4OR-Q J Oper Res (2009) 7:51–60
DOI 10.1007/s10288-007-0067-z

RESEARCH PAPER

The transportation problem with exclusionary side constraints

Dries Goossens · Frits C. R. Spieksma

Received: 15 December 2005 / Revised: 5 December 2006 / Published online: 12 January 2008

Abstract We consider the so-called Transportation Problem with Exclusionary Side Constraints (TPESC), which is a generalization of the ordinary transportation problem. We confirm that the TPESC is *NP*-hard, and we analyze the complexity of different special cases. For instance, we show that in case of a bounded number of suppliers, a pseudo-polynomial time algorithm exists, whereas the case of two demand nodes is already hard to approximate within a constant factor (unless $P = NP$).

Keywords Transportation problem · Exclusionary side constraints · Complexity

MSC classification (2000) 68Q25

1 Introduction

The ordinary transportation problem is well-known: given a number of supply nodes each with a certain supply of items, a number of demand nodes each with a certain demand for items, and a unit transportation cost for each pair consisting of a supply node and a demand node, send the items from the supply nodes to the demand nodes at a minimum cost.

In this note we consider the variant where for each demand node a set of pairs of supply nodes is given such that at most one supply node of each given pair is allowed

This research was partially supported by FWO Grant No. G.0114.03.

D. Goossens (✉) · F. C. R. Spieksma
Department of Operations Research and Business Statistics, Katholieke Universiteit Leuven,
Naamsestraat 69, 3000 Leuven, Belgium
e-mail: Dries.Goossens@econ.kuleuven.be

to send items to that demand node. Following the literature, we refer to this problem as the transportation problem with exclusionary side constraints (TPESC).

As far as we are aware, this problem has first been introduced by Cao (1992), who described an application in storage management of containers. In this application, arriving containers must be positioned in rows of a storage yard, such that the costs of operations (searching, loading, retrieving) are minimized. Differences in size, ownership, or content may disallow containers to be stored in the same row, giving rise to exclusionary side constraints. A branch-and-bound approach was described to solve the problem. Other branch-and-bound approaches are described and tested in Sun (2002), while evolutionary algorithms have been proposed and tested by Cao and Uebe (1995), and Syarif and Gen (2003). These contributions suggest that the problem is *NP*-hard, although no formal statement of this result seems to have been made. We confirm this suggestion by providing an explicit proof, and we study the computational complexity of various special cases of the problem.

Our interest in this generalization of the transportation problem stems from an application that occurs in the context of a procurement problem (see Sect. 2 ; Goossens et al. (2007) for a description).

1.1 Problem statement

TPESC can be formulated as follows. Let there be a set S of *supply* nodes, each with a supply of s_i, $i \in S$, and a set D of *demand* nodes, each with a demand of d_j, $j \in D$. For each pair consisting of supply node $i \in S$ and demand node $j \in D$, a unit cost $c_{ij} \geq 0$ is given. Finally, for each demand node $j \in D$, a (possibly empty) set of pairs of supply nodes, called F_j, is given; thus $F_j = \{(i_1, i_2) | (i_1, i_2) \in S \times S, i_1 \neq i_2\}$. We assume that all data are integral. The problem is to send all supply to the demand nodes at minimum cost, such that each demand node $j \in D$ receives items from at most one supply node for each pair of supply nodes present in F_j. Obviously, if $F_j = \emptyset$ for all $j \in D$, the ordinary transportation problem arises. Notice that we assume that total supply equals total demand, that is $\sum_{i \in S} s_i = \sum_{j \in D} d_j$ (since otherwise no feasible solution exists). For a mathematical formulation, we refer to Sun (2002). When we use the phrase "the feasibility version of TPESC", we refer to the situation where the (given) bipartite network between supply nodes and demand nodes is not necessarily complete, and no costs are specified. Therefore, the question to answer is whether a feasible solution (using only edges from the network) exists.

1.2 Special cases

In this paper, we study three relevant special cases of the transportation problem with exclusionary side constraints: TPESC with identical exclusionary sets (Sect. 2), TPESC with a single exclusionary set (Sect. 3), and TPESC with a fixed number of supply nodes (Sect. 4).

