



The transportation problem with conflicts

Annette M. C. Ficker¹ · Frits C. R. Spieksma² · Gerhard J. Woeginger³

Published online: 21 August 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

The transportation problem is a fundamental problem in operations research, where items need to be transported from supply nodes (each with a given supply) to demand nodes (each with a given demand) in the cheapest possible way. Here, we are interested in a generalization of the transportation problem where, each supply node has a (possibly empty) set of conflicting pairs of demand nodes, and each demand node a (possibly empty) set of conflicting pairs of supply nodes. Each supply node may only send supply to at most one demand node of each conflicting pair. Likewise, each demand node may only receive supply from at most one supply node of each conflicting pair. We call the resulting problem the transportation problem with conflicts (TPC). We show that the complexity of TPC depends upon the structure of the so-called conflict graph that follows from the conflicting pairs. More concrete, we show that for many graph-classes the corresponding TPC remains NP-hard, and for some special cases we derive constant factor approximation algorithms.

Keywords Transportation problem · Conflict graph · Computational complexity · Approximation

1 Introduction

Consider the classical Transportation Problem, in which we are given suppliers, each having a supply, and locations, each having a demand. For all possible pairs consisting of a supplier and a location we are given a unit transportation cost. The goal is to fulfill the demand with minimum cost. This problem is well-known and efficiently solvable.

✉ Annette M. C. Ficker
annette.ficker@kuleuven.be

Frits C. R. Spieksma
f.c.r.spieksma@tue.nl

Gerhard J. Woeginger
woeginger@cs.rwth-aachen.de

¹ Faculty of Economics and Business, KU Leuven, 3000 Leuven, Belgium

² Department of Mathematics and Computer Science, Eindhoven University of Technology, 5600, MB Eindhoven, The Netherlands

³ Lehrstuhl für Informatik 1, RWTH Aachen, D-52056 Aachen, Germany

Many situations in practice have, as a base, this transportation problem. However, additional properties are often present. To illustrate this, consider a setting described by Vancroonenburg et al. (2014), where patients (suppliers) have to be allocated to hospital rooms (locations), with the additional constraint that each room should only contain patients of the same gender. We call a pair of patients with different gender a *forbidden pair*, and further, we call the set of forbidden pairs the *conflict set*. This example gives rise to the so-called Red–Blue transportation problem.

Another example, see Cao (1992), comes from storage management where containers (suppliers) need to be placed in rows of a storage yard (locations), such that costs of operations (search, load) is minimized. However, some containers are not allowed to be placed in the same row, due to their content or size. Again, two containers that cannot be placed in the same row are called a forbidden pair, and the set of forbidden pairs for a particular row form its conflict set. The resulting situation gives rise to the Transportation Problem with Exclusionary Side Constraints, which is slightly more general than the Red–Blue transportation.

Our last example comes from Chen et al. (2016), where companies (suppliers) want to promote their products to potential customers (locations). On the one hand, a customer wants to limit the number of promotions received from similar companies, inducing forbidden pairs of companies for each customer. On the other hand, companies want to geographically spread their promotion and therefore limit the number of promotions to customers living close to each other, inducing forbidden pairs of customers. Chen et al. (2016) call this problem the conflict-aware weighted bipartite b-matching problem.

We introduce the so-called transportation problem with conflicts (TPC, see Sect. 1.1 for a precise description) in which there can be conflict sets consisting of pairs of supply nodes and/or conflict sets consisting of pairs of demand nodes. Since conflict sets consists of forbidden pairs we can model a conflict set with a so-called conflict graph. A conflict graph belonging to a specific graph class indicates a specific structure of the conflict set. In this paper we study the complexity and the approximability of the TPC depending upon different classes of conflict graphs.

The paper is organized as follows. In the remaining of this section we give a formal problem description of TPC, introduce some terminology and notation, and provide an overview of the related literature. In Sect. 2 we focus on so-called conflict graphs, and present an overview of our results. Section 3 presents hardness results for different types of conflict graphs. Section 4 provides constant-factor approximation algorithms for special cases, and we conclude in Sect. 5.

1.1 Problem statement

In the transportation problem with conflicts (TPC) we are given a complete bipartite graph $(S \cup D, E)$, where (see also Fig. 1):

S : is the set of supply nodes (suppliers), with for each $i \in S$,

- A supply $s_i \in \mathbb{N}$, and
- A conflict set C_i containing forbidden pairs of demand nodes (locations),

D : is the set of demand nodes (locations), with for each $j \in D$,

- A demand $d_j \in \mathbb{N}$, and
- A conflict set F_j containing forbidden pairs of supply nodes,

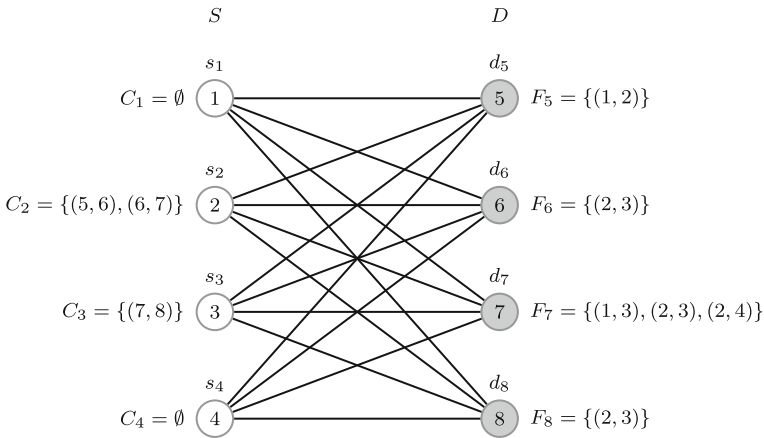


Fig. 1 Transportation problem with conflicts

E : is the complete edge set, with for each $i \in S$ and $j \in D$ an edge $(i, j) \in E$ for which we have,

- A capacity $u_{i,j} \in \mathbb{N}$, and
- A weight $w_{i,j} \in \mathbb{N}$.

Similar to the classical transportation problem we assume that $\sum_{i \in S} s_i = \sum_{j \in D} d_j$. A solution is an integral vector x , with $x_{i,j} \in \mathbb{N}$ indicating for each edge $(i, j) \in E$, how much supply is sent from supply node $i \in S$ to demand node $j \in D$. The value of a solution equals $\sum_{i \in S} \sum_{j \in D} w_{i,j} \cdot x_{i,j}$.

Given a solution x , we say that a conflict occurs if $x_{i,j_1} > 0$ and $x_{i,j_2} > 0$, while $(j_1, j_2) \in C_i$ for some $i \in S$. Likewise, a conflict occurs if $x_{i_1,j} > 0$ and $x_{i_2,j} > 0$, while $(i_1, i_2) \in F_j$. A solution x is not feasible when a conflict occurs or when $x_{i,j} > u_{i,j}$ for an edge $(i, j) \in E$.

We say that demand (supply) constraints are **fulfilled** if for all $j \in D$, $\sum_{i \in S} x_{i,j} = d_j$ (for all $i \in S$, $\sum_{j \in S} x_{i,j} = s_i$). We say that demand (supply) constraints are **respected** if for all $j \in D$, $\sum_{i \in S} x_{i,j} \leq d_j$ (for all $i \in S$, $\sum_{j \in S} x_{i,j} \leq s_i$).

We consider two problems:

- (1) Find a feasible solution fulfilling all demand and supply constraints, while minimizing the value of a solution (min-TPC), and
- (2) Find a feasible solution respecting all demand and supply constraints, while maximizing the value of a solution (max-TPC).

