time complexity, we recall a well-known result of Blum, Floyd, Pratt, Rivest & Tarjan [3]: The median of n numbers can be determined in $O(n)$ time. With this, the time complexity $T(n)$ of the above recursive procedure fulfills $T(n) \ \leq \ T(\lfloor \frac{n}{2} \rfloor) + O(n)$. This clearly yields a linear running time $T(n) = O(n)$. ∎

Next, we want to discuss transportation problems with forbidden connections: Similarly as above, there are sources $1, \ldots, s$, destinations $1, \ldots, n$, supplies a_i at the sources and demands b_j at the destinations. However, now some of the source-destination connections are forbidden, i.e., not every destination is reachable from every source: Every destination j is labeled by a subset $\ell(j) \subseteq \{1, \ldots, s\}$, and shipment from source i to destination j is possible if and only if $i \in \ell(j)$ holds.

Lemma 2 *A transportation problem with forbidden connections has a feasible solution if and only if $\sum_{i \in S} a_i \ \geq \ \sum_{j: \ell(j) \subseteq S} b_j$ holds for all $S \subseteq \{1, \ldots, s\}$.* ∎

The statement in Lemma 2 is folklore: It is e.g. contained in Tutte's characterization [15] of so-called f-soluble graphs. Fujishige [6] attributes it to Gale [7]. Its straightforward proof by induction is omitted.

3 The linear time algorithm

In this section, we derive a simple $O(n)$ time solution algorithm for the bottleneck transportation problem with a bounded number s of sources. The algorithm is based on the linear time algorithm described in Lemma 1 and on the combinatorial result in Lemma 2.

For an arbitrary non-negative integer c and for an arbitrary destination j in the BTP, we denote by $\ell(c,j)$ the set of all sources i for which $c_{ij} \le c$ holds. In other words, $\ell(c,j)$ contains exactly those sources that may support destination j while keeping the bottleneck cost smaller or equal to c. Then according to Lemma 2, the BTP may be reformulated as

$$\begin{array}{lllll} \min & c & & & \\ \text{s.t.} & \sum_{i\in S} a_i & \ge & \sum_{j:\ell(j,c)\subseteq S} b_j & \text{for all } S \subseteq \{1,\ldots,s\} \end{array} \tag{1}$$

Note that for $S = \{1,\ldots,s\}$, the constraint in (1) is fulfilled a priori, since $\sum_{i=1}^{s} a_i = \sum_{j=1}^{n} b_j$ holds. Next, consider some fixed set $S \subset \{1,\ldots,s\}$. For every destination j, let

$$c(S,j) = \min\ \{c_{ij} :\ i \notin S\}.$$

In other words, $c(S,j)$ is the smallest bottleneck cost for which the destination j may be supported by some source outside of set S. Then another equivalent formulation of BTP is

$$\begin{array}{lllll} \min & c & & & \\ \text{s.t.} & \sum_{i\in S} a_i & \ge & \sum_{j:c(S,j)>c} b_j & \text{for all } S \subset \{1,\ldots,s\}. \end{array} \tag{2}$$

The constraints in (2) are exactly of the form that has been treated in Lemma 1. This suggests the following approach. For every set $S \subset \{1,\ldots,s\}$, proceed as follows: First determine all the numbers $c(S,j)$ in $O(n)$ overall time. Then set $R = \sum_{i\in S} a_i$, set $r_j = b_j$ and $q_j = c(S,j)$ for $1 \le j \le n$ and apply the linear time algorithm in Lemma 1. The output $c^*(S)$ of this algorithm is a lower bound on the optimal bottleneck cost c^{opt}, i.e., $c^{opt} \ge c^*(S)$ must hold for every set S.

In the end the solution value c^{opt} of the BTP is set to the maximum of the values $c^*(S)$, where $S \subset \{1,\ldots,s\}$. All in all, there are $O(2^s)$ sets to handle, and every set is handled in $O(n)$ time. Since s is considered constant in this paper, this yields an overall running time of $O(n)$.

Summarizing, we formulate the following theorem.

Theorem 3 *The $s \times n$ bottleneck transportation problem with a bounded number s of sources can be solved to optimality in $O(n)$ time.* ■

Finally, we remind the reader that once the preceding algorithm has determined the optimal flow value of the BTP in linear time, the algorithm of Gusfield, Martel & Fernandez-Baca [9] can then determine an optimal flow also in linear time.

Acknowledgement. The comments of an unknown referee helped to improve the presentation.

References

[1] D.L. ADOLPHSON AND G.N. THOMAS, A linear time algorithm for a $2 \times n$ transportation problem, *SIAM Journal on Computing* **6**, 1977, 481–486.

[2] A.V. AHO, J.E. HOPCROFT, AND J.D. ULLMAN, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.

[3] M. BLUM, R.W. FLOYD, V. PRATT, R.L. RIVEST, AND R.E. TARJAN, Time bounds for selection, *Journal of Computer and System Sciences* **7**, 1973, 448–461.

[4] U. DERIGS AND U. ZIMMERMANN, An augmenting path method for solving linear bottleneck transportation problems, *Computing* **22**, 1979, 1–15.

[5] M.E. DYER, An $O(n)$ algorithm for the multiple-choice knapsack linear program, *Mathematical Programming* **29**, 1984, 57–63.

[6] S. FUJISHIGE, *Submodular Functions and Optimization*, Annals of Discrete Mathematics, Vol. 47, North-Holland, Amsterdam, 1991.

[7] D. GALE, A theorem of flows in networks, *Pacific Journal of Mathematics* **7**, 1957, 1073–1082.

[8] R.S. GARFINKEL AND M.R. RAO, The bottleneck transportation problem, *Naval Research Logistics Quarterly* **18**, 1971, 465–472.

[9] D. GUSFIELD, C. MARTEL, AND D. FERNANDEZ-BACA, Fast algorithms for bipartite network flow, *SIAM Journal on Computing* **16**, 1987, 237–251.

[10] D.S. HOCHBAUM, A faster strongly polynomial algorithm for the transportation problem with few sources, talk at the *Symposium on Mathematical Programming*, August 1991, Amsterdam.

[11] T. MATSUI, A linear time algorithm for the Hitchcock transportation problem with a fixed number of supply points, talk at the *Symposium on Mathematical Programming*, August 1991, Amsterdam.

[12] N. MEGIDDO, Linear programming in linear time when the dimension is fixed, *Journal of the ACM* **31**, 1984, 114–127.

[13] G.L. NEMHAUSER AND L.A. WOLSEY, *Integer and Combinatorial Optimization*, John Wiley & Sons, New York, 1988.

[14] W. SWARC, An instant solution of the $2 \times n$ bottleneck transportation problem, *Operations Research Letters* **14**, 1993, 261–264.

[15] W.T. TUTTE, A short proof of the factor theorem for finite graphs, *Canadian Journal of Mathematics* **6**, 1954, 347–352.

[16] R. VARADARAJAN, An optimal algorithm for $2 \times n$ bottleneck transportation problems, *Operations Research Letters* **10**, 1991, 525–529.

[17] E. ZEMEL, An $O(n)$ algorithm for the linear multiple choice knapsack problem and related problems, *Information Processing Letters* **18**, 1984, 123–128.