The transportation problem with identical exclusionary sets arises in a procurement problem (see Goossens et al. 2007). Consider a buyer procuring given amounts of different goods from different suppliers. Each of the suppliers uses a so-called *total* *quantity discount* policy to set the prices for the different goods; more in particular, each supplier distinguishes volume intervals on the total number of sold items that determine the prices charged for each individual good. The resulting procurement problem (referred to as the TQD problem) is to obtain the given amounts of each of the different goods from the suppliers at minimum cost. Thus, a solution for an instance of this TQD problem prescribes how much items of each good are ordered from each supplier. In Chauhan et al. (2005), a PTAS is described for a special case of the problem involving a single good. Observe that the TQD problem (as the TPESC) is a generalization of the ordinary transportation problem. Indeed, by associating a demand node with each good (with its demand equal to the amount that needs to be bought), and by associating a supply node with appropriate lower and upper bounds with each volume interval of each supplier, the TQD problem boils down to selecting supply nodes (at most one from each supplier) and by finding the right amount of items of each good to be transported. (In case a supplier can only deliver a fixed number of items, that is, there is only one supply node for each supplier with coinciding upper and lower bound, the ordinary transportation problem arises). One important aspect in this generalization of the transportation problem is the fact that for each demand node, a set of supply nodes is given (namely the nodes corresponding to the intervals of a single supplier) from which at most one can be used to actually supply that demand node; this corresponds to our F_j, $j \in D$ sets. Observe that if a supplier uses more than two intervals, this is easily accommodated by having an element in the exclusionary set for each pair of volume intervals (which gives rise to a polynomial number of elements in the exclusionary set). Also, observe that these sets are the same for all demand nodes, in other words, we are dealing with an instance of TPESC with identical F-sets.

In Sect. 3, we deal with the problem that arises when exactly one F-set is nonempty. The following practical application illustrates the relevance of this special case of TPESC. When a company decides to store its goods, it basically has the choice between constructing its own private warehouse and renting a public warehouse. Assuming that there are seasonal changes in the need for storage space, Ballou (1998) shows that it is advisable to make use of both options. This leaves the company with the problem of where to store what goods, minimizing the total cost. One can imagine that the public warehouse imposes constraints on what goods can be stored together (e.g. hazardous materials), whereas these constraints could be non-existing in a private warehouse, since this warehouse can be built specifically according to the (safety) needs of the company. This practical application boils down to a TPESC with only two demand nodes, where only one has a nonempty F-set (namely the demand node corresponding to the public warehouse).

Finally, in Sect. 4, we consider a setting of TPESC where the number of supply nodes is fixed.

1.3 Our results

In this paper, we show that the feasibility version of TPESC is *NP*-complete. In fact, the problem is *NP*-complete already for the smallest nontrivial case, that is, the case with two demand nodes. Even more specifically, we establish for each of the three

Table 1 The complexity status of TPESC

TPESC with identical exclusionary sets		TPESC with a single exclusionary set	TPESC with a fixed number of supply nodes
$\|D\| = 2$	$\|D\| \geq 3$	$\|D\| \geq 2$	$\|S\| \geq 2$
Weakly *NP*-hard; Pseudo-pol. time algorithm	Strongly *NP*-hard	No polynomial-time constant-factor approximation (unless $P = NP$)	Weakly *NP*-hard; Pseudo-pol. time algorithm

special cases of TPESC its complexity status (we refer to Garey and Johnson (1979) or Ausiello et al. (1999) for an introduction to these issues). For the case with identical exclusionary sets, we show that in case of two demand nodes the feasibility version of TPESC is (weakly) *NP*-complete and a pseudo-polynomial time algorithm exists. In case of three demand nodes, this problem becomes strongly *NP*-complete (see Sect. 2). For the setting with a single exclusionary set, we show that the existence of a polynomial-time algorithm with a fixed performance ratio would imply $P = NP$, even in the case of two demand nodes (see Sect. 3). Finally, in Sect. 4 we present a pseudo-polynomial time algorithm for the case of a fixed number of supply nodes, and we show that the feasibility version of TPESC with two supply nodes is already (weakly) *NP*-complete. Table 1 gives an overview of our results. Our results explain the use of heuristics and branch-and-bound approaches (Cao 1992; Sun 2002; Cao and Uebe 1995; Syarif and Gen 2003) for solving large instances of the TPESC.