1.2 Terminology and notation

An important tool in our analysis are conflict graphs. We build an induced *conflict graph* G_{F_j} for each demand node $j \in D$ as follows: there is a vertex for each supply node and two vertices are connected if and only if the corresponding supply nodes constitute a forbidden pair in F_j . A similar procedure is used to build an induced conflict graph G_{C_i} for each supply node $i \in S$. For an example, we refer to Figs. 2 and 3 for the conflict graphs of C_1 and F_7 from Fig. 1.

Clearly, in case of identical conflict sets F , the induced conflict graphs are identical as well; we then refer to *the* conflict graph G_F .

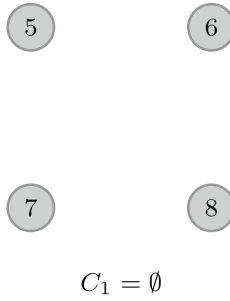


Fig. 2 Conflict graph G_{C_1}

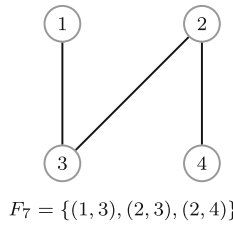


Fig. 3 Conflict graph G_{F_7}

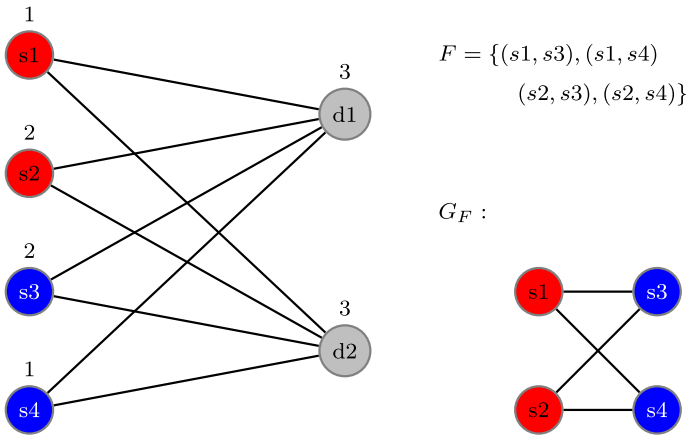


Fig. 4 Red-Blue transportation

The use of conflict graphs allows us to easily express special structures occurring in the conflict sets. For example, consider the previously mentioned Red-Blue transportation problem, in which we need to assign male and female patients to hospital rooms. In the Red-Blue transportation problem we have an identical conflict set F that contains every pair consisting of a male and a female patient. Equivalently, the conflict graph induced by F is complete bipartite. Indeed, in the conflict graph there is an edge between every red supply node (female patient) and every blue supply node (male patient), see also Fig. 4.

Throughout this paper we will discuss several special cases of TPC. We use a three-field notation to systematize the different special cases of TPC: $TPC(\alpha, \beta, \chi)$.

Definition 1 $\text{TPC}(\alpha, \beta, \chi)$ is a special case of TPC where;

$\alpha \in \{C_i, C, C^B, \emptyset\}$ describes the nature of the conflict sets of the supply nodes.

C_i : arbitrary conflict sets, each supply node $i \in S$ has a (possible empty) conflict set.

C : identical conflict sets, each supply node $i \in S$ has conflict set C .

C^B : identical conflict sets, and its induced conflict graph G_C is complete bipartite.

\emptyset : no conflict between demand nodes, each supply node $i \in S$ has an empty conflict set.

$\beta \in \{F_j, F, F^B, \emptyset\}$ describes the nature of the conflict sets of the demand nodes.

F_j : arbitrary conflict sets, each demand node $j \in D$ has a (possible empty) conflict set.

F : identical conflict sets, each demand node $j \in D$ has conflict set F .

F^B : identical conflict sets, and its induced conflict graph G_F is complete bipartite.

\emptyset : no conflict between supply nodes, each demand node $j \in D$ has an empty conflict set.

$\chi \in \{u_{i,j}, \infty, 1\}$ describes the nature of the edge capacities.

$u_{i,j}$: arbitrary capacities.

∞ : uncapacitated, or equivalently $u_{i,j} \geq \min\{s_i, d_j\}$

1: then $u_{i,j} = 1$, for every edge $(i, j) \in E$.

We call the TPC *one-sided* if either $C_i = \emptyset$ for all supply nodes $i \in S$, or $F_j = \emptyset$ for all demand nodes $j \in D$.

1.3 Related literature

1.3.1 Special cases and applications

We first explicitly point out how the three examples mentioned in the Introduction are special cases of TPC.

Conflict-aware weighted Bipartite b -matching (CA-WBM): Chen et al. (2016) describe the conflict-aware weighted Bipartite b -matching problem mentioned in the introduction. In their most general problem description there is, as input, a threshold t , the number of conflicts allowed in a feasible solution. The problem with $t = 0$ is a special case of max-TPC where we have edge capacities $u_{i,j} = 1$ for each $(i, j) \in E$, identical conflict sets $C_i = C$ for each supply node $i \in S$ and identical conflict set $F_j = F$ for each demand node $j \in D$, i.e. max-TPC($C, F, 1$).

They prove that their problem is NP-hard even if the problem is one-sided and no conflicts ($t = 0$) are allowed. Let d be the maximum degree of both conflict graphs G_C and G_F . They present a greedy $(2 + d)$ -approximation for the one-sided case. The same analysis can be used to give a greedy $(2 + 2d)$ -approximation algorithm for the two-sided case. Transportation problem with exclusionary side constraints (TPESC): Cao (1992) describes the transportation problem with exclusionary side constraints mentioned in the introduction. This problem is a special case of min-TPC where we have one-sided conflicts and no edge capacities, i.e. min-TPC(\emptyset, F_j, ∞).

Cao (1992) and later Sun (2002) describe Branch and Bound approaches to solve the problem. Cao and Uebe (1995) and later Sun (1998) use Tabu-Search methods to solve the problem. Syarif and Gen (2003) describe a genetic algorithm. Goossens and Spieksma (2009) prove that this problem is strongly NP-hard even if the conflict sets are identical, i.e. $F_j = F$ for all $j \in D$. They also prove that no polynomial time constant factor approximation exists even if there is only one non-empty conflict set and $|D| = 2$ (unless $P = NP$).

Red–Blue transportation problem (RBTP): Vancroonenburg et al. (2014) describe the Red–Blue transportation problem mentioned in the introduction. RBTP is defined for both the minimization as well as the maximization objective. This problem is a special case of TPC where we have a one-sided identical conflict set F and no edge capacities. Actually, the conflict set F induces a complete bipartite graph as conflict graph (see also Sect. 2), i.e. RBTP can be described as $\text{TPC}(\emptyset, F^B, \infty)$.

For the minimization objective they prove that no constant-factor approximation algorithm for RBTP can exist, even if $s_i = 1$ and $d_j = 3$ for all $i \in S$, $j \in D$ (unless $P = NP$). For the maximization objective the problem remains NP-hard and they present three different $\frac{1}{2}$ -approximation algorithms.

1.3.2 Other related problems

Max flow with disjunctive constraints: In the maximum flow problem with negative disjunctive constraints we are given a directed graph $G = (N, A)$, source $s \in N$, a sink $t \in N$, and there is a capacity $u_{i,j}$ for each arc $(i, j) \in A$. In addition there are negative disjunctive constraints, meaning that pairs of arcs are given, at most one of which is allowed to carry a positive amount of flow in a feasible solution. The goal is to maximize the amount of flow sent from s to t while respecting the capacities and the negative disjunctive constraints.

Similar to the construction sketched in Sect. 1.2 one can build a conflict graph in which there is a node for each arc (i, j) and there is an edge between two arcs if they are in conflict.