2 TPESC with identical exclusionary sets

In this section we focus on the TPESC with identical exclusionary sets. We first prove that the problem with $|D| = 2$, that is, the case of two demand nodes, is weakly *NP*-complete, then we exhibit a pseudo-polynomial time algorithm for this case, and finally we show that the problem with $|D| = 3$ is strongly *NP*-complete.

2.1 The case $|D| = 2$

The following theorem shows that the feasibility version of TPESC with identical exclusionary sets is *NP*-complete already for the smallest nontrivial case.

Theorem 1 *The feasibility version of TPESC with identical exclusionary sets is NP-complete, even if* $|D| = 2$.

Proof We prove the theorem by presenting a reduction from Even-Odd Partitioning (EOP) to TPESC. EOP is proved to be *NP*-complete in Garey et al. (1988).
EOP Input: n pairs of positive integers (x_{2i-1}, x_{2i}), $i = 1, \ldots, n$.
Question: does there exist a partition of $\{1, \ldots, 2n\}$ into disjoint subsets A and B with $|A \cap \{2i-1, 2i\}| = |B \cap \{2i-1, 2i\}| = 1$ for $i = 1, \ldots, n$, and with $\sum_{i \in A} x_i = \sum_{i \in B} x_i$?

For each integer in the input of EOP, we construct a supply node with supply equal to the value of the integer, that is, we set $S = \{1, 2, \ldots, 2n\}$ with $s_i = x_i$ for $i = 1, \ldots, 2n$. There are two demand nodes, each having demand $d_1 = d_2 = \frac{1}{2}\sum_{i=1}^{2n} x_i$. We set $F_1 = F_2 = \{(x_{2i-1}, x_{2i}) | \, i = 1, \ldots, n\}$, implying that at most one supply node per pair is allowed to send items to that demand node. Each supply node is connected to each demand node. This completes the description of the instance of TPESC.

A yes-answer to the EOP instance directly corresponds to a feasible solution of the TPESC instance. Also, by observing the fact that the two demand nodes have identical exclusionary constraints, it is clear that in any feasible solution of the TPESC instance, each supply node sends its entire supply to precisely one of the demand nodes which in turn corresponds to a yes-answer of the EOP instance. □

Of course, this result does not rule out the existence of a pseudo-polynomial time algorithm for TPESC with common exclusionary sets and two demand nodes. In the remainder of this section, we present such an algorithm. First we show how we can formulate TPESC with common exclusionary sets as a generalization of the change making problem. Then, we modify a dynamic program for the change making problem to solve this generalization.

We first construct a graph $G = (V, E)$. There is a node in G for each supply node in the TPESC instance. For each exclusionary constraint in F, let there be an edge between the pair of supply nodes involved in the exclusionary constraint. The resulting graph can be partitioned into a number of connected components (V_k, E_k), $k = 1, \ldots, c$, such that there is no exclusionary constraint between any two vertices in different sets V_k.

We now sketch a preprocessing phase in which we find out whether there is no contradiction caused by the exclusionary constraints. If, for instance, there is an exclusionary constraint for supply nodes 1 and 2, for supply nodes 1 and 3, and for supply nodes 2 and 3, it follows that not all supply can be sent, and hence no feasible solution exists. Thus, if the vertices of each component can be colored using two colors (say red and blue) such that vertices joined by an edge receive a different color, then there is a way to distribute the supply over the demand nodes without violating the exclusionary constraints. This is accomplished by sending the supply of the supply nodes with the same color to a single demand node. Further, observe that a feasible way of sending all supply to the demand nodes amounts to a 2-coloring of G. It follows that verifying 2-coloredness of G determines whether the exclusionary sets allow a feasible way of sending all supply from the supply nodes to the demand nodes.

From this phase, we can assume that each component has red nodes and blue nodes. Let us assume without loss of generality that the supply of the red nodes (say a_k^{red}) is at least as large as the supply of the blue nodes (a_k^{blue}), or $a_k^{red} \geq a_k^{blue}$. As observed earlier, the fact that the two demand nodes have identical exclusionary constraints, and that total supply equals total demand, implies that in any feasible solution, each supply node sends its entire supply to precisely one of the demand nodes. Thus, there are two ways of distributing the supply of each component: either the red nodes send their supply to demand node 1, and the blue nodes send their supply to demand node 2, or vice versa. The two corresponding costs are denoted by p_k^{red} and p_k^{blue}. We model these two possibilities with a binary variable x_k, which is 1 if the red nodes of component k supply demand node 1, and the blue nodes supply demand node 2, and 0 vice versa. Let us define $a_k = a_k^{red} - a_k^{blue}$, $p_k = p_k^{red} - p_k^{blue}$, and $B = d_1 - \sum_{k=1}^{c} a_k^{blue}$. We can now formulate TPESC with identical exclusionary sets and two demand nodes as follows:

$$\text{minimize} \quad \sum_{k=1}^{c} p_k x_k \tag{1}$$

$$\text{subject to} \quad \sum_{k=1}^{c} a_k x_k = B \tag{2}$$

$$x_k \in \{0, 1\} \quad \text{for } k = 1, \ldots, c. \tag{3}$$

Notice that the definitions above imply that $a_k \geq 0$. In fact, we can eliminate those variables x_k which have as coefficient $a_k = 0$ (since, in an optimal solution we set, in case $a_k = 0$: $x_k = 1$ if $p_k \leq 0$, else we set $x_k = 0$). Thus, henceforth we will assume that $a_k \geq 1$. Furthermore, we assume that $B \geq 0$, since no solution exists if $B < 0$.

This problem is a generalization of the change making problem (see Martello and Toth 1990), since there is a cost p_k associated to each variable x_k. Furthermore, there are bounds equal to 1 on the variables. Wright (1975) developed a dynamic program for the change-making problem. The following modified version of this algorithm, to which we refer as algorithm DP, provides an optimal solution for formulation (1)–(3).

Let $f_q(z)$ be the optimal solution value of a sub-instance of (1)–(3), consisting of components $1, \ldots, q$ and a right-hand side of z, with $1 \leq q \leq c$ and $0 \leq z \leq B$. If no solution exists for a combination of values q and z, then $f_q(z) = \infty$. It is clear that

$$f_1(z) = \begin{cases} 0 & \text{if } z = 0; \\ p_1 & \text{if } z = a_1; \\ \infty & \text{if } z \neq a_1. \end{cases}$$

Now, $f_q(z)$ can be computed by considering increasing values of q from 2 to c and, for each q, increasing values of z from 0 to B as

$$f_q(z) = \begin{cases} f_{q-1}(z) & \text{if } z = 0, 1, \ldots, a_q - 1; \\ \min(f_{q-1}(z), f_{q-1}(z - a_q) + p_q) & \text{if } z = a_q, \ldots, B. \end{cases}$$

The optimal solution value of formulation (1)–(3) is then given by $f_c(B)$. The time complexity of algorithm DP is $O(cB)$, which proves that TPESC with two demand nodes with identical exclusionary constraints can be solved in pseudo-polynomial time. We have shown the following:

Theorem 2 *Algorithm DP is a pseudo-polynomial time algorithm for TPESC with identical exclusionary constraints and two demand nodes.*

2.2 The case $|D| \geq 3$

We now argue that it is unlikely that algorithm DP can be extended to the case of three demand nodes by showing that TPESC with identical exclusionary sets and three demand nodes is strongly *NP*-hard.

Theorem 3 *The feasibility version of TPESC with identical exclusionary sets is strongly NP-complete, even if* $|D| = 3$.

Proof We prove the theorem by presenting a reduction from Graph 3-colorability (see Garey and Johnson 1979) to TPESC.
Graph 3-colorability Input: a graph $G = (V, E)$.
Question: is G 3-colorable, that is, does there exist a coloring of the vertices of G such that two vertices connected by an edge in E receive different colors, and such that no more than three different colors are used?