Pferschy and Schauer (2013) prove that this problem is NP-hard, even if the induced conflict graph consists only of unconnected edges and directed graph $G = (N, A)$ only consists of disjoint paths of length three.

Notice that this problem is more general than max-TPC due to the arbitrary graph structure.

Maximum matching with disjunctive constraints: Another well-known problem related to the classical Transportation problem is maximum matching. In the maximum matching problem with negative disjunctive constraint we are given a graph $G = (V, E)$ with edge weights and negative disjunctive constraints on the edges. The goal is to find a matching with maximum weight. Darmann et al. (2011) show that the maximum matching problem with negative disjunctive constraints is NP-hard even for conflict graphs where every connected component is a single edge. Again, this result does not apply to TPC due to the fact that the graph in TPC is a bipartite graph.

Two-to-one assignment problem: Goossens et al. (2012) describe the two-to-one assignment problem (2-1-AP) as follows. We are given a set X of $2n$ elements and a set Y of n elements. A feasible triple cost $c_{i_1, i_2, j}$ is associated to two distinct elements $i_1, i_2 \in X$ and an element $j \in Y$; the goal is to select n feasible triples of minimum total cost such that each element occurs once in a selected triple.

They prove that the existence of a PTAS would imply $P = NP$ and they provide a constant-factor approximation for the special case of decomposable costs satisfying the triangle inequality.

It is not difficult to see that $\text{min-TPC}(\emptyset, F, \infty)$ with $s_i = 1$ for all $i \in S$ and $d_j = 2$ for all $j \in D$ is a special case of 2-1-AP. Indeed, observe that any feasible solution of $\text{min-TPC}(\emptyset, F, \infty)$ with $d_j = 2$ also consists of triples, where a triple consists of a demand node and two supply nodes that are not in conflict. Given an instance of min-TPC, we create an instance of the 2-1-AP as follows. Let set X contain all supply nodes

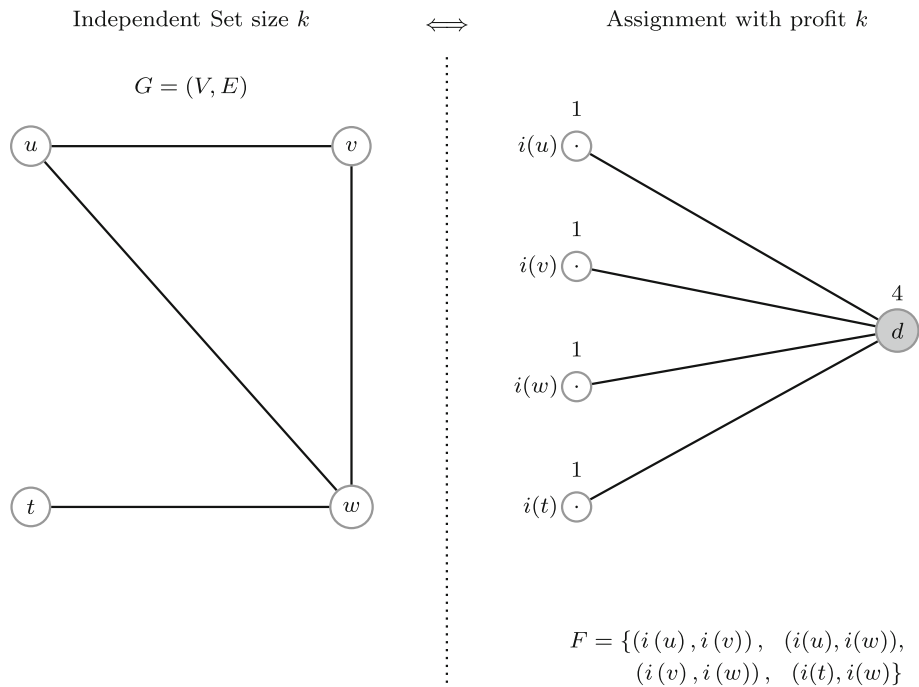


Fig. 5 MIS and max-TPC

and set Y all demand nodes. Consider now a triple consisting of a node $j \in Y$ and two nodes $i_1, i_2 \in X$, that are not in conflict for demand node $j \in D$. We set its cost to be $c_{i_1, i_2, j} = w_{i_1, j} + w_{i_2, j}$; the cost of any other feasible triple is set to a large number. This shows that min-TPC is a special case of 2-1-AP. The costs we define however are not decomposable in the sense of Goossens et al. (2012).

2 A preview

Let us first informally argue why max-TPC in case of a single demand node is an NP-hard problem. Clearly, in any feasible solution, the set of supply nodes $i \in S$ that send flow to a particular demand node $j \in D$ (i.e. those $i \in S$ with $x_{i,j} > 0$) form an independent set in the induced conflict graph of that demand node $j \in D$. Indeed, since apparently none of the supply nodes assigned are in conflict, there is no edge connecting them in the conflict graph. This shows that Maximum Independent Set (MIS) is a special case of the TPC. Indeed, given an instance of MIS, i.e. a graph $G = (V, E)$ with $|V| = n$, we can construct an instance of max-TPC with n supply nodes each with supply 1 and a single demand node with demand n whose induced conflict graph is equivalent to the graph G (see also Fig. 5), all edge weights are 1.

This means that max-TPC with a single demand node is at least as difficult the maximum independent set problem.

Note that for min-TPC the above construction does not work, as min-TPC requires that all supply and demand constraints are fulfilled. In the above construction supply constraints

can only be fulfilled if the graph G of the MIS instance contains no edges, i.e. no pair of supply nodes is in conflict. However, with a similar reduction, it is shown that no PTAS exists for $\text{TPC}(\emptyset, F_j, \infty)$, even with $|D| \geq 2$ and exactly one demand node having a non-empty conflict set (unless $P = NP$), (Goossens and Spieksma 2009).

We now review the existing complexity results for TPC.

Goossens and Spieksma (2009) prove that:

- (1) $\text{min-TPC}(\emptyset, F, \infty)$ is weakly NP-hard for $|D| = 2$,
- (2) $\text{min-TPC}(\emptyset, F_j, \infty)$ is weakly NP-hard for each fixed $|S| \geq 2$, and
- (3) $\text{min-TPC}(\emptyset, F, \infty)$ is strongly NP-hard for each fixed $|D| \geq 3$.

These results do not assume any structure on the conflict graph G_F . Motivated by an application, Vancroonenburg et al. (2014) assume that G_F is a complete bipartite graph; they prove that:

- (1) Both $\text{min-TPC}(\emptyset, F^B, \infty)$ and $\text{max-TPC}(\emptyset, F^B, \infty)$ remain NP-hard even if $s_i = 1$ for all $i \in S$ and $d_j = 3$ for all $j \in D$, and
- (2) $\text{min-TPC}(\emptyset, F^B, \infty)$ remains NP-hard, even if edge set E is complete, all edge weights are equal and there are only 2 supply nodes with equal supply.

Our results focus on the nature of the conflict graph G_F . First, we point out that, in the extreme cases where either F is empty (i.e. G_F contains no edges), or F contains all pairs (i.e. G_F is a clique), the resulting problem $\text{TPC}(\emptyset, F, u_{i,j})$ is easily seen to be polynomial solvable.

Second, it is clear from our previous discussion that any graph class for which MIS is NP-hard, the corresponding problem $\text{TPC}(\emptyset, F, \infty)$ is NP-hard as well. However, there are graph classes for which independent set is easy, i.e. polynomial solvable; thus, for these graph classes the previous argument does not allow us to conclude that the corresponding special case of TPC is NP-hard. To deal with these graph classes, we provide the following results.