We build an instance of TPESC by having a supply node for every vertex of V, and a single dummy node d. Thus $S = V \cup \{d\}$. Each supply node corresponding to a vertex of G has $s_j = 1$, $j \in S \backslash \{d\}$, the supply corresponding to the dummy node equals $s_d = 2|V|$. There are three demand nodes, each having demand $d_j = |V|$. Let the two endpoints of an edge $e \in E$ be denoted by v_e and w_e. For each edge e in E there is a pair of supply nodes in F:

$$F = \{(v_e, w_e) |\ e \in E\}.$$

Further, each supply node is connected to each demand node. This completes the description of an instance of TPESC.

Suppose that G admits a 3-coloring. We associate a different color with each of the three demand nodes. Next, we send the unit supply of each supply node corresponding to a vertex $v \in V$ to the appropriate demand node (the one with v's color in the coloring). We use the supply from the dummy node to satisfy all demand from the demand nodes exactly. Observe that we have satisfied the exclusionary constraints, and hence we have a feasible solution to TPESC.

Suppose there is a feasible solution for TPESC. Consider the supply nodes v_e and w_e associated to edge e. Due to the choice for F, it follows that the supply of each of these supply nodes is sent to a different demand node. Thus, the supply of supply nodes that correspond to adjacent vertices in G, goes to different demand nodes. Since there are three demand nodes, we have found a 3-coloring. □

3 TPESC with a single exclusionary set

In this section we deal with the special case of TPESC where exactly one F-set is nonempty. We show that this special case is already hard to approximate, even for two demand nodes.

Theorem 4 *TPESC with a single exclusionary set does not admit a polynomial-time constant-factor approximation algorithm unless* $P = NP$*, even if* $|D| = 2$.

Proof We prove the theorem by presenting a reduction from Independent Set (IS) to TPESC.

Independent Set Input: a graph $G = (V, E)$ and an integer $K \leq |V|$.
Question: does there exist an independent set of cardinality at least K, that is, a subset $V' \subseteq V$ with $|V'| \geq K$, such that no two vertices in V' are joined by an edge in E?

For each vertex $j \in V$ we construct a supply node with supply $s_j = 1$; there is an additional supply node q with supply $s_q = K$. There are two demand nodes; the first one has demand $d_1 = K$, the second one has demand $d_2 = |V|$. The cost of the edge between supply node q and the first demand node equals $c > 0$, all other edges have cost 0. The first demand node has a set of exclusionary constraints $F_1 = \{(k, l)|\ k, l \in V \wedge (k, l) \in E\}$. The second demand node has no exclusionary constraints, that is, $F_2 = \emptyset$.

We now show that the existence of a polynomial-time algorithm with a constant performance ratio for TPESC would imply $P = NP$.

Suppose that the instance of IS has a yes-answer, that is, there exists an independent set V' of cardinality at least K. In this case, given the construction of F_1, there exist K supply nodes corresponding to nodes from the set V' that satisfy the exclusionary constraints. It is now easy to see that a solution where the supply of these nodes is sent to the first demand node and where the other nodes supply the second demand node, is a feasible solution to TPESC that has zero cost.

In case that the instance of TPESC admits a zero cost solution, apparently the edge between supply node q and the first demand node is not used. Hence, the demand of this node is fulfilled by K supply nodes that correspond vertices in G that form an independent set of size K.

Thus a polynomial-time algorithm with a constant performance ratio for TPESC would find a zero cost solution if one exists, and hence would be able to distinguish between the yes-instances and the no-instances of IS. □

4 TPESC with a fixed number of supply nodes

In this section we show that if the number of supply nodes is not part of the input, a pseudo-polynomial time algorithm exists to solve the problem. Observe that this contrasts with the case of a fixed number of demand nodes (in particular Theorem 4), where the case of two demand nodes renders a problem that does not allow a polynomial-time algorithm with a constant performance ratio (unless $P = NP$).

Theorem 5 *TPESC with a fixed number of supply nodes can be solved by a pseudo-polynomial time algorithm.*

Proof We prove the theorem by presenting a dynamic programming algorithm for TPESC with a fixed number of supply nodes. To facilitate the exposition, let $m = |S|$, $n = |D|$, and let L be the largest number in the input. As a state in the dynamic program, we use $(f_1, f_2, \ldots, f_m)$ where f_i denotes the amount of items sent by supply node i to all demand nodes. Observe that the number of states is bounded by $(L + 1)^m$. Further we define H_j, $1 \leq j \leq n$, as the set of states that can be reached after having fulfilled the demand of the demand nodes $1, 2, \ldots, j$, and we start with

$H_0 = (0, 0, \ldots, 0)$. In iteration j, we deal with demand node j that has demand d_j, $1 \leq j \leq n$. We enumerate all possible integral ways of distributing demand d_j over the m supply nodes. Notice that we use here the fact that if a solution exists, there exists one with integral flows. Let us define E_j as the set of m-vectors that correspond to a feasible way of distributing demand d_j over the m supply nodes. In the absence of exclusionary constraints, i.e., if $F_j = \emptyset$, then

$$|E_j| = \binom{d_j + m - 1}{m - 1}.$$

By enumerating all $\binom{d_j + m - 1}{m - 1}$ potential ways of distributing demand d_j over the m supply nodes, and next verifying, for each way, whether it is feasible with respect to the exclusionary constraints (whose number $|F_j|$ is bounded by $\binom{m}{2}$), we can find in $O(m^2(d_j + m)^m)$, the set E_j. Now, we can compute H_j as follows:

$$H_j = \{f + g | \, f \in H_{j-1}, g \in E_j\}.$$

States in which a value f_i exceeds s_i are omitted since they cannot lead to a feasible solution. Finally, we need to inspect whether $(s_1, s_2, \ldots, s_m) \in H_n$. If so, a solution is found, else no solution exists. The complexity of this algorithm is $O(n \cdot L^m \cdot m^2(L + m)^m)$, which, in case of a fixed m leads to a pseudo-polynomial time algorithm. Notice that when arbitrary costs c_{ij} are given, we can, by keeping track of the cost of an element of E_j, compute the cost of a state, thereby finding the cost of an optimal solution. □

It is not hard to see that Theorem 5 is best possible in the sense that one easily verifies that the existence of a polynomial-time algorithm for TPESC even with two supply nodes would imply $P = NP$. Indeed, the well-known Partition problem is easily seen to be a special case of the feasibility version of TPESC with two supply nodes.

Acknowledgements We thank Gerhard Woeginger for a discussion of the problem. We also wish to thank the referees of this paper for providing constructive comments.

References

Ausiello G, Crescenzi P, Gambosi G, Kann V, Marchetti-Spaccamela A, Protasi M (1999) Complexity and approximation. Springer, Heidelberg

Ballou RH (1998) Business logistics management. Prentice Hall, New York

Cao B (1992) Transportation problem with nonlinear side constraints a branch and bound approach. Math Methods Oper Res (ZOR) 36:185–197

Cao B, Uebe G (1995) Solving transportation problems with nonlinear side constraints with tabu search. Comp Oper Res 22:593–603

Chauhan SS, Eremeev AV, Romanova AA, Servakh VV, Woeginger GJ (2005) Approximation of the supply scheduling problem. Oper Res Lett 33:249–254

Garey MR, Johnson DS (1979) Computers and intractability: a guide to the theory of *NP*-completeness. W.H. Freeman, New York

Garey MR, Tarjan RE, Wilfong G (1988) One-processor scheduling with symmetric earliness and tardiness penalties. Math Oper Res 13:330–349

Goossens DR, Maas AJT, Spieksma FCR, van de Klundert JJ (2007) Exact algorithms for procurement problems under a total quantity discount structure. Eur J Oper Res 178:603–626

Martello S, Toth P (1990) Knapsack problems: algorithms and computer implementations. Wiley, New York

Sun M (2002) The transportation problem with exclusionary side constraints and two branch-and-bound algorithms. Eur J Oper Res 140:629–647

Syarif A, Gen M (2003) Solving exclusionary side constrained transportation problem by using a hybrid spanning tree-based genetic algorithm. J Intell Manuf 14:389–399

Wright JW (1975) The change-making problem. J Assoc Comput Mach 22:125–128