- (1) $\text{TPC}(\emptyset, F, \infty)$ remains NP-hard, even if G_F is a matching (Theorem 1); this immediately implies NP-hardness of $\text{TPC}(\emptyset, F, \infty)$ in case G_F is a bipartite graph or an interval graph.
- (2) $\text{TPC}(\emptyset, F, \infty)$ remains NP-hard, even if G_F is a simple path (Theorem 2); this immediately implies NP-hardness of $\text{TPC}(\emptyset, F, \infty)$ in case G_F is a tree.
- (3) $\text{TPC}(\emptyset, F, \infty)$ remains NP-hard, even if G_F is a split graph (Theorem 3).

Notice that indeed, MIS is polynomially solvable for the graph classes mentioned above.

Further, in case of a complete bipartite conflict graph, we close the gap between polynomial and NP-complete cases of $\text{TPC}(\emptyset, F^B, \infty)$:

- (4) $\text{TPC}(\emptyset, F^B, \infty)$ remains NP-hard, even if supply $s_i = 1$ for each supply node $i \in S$ and all demand $d_j = 2$ for each demand node $j \in D$, thereby settling a case left open by Vancroonenburg et al. (2014) (Theorem 4).

In addition, we give approximation results depending on the approximation factor for finding a certain independent set, for the following special cases of max-TPC:

- (1) $\text{max-TPC}(\emptyset, F_j, \infty)$ with supply $s_i = 1$ for each $i \in S$ (Theorem 6);
- (2) $\text{max-TPC}(\emptyset, F_j, 1)$ (Theorem 7);
- (3) $\text{max-TPC}(C, F_j, \infty)$ with supply $s_i = 1$ for each $i \in S$ and a valid q -colouring on the conflict graph induced by C (Theorem 8);
- (4) $\text{max-TPC}(C, F_j, 1)$ with a valid q -colouring on the conflict graph induced by C (Theorem 9).

Finally, we mention here that assuming a constant number of demand nodes, (i.e. $|D|$ is fixed), can lead to some cases that are solvable in polynomial time. For instance, we point out that, in case of a fixed $|D|$, and G_F being a star (centered around supply node $s \in S$), it suffices to solve $2^{|D|}$ classical transportation problems. Indeed, observe that for a single demand node $d \in D$, either node d is supplied exclusively by node s , or, node s cannot be used to supply node d . This observation implies that solving $2^{|D|}$ classical transportation problems is sufficient to solve $\text{TPC}(\emptyset, F, \infty)$. In the same spirit, we claim that if $|F|$ has a constant size, it is sufficient to solve $(2^{|F|})^{|D|}$ classical transportation problems, in order to solve $\text{TPC}(\emptyset, F, \infty)$.

3 Proofs of our complexity results

As announced above, in this section we show that $\text{TPC}(\emptyset, F, \infty)$ is NP-hard in case G_F is a bipartite graph, interval graph, a matching, a simple path (Sect. 3.1) or a split graph (Sect. 3.2). In Sect. 3.3, we show that $\text{TPC}(\emptyset, F^B, \infty)$ remains NP-hard even if $s_i = 1$ for all $i \in S$ and $d_j = 2$ for all $j \in D$.

All proofs in this section assume a one-sided conflict set F (thus, conflict sets C_i are empty) and there is an infinite capacity on the edges. In fact, observe that to specify an instance of the resulting TPC problem it is sufficient to specify S (the set of supply nodes), D (the set of demand nodes), F (the conflict set), and w (the edge weights, edge set E is complete).

3.1 When the conflict graph is a matching or simple path

In this section we show that even if the induced conflict graph G_F is a perfect matching or a simple path, both max-TPC and min-TPC remain NP-hard. This implies that the problem remains NP-hard for bipartite graphs, interval graphs, trees and forests. We first prove that max-TPC and min-TPC remain NP-hard even if the induced conflict graph of F is a matching (only contains disconnected edges); after that we show that it is not difficult to extend the proof such that the results holds for a simple path as well.

To prove the theorem below we reduce our problem to monotone 1-in-3SAT, which is known to be NP-Complete (Garey and Johnson 1979). An instance is defined as follows.

Given: A set of n variables $\{v_1, \dots, v_n\}$, and a set of m clauses $\{c_1, \dots, c_m\}$, each containing 3 variables, each variable in positive form (unnegated).

Question: Is there a truth-assignment that satisfies all clauses in C such that in each clause exactly 1 variable is TRUE?

Theorem 1 *Min-TPC* (\emptyset, F, ∞) is NP-hard, even if the induced conflict graph of G_F is a (perfect) matching.

Proof Given an instance I of monotone 1-in-3SAT, let λ_v be the number of clauses in which variable v occurs. We construct an instance of min-TPC (\emptyset, F, ∞) as follows (see also Fig. 6).

- S: For each variable v , we have supply nodes $i(v)$, $i(\bar{v})$, each with supply λ_v . Hence we have $2n$ supply nodes.
- D: For each clause $c \in C$ we have a demand node $j(c)$ with demand 1, and a demand node $j(\bar{c})$, with demand 2. And, for each variable v we have a demand node $j(v)$, with demand λ_v . Hence, we have $2m + n$ demand nodes.

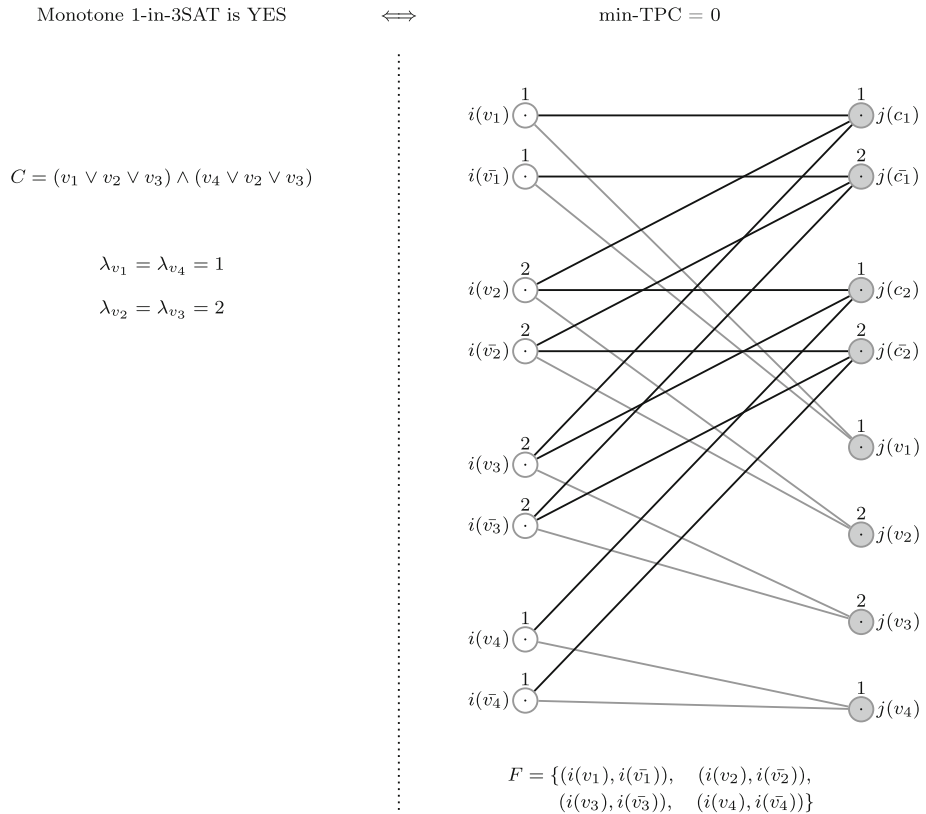


Fig. 6 Monotone 1-in-3SAT and TPC(\emptyset, F, ∞), only edges with weight 0 are shown

F: The conflict set contains forbidden pairs $(i(v), i(\bar{v}))$, for every variable v . Note that, the conflict graph G_F is indeed a perfect matching.

E: The weights on the edges are defined as follows:

- For each variable v that occurs (positively) in clause c we have $w_{i(v),j(c)} = 0$ and $w_{i(\bar{v}),j(\bar{c})} = 0$.
- For each variable v we have $w_{i(v),j(v)} = 0$ and $w_{i(\bar{v}),j(v)} = 0$.
- For any other edge $e \in E$ we have $w_e = 1$.

We now argue that an instance of monotone 1-in-3SAT is a YES-instance if and only if the corresponding instance of min-TPC(\emptyset, F, ∞) has a solution with cost 0.

Suppose we have a YES-instance of monotone 1-in-3SAT, with a certificate indicating whether variable v is set to TRUE or FALSE. We create the following solution of the corresponding TPC instance.

If variable v is set to TRUE, then:

- For each clause c in which variable v occurs, 1 unit of supply is sent from supply node $i(v)$ to demand node $j(c)$,
- λ_v units of supply are sent from supply node $i(\bar{v})$ to demand node $j(v)$.

If variable v is set to FALSE, then:

- For each clause c in which variable v occurs, 1 unit of supply is sent from supply node $i(\bar{v})$ to demand node $j(\bar{c})$,
- λ_v units of supply are sent from supply node $i(v)$ to demand node $j(v)$.

It is easy to see that no conflicts occur in this solution. Hence we only need to show that all supply and demand constraints are fulfilled.

Since λ_v equals the number of clauses containing variable v , we know that all supply constraints are fulfilled and that the demand for each demand node $j(v)$ is fulfilled.

We now show that the demand constraints are fulfilled as well. Since we have a YES-certificate, we know that exactly 1 variable is set to TRUE for each clause c . Hence, the demand for each demand node $j(c)$ is fulfilled. Furthermore it follows that in each clause c two variables are set FALSE, hence, the demand for each demand node $j(\bar{c})$ is fulfilled as well.

Observe that since we only use edges of weight 0, the cost of the solution is 0.

Now, suppose that we have a solution with cost 0 of the TPC instance.

This solution fulfills all supply and demand constraints and only uses edges of weight 0. Thus, we know that either λ_v units of supply are sent from $i(v)$ to $j(v)$, or from $i(\bar{v})$ to $j(v)$. (Since $i(v)$ and $i(\bar{v})$ are in conflict, there is no solution in which both $i(v)$ and $i(\bar{v})$ send supply to $j(v)$.) We will construct a truth-assignment as follows: for each variable v , if $i(v)$ sends supply to $j(v)$, then we set variable v to FALSE, else we set variable v to TRUE. We now argue that this is indeed a YES-certificate for monotone 1-in-3SAT.

It is easy to see that each clause has at least one variable set to TRUE. Indeed, the solution of the corresponding TPC instance with cost 0 is feasible, hence all demand constraints are fulfilled. Thus, demand node $j(c)$ receives one unit of supply from one of the supply nodes $i(v)$. Hence, for each clause at least one variable is set to TRUE.

Likewise, we know that all supply constraints are fulfilled. Thus, each supply node corresponding to a variable v that is set to TRUE distributes λ_v units of supply over demand nodes $j(c)$. Hence, the sum of λ_v of the variables v that are set to TRUE equals m , the number of clauses. Hence, each clause has exactly one variable set to TRUE. \square

Corollary 1 *There is no constant-factor approximation algorithm for $\text{min-TPC}(\emptyset, F, \infty)$, even if the induced conflict graph of F is a (perfect) matching, unless $P = NP$.*

Corollary 2 *Max-TPC(\emptyset, F, ∞) is NP-hard, even if the induced conflict graph G_F is a (perfect) matching.*

Proof For $\text{max-TPC}(\emptyset, F, \infty)$ the reduction is basically the same, the only change is the edge-weights. The weights on the edges are defined as follows:

- For each variable v that occurs in clause c we have $w_{i(v),j(c)} = 1$ and $w_{i(\bar{v}),j(\bar{c})} = 1$.
- For each variable v we have $w_{i(v),j(v)} = 1$ and $w_{i(\bar{v}),j(v)} = 1$.
- For any other edge $e \in E$ we have $w_e = 0$.

Note that now we have that a monotone 1-in-3SAT instance is a YES-instance, if and only if $\text{max-TPC}(\emptyset, F, \infty)$ has solution of profit $\sum_{v \in V} 2 \cdot \lambda_v$. \square

It follows immediately from Theorem 1 and Corollary 2 that $\text{TPC}(\emptyset, F, \infty)$ remains NP-hard even if the induced conflict graph G_F is a bipartite graph or an interval graph.

Let us now informally discuss the version of $\text{TPC}(\emptyset, F, \infty)$ for which the induced conflict graph of F is a simple path. Given an instance of $\text{TPC}(\emptyset, F, \infty)$, of which the induced conflict graph of F only contains disconnected edges, it is easy to create an instance of $\text{TPC}(\emptyset, F, \infty)$ of which the induced conflict graph F is a simple path. We can add $n - 1$ dummy supply

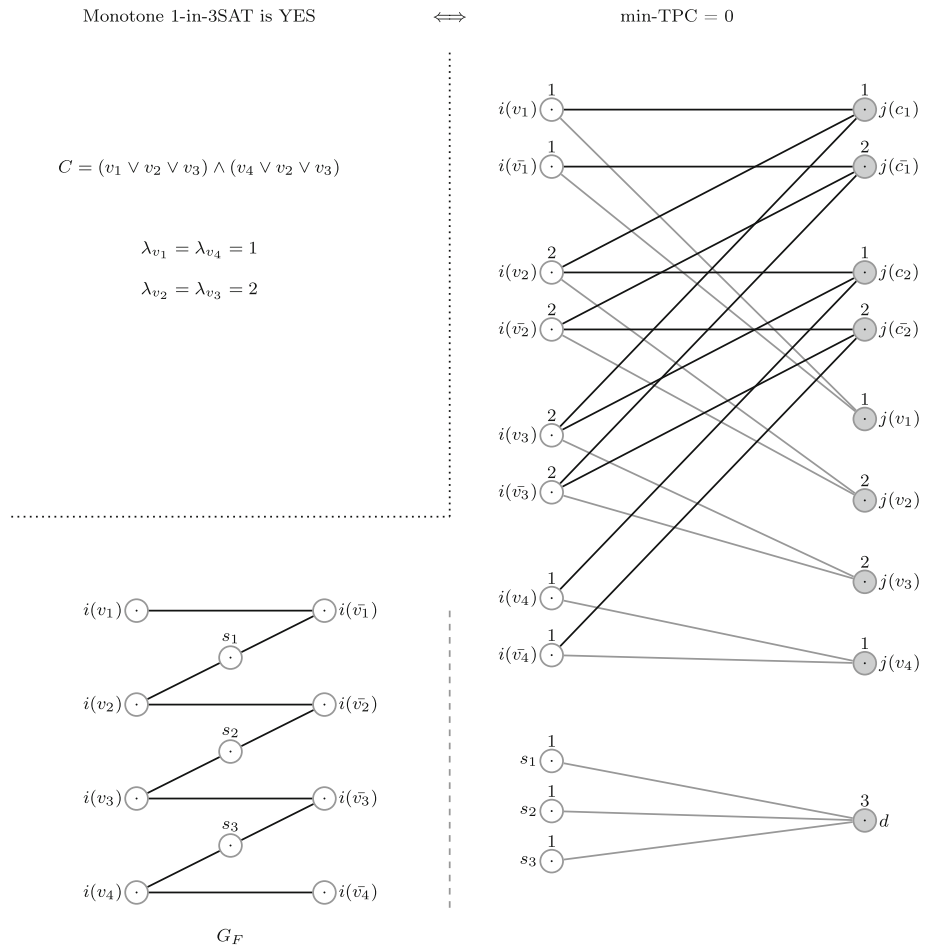


Fig. 7 Monotone 1-in-3SAT and TPC with simple path as induced conflict graph

nodes, each with supply 1, and 1 dummy demand node with demand $n - 1$ (recall that n is the number of variables in the monotone 1-in-3SAT instance). Edge set E is by definition a complete edge set, so we add the missing edges connecting the newly added dummy nodes. The edges connecting the dummy supply nodes to the dummy demand node will have weight 0, all other added edges will have weight 1. In the induced conflict graph we connect the dummy supply nodes to the non-dummy supply nodes, such that the induced conflict graph becomes a simple path (see Fig. 7). Observe that this does not affect the equivalence as described in the proof of Theorem 1.

Theorem 2 *TPC(\emptyset, F, ∞) is NP-hard, even if the induced conflict graph of G_F is a simple path.*

Corollary 3 *There is no constant-factor approximation algorithm for min-TPC(\emptyset, F, ∞), even if the induced conflict graph of F is a simple path (unless $P = NP$).*

It follows immediately from Theorem 2 that TPC(\emptyset, F, ∞) remains NP-hard even if the induced conflict graph G_F is a tree.

3.2 When the conflict graph is a split graph

In this section we show that for every graph class \mathcal{G} for which the so-called minimum sum colouring problem is NP-hard, the TPC restricted to conflict graphs from \mathcal{G} is also NP-hard. For some graph classes (for instance bipartite graphs and interval graphs), this result provides an alternative proof of $\text{TPC}(\emptyset, F, \infty)$, since minimum sum colouring is NP-hard for these classes, see Bar-Noy and Kortsarz (1998), Marx (2005), Szkaliczki (1999). However, there exists a graph class not captured by Theorems 1 and 2 for which MIS is easy, while minimum sum colouring is NP-hard; the class of split graphs. We refer to Salavatipour (2000) where it is shown that minimum sum colouring is NP-hard for split graphs. Summarizing, the results in this section imply that $\text{TPC}(\emptyset, F, \infty)$ remains NP-hard even if the conflict graph is a split graph.

The minimum sum colouring problem is defined as follows.

Given: A graph $G = (V, E_G)$ with $|V| = n$.

Goal: Find a colouring $c : V \rightarrow \{1, \dots, n\}$ of minimum weight $\sum_{v \in V} c(v)$ such that if $(v, u) \in E_G$, then $c(v) \neq c(u)$.

Theorem 3 *Let \mathcal{G} be a graph class for which the minimum sum colouring problem is NP-hard. $\text{Min-TPC}(\emptyset, F, \infty)$ is NP-hard, even if the induced conflict graph $G_F \in \mathcal{G}$.*

Proof Given an instance of the minimum sum colouring problem, specified by a graph $G = (V, E_G)$, we construct an instance of $\text{min-TPC}(\emptyset, F, \infty)$ as follows (see also Fig. 8).

- S: For each vertex $v \in V$ we have supply node $i(v)$ with supply 1. We also have an additional dummy node s with supply $n(n-1)$. Hence, we have $n+1$ supply nodes.
- D: We have n demand nodes each with demand n (one for each colour). Hence, we have $D := \{1, \dots, n\}$.
- F: We have a forbidden pair $(i(v), i(u)) \in F$ for each edge in $(v, u) \in E_G$, i.e. $F := E_G$. Hence, the induced conflict graph G_F is equivalent to graph $G = (V, E_G)$.
- E: The weights on the edges are defined as follows:

- For each $i(v) \in S \setminus \{s\}$ and for each $j \in D$: $w_{i(v), j} = j$;
- For each $j \in D$: $w_{s, j} = 0$.

The goal is to find a feasible solution of minimum cost. This specifies the instance of min-TPC .

We will now argue that a solution of minimum sum colouring with cost Z corresponds to a solution of min-TPC with cost Z and vice versa.

Suppose we have a feasible solution for the minimum sum colouring problem instance, meaning that there is a colour $c(v)$ assigned to each $v \in V$; the resulting cost of this solution is $Z = \sum_{v \in V} c(v)$. In the corresponding instance of min-TPC we send the single unit of supply of node $i(v) \in S$ to node $j \in D$ if vertex v is coloured with colour j , i.e. if $c(v) = j$; the weight of this flow equals j . Further, we distribute the supply from node s to all nodes $j \in D$ such that all supply and demand constraints are fulfilled; the weight of this flow equals 0. Since the colouring is a valid colouring in $G = (V, E)$, we know that vertices adjacent in G received distinct colours; in other words, no demand node receives supply from two supply nodes that are a forbidden pair in F . Also, the total value of this solution equals Z .

Now, suppose we have a feasible solution for our min-TPC instance with a total value Z . Then each $i(v) \in S$ sends a single unit of flow to exactly one demand node $j \in D$ (supply equals 1); the weight of this flow equals j . In the corresponding minimum sum colouring

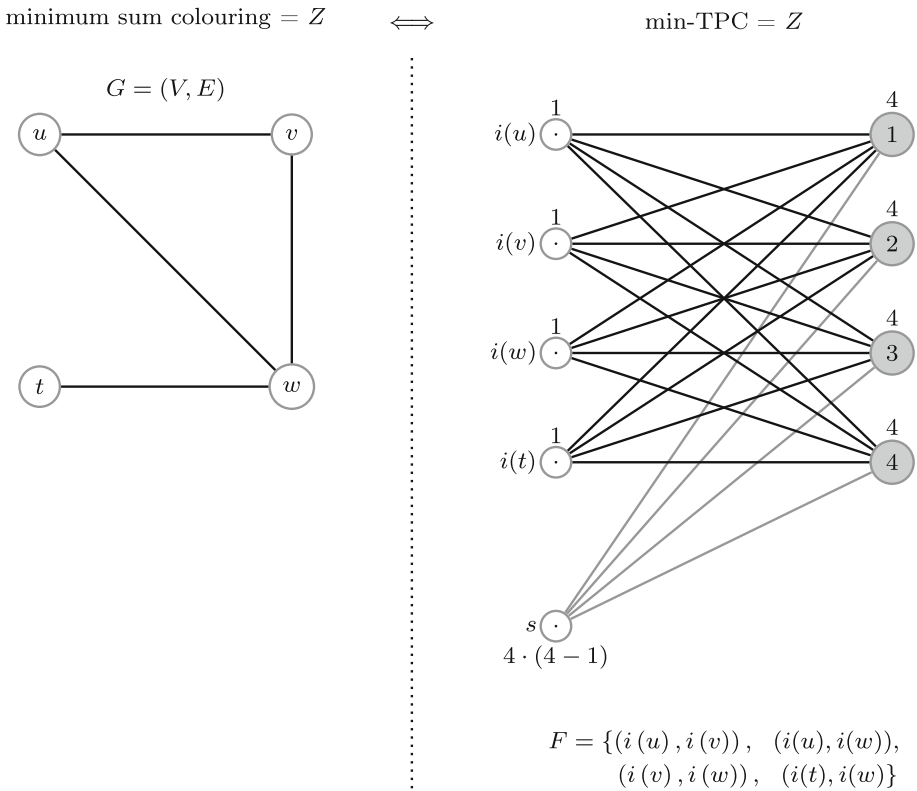


Fig. 8 Example of construction, where $n = 4$

instance we give vertex $v \in V$ colour j if supply node $i(v) \in S$ sends supply to demand node $j \in D$. Since no conflicts are allowed in our assignment, we know that no two vertices adjacent in G receive the same colour, and the total cost of this solution, $\sum_v c(v)$, equals Z . □

Corollary 4 *Let \mathcal{G} be a graph class for which the minimum sum colouring problem is NP-hard. Max-TPC(\emptyset, F, ∞) is NP-hard, even if the induced conflict graph $G_F \in \mathcal{G}$.*

Proof For max-TPC the reduction is basically the same, the only change are the edge weights. For max-TPC we define the weight on the edges as follows:

- For each $i(v) \in S \setminus \{s\}$ and for each $j \in D$: $w_{i(v),j} = n - j$.
- For each $j \in D$: $w_{s,j} = 0$.

Note that in this case the profit of the assignment in max-TPC does not equal the cost of the solution to the minimum colouring problem. However, it still holds that, if we are able to solve max-TPC in polynomial time, we will find an optimal solution to minimum sum colouring in polynomial time as well. □

It follows immediately from Theorem 3 and Corollary 4 that TPC(\emptyset, F, ∞) remains NP-hard even if the induced conflict graph G_F is a split graph. Note that in Sect. 3.1 we have already shown that TPC(\emptyset, F, ∞) remains NP-hard even if the induced conflict graph G_F is a bipartite or interval graph.

3.3 A complete bipartite conflict graph with $s_i = 1$ and $d_j = 2$

As mentioned before $\text{TPC}(\emptyset, F^B, \infty)$ is also known as the Red–Blue transportation problem (RBTP). Recall that for RBTP we know that the problem is NP-hard even if all supply $s_i = 1$ for all supply nodes $i \in S$ and all demand $d_j = 3$ for all demand nodes $j \in D$.

Vancroonenburg et al. (2014) state that it remains open whether the problem is still NP-hard even if all demand equals 2. We prove here that this problem remains NP-hard. To do so we make a reduction from a special case of SAT, known as (2, 2)-SAT, which is proven to be NP-hard by Ratner and Warmuth (1990). An instance I of (2, 2)-SAT is defined as follows.

Given: A set of n variables $\{v_1, \dots, v_n\}$ and a set of m clauses $C = \{c_1, \dots, c_m\}$, such that every variable occurs exactly twice in positive form and exactly twice in negative form.

Question: Is there a truth-assignment that satisfies all clauses in C ?

Theorem 4 *Min-TPC* (\emptyset, F^B, ∞) is NP-hard, even if $s_i = 1$ and $d_j = 2$ for all $i \in S$ and $j \in D$.

Proof We stress that (2,2)-SAT does not impose any bound on the size of the clauses. Without loss of generality we assume that $|C|$ is even.

Recall that the induced conflict graph G_{FB} is a complete bipartite graph. Hence, it is sufficient to define two sets of supply nodes, S_R and S_B . Then, the conflict set F^B contains a conflict for each pair of nodes with one node in S_R and one node in S_B . We shall split the set of supply nodes S into red supply nodes, S_R , and blue supply nodes, S_B .

Given instance I of (2,2)-SAT we construct an instance of min-TPC (\emptyset, F^B, ∞) , where all supply is 1 and all demand is 2, as follows.

S_R : For each clause $c \in C$ we have a supply node $r(c)$, and for each variable v occurring positively in clause c we have supply node $r(v, c)$.

S_B : For each clause $c \in C$ we have a supply node $b(c)$, and for each variable v occurring negatively in clause c we have supply node $b(\bar{v}, c)$.

D : For each clause $c \in C$ we have demand node $j(c)$, and for each variable v we have demand node $j(v)$, and we have n dummy nodes j_i , for $i = 1, \dots, n$.

E : The weights on the edges are defined as follows:

- For each variable v that occurs in positive form in clause c we have $w_{r(v,c),j(c)} = 0$ and $w_{r(v,c),j_i} = 0$.
- For each variable v that occurs in negative form in clause c we have $w_{b(\bar{v},c),j(c)} = 0$ and $w_{b(\bar{v},c),j_i} = 0$.
- For every clause c we have $w_{r(c),j(c)} = 0$ and $w_{b(c),j(c)} = 0$.
- Any edge connecting a supply node to a dummy demand node j_i ($i = 1, \dots, n$) has weight 0.
- For any other edge $e \in E$ we have $w_e = 1$.

Note that each of the sets S_R , S_B and D contain $2n + m$ nodes, which is even, because we assumed that the number of clauses, m , is even.

We now argue that the (2,2)-SAT instance is a YES-instance, if and only if min-TPC (\emptyset, F^B, ∞) has solution of value 0 (see also Fig. 9).

First, suppose that we have a solution with value 0 of the corresponding TPC instance.

This solution fulfills all supply and demand constraints and only uses edges of weight 0. Hence, we know that a demand node $j(v)$ receives supply either from the two red supply

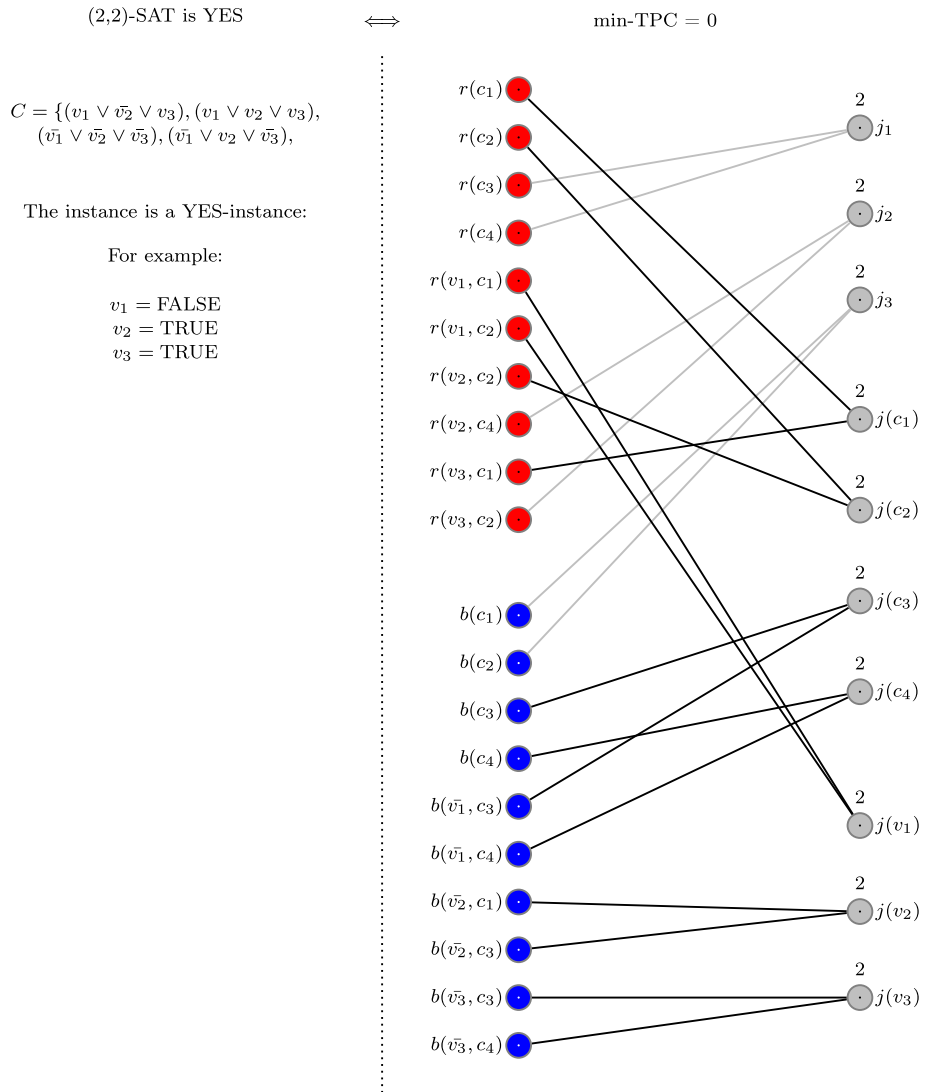


Fig. 9 Example solution for (2,2)-SAT and $\text{TPC}(\emptyset, F^B, \infty)$

nodes representing the two positive occurrences of variable v in a clause, or from the two blue supply nodes representing the two negative occurrences of variable v in a clause. If it receives supply from the two blue supply nodes we set variable v to be TRUE; otherwise we set variable v to be FALSE.

We now argue that this is a satisfying truth-assignment. Consider an arbitrary clause $c \in C$ and the corresponding demand node $j(c)$. Demand node $j(c)$ receives supply from at least one supply node representing the occurrence of a variable v in that clause. Suppose that supply node $r(v, c)$ sends supply to $j(c)$ and therefore no supply to $j(v)$. Hence variable v is set to TRUE and clause c in which v occurs in positive form is satisfied. Suppose $b(\bar{v}, c)$,

sends supply to $j(c)$ and therefore no supply to $j(v)$. Hence variable v is set to FALSE and clause c in which it occurs in negative form is satisfied.

Second, suppose we have a satisfying truth-assignment.

If variable v has been set to TRUE, then demand node $j(v)$ receives supply from the two blue supply nodes representing the two negative occurrences of variable v . If variable v has been set to FALSE, then demand node $j(v)$ receives supply from the two red supply nodes representing the two positive occurrences of variable v .

Now consider a clause c and pick one variable v that satisfies this clause. If variable v occurs in positive form, then we know v has been set to TRUE in order to satisfy the clause. In this case both $r(v, c)$ and $r(c)$ send supply to demand node $j(c)$. If variable v occurs in negative form, then we know v has been set to FALSE in order to satisfy the clause. In this case both $b(\bar{v}, c)$ and $b(c)$ send supply to demand node $j(c)$.

So far we have assigned an even number of red supply nodes and an even number of blue supply nodes. The remaining supply nodes can therefore be assigned to the remaining dummy demand nodes, such that each demand node receives supply of only one colour. \square

Corollary 5 *There is no constant-factor approximation algorithm for $\text{min-TPC}(\emptyset, F^B, \infty)$, even if $s_i = 1$ and $d_j = 2$, unless $P = NP$.*

Corollary 6 *$\text{Max-TPC}(\emptyset, F^B, \infty)$ is NP-hard, even if $s_i = 1$ and $d_j = 2$.*

Proof For max-TPC the reduction is basically the same, the only change is the edge-weights. The weights on the edges are defined as follows:

- For each variable v that occurs in positive form in clause c we have $w_{r(v,c),j(v)} = 1$ and $w_{r(v,c),j(c)} = 1$.
- For each variable v that occurs in negative form in clause c we have $w_{b(\bar{v},c),j(v)} = 1$ and $w_{b(\bar{v},c),j(c)} = 1$.
- For every clause c we have $w_{r(c),j(c)} = 1$ and $w_{b(c),j(c)} = 1$.
- Any edge connecting a supply node to a dummy demand node has weight 1.
- For any other edge $e \in E$ we have $w_e = 0$.

Note that now we have that (2,2)-SAT instance is a YES-instance, if and only if $\text{max-TPC}(\emptyset, F^B, \infty)$ has solution of value $2(2n + m)$. \square

4 Approximation for max-TPC

From the previous section we conclude that there exists no constant-factor approximation for min-TPC (unless $P = NP$). When it comes to max-TPC, we have shown in Sect. 2 that for any instance of the Maximum Independent Set Problem, there exists an equivalent instance of $\text{Max-TPC}(\emptyset, F, \infty)$. For the maximum independent set problem on graph $G = (V, E)$ with $|V| = n$, it is known that it is hard to approximate in polynomial time within a factor $n^{1-\epsilon}$, for any $\epsilon > 0$, unless $P = NP$ (Håstad 1996). Hence for $\text{max-TPC}(\emptyset, F, \infty)$ we know that no constant-factor approximation algorithm exists, even if $|D| = 1$, unless $P = NP$.

Theorem 5 *$\text{Max-TPC}(\emptyset, F, \infty)$, with $n := |S|$, is hard to approximate in polynomial time within a factor $n^{1-\epsilon}$, for any $\epsilon > 0$, even if supply $s_i = 1$ for each supply node $i \in S$, and $|D| = 1$ with $d_1 = n$ (unless $P = NP$).*

We now describe the so-called Separable Assignment Problem and show how existing approximation results can be applied to max-TPC.

4.1 Max-TPC meets the separable assignment problem

The separable assignment problem (SAP) generalizes several well-known assignment problems. An informal description of the SAP is as follows. We are given a set of bins, a set of items and a profit for assigning a certain item to a certain bin. We are also given a packing constraint for each bin. Such a packing constraint must be such that every subset of a feasible packing is also feasible. So, for example, a packing constraint cannot state that each bin contains exactly 4 items, because a subset of 3 items would no longer be feasible. A packing constraint can state that each bin contains at most 4 items. The goal is to find an assignment of items to bin that maximizes total profit.

Suppose that, for a single bin, there is a β -approximation algorithm for finding the maximum profit assignment. Fleischer et al. (2011) present the following results for SAP:

- (1) A $\beta(1 - \frac{1}{e})$ -approximation algorithm (based on Randomized LP-Rounding), and
- (2) A $(\frac{\beta}{\beta+1} - \epsilon)$ -approximation algorithm (based on Local Search).

Thus, with $\gamma = \max\{\beta(1 - \frac{1}{e}), (\frac{\beta}{\beta+1} - \epsilon)\}$, we summarize the results of Fleischer et al. (2011) by saying that they provide a γ -approximation algorithm for SAP. Note if $\beta \geq \frac{1}{e-1}$ then $\gamma = \beta(1 - \frac{1}{e})$, else $\gamma = \frac{\beta}{\beta+1} - \epsilon$.

Let us now consider the following special case of max-TPC(\emptyset, F_j, ∞), where all supply $s_i = 1$ for each supply node $i \in S$. We now show that we can describe this problem as a special case of SAP.

To do so, we view each supply node $i \in S$ in max-TPC as an item v_i in SAP and we view each demand node $j \in D$ with demand d_j in max-TPC as a bin b_j in SAP.

The packing constraint of each bin b_j states that;

- (1) The total number of packed items does not exceed d_j , and
- (2) A pair of items v_{i_1}, v_{i_2} can only be packed in bin b_j if the corresponding pair of supply nodes does not occur in the conflict set of the corresponding demand node, i.e. if $(i_1, i_2) \notin F_j$.

Note that, in our setting, finding the maximum profit solution for a single demand node $j \in D$ is equivalent to finding a maximum weighted independent set of size at most d_j in conflict graph G_{F_j} , where the weight of a node $i \in S$ equals edge weight $w_{i,j}$. Finding a maximum weighted independent set of size at most a given cardinality d (called MWIS- d) is an interesting combinatorial optimization problem in its own right; notice that this problem is more general than the well-studied maximum weight independent set problem (MWIS). Indeed, Kalra et al. (2017) study MWIS- d , and show that for bipartite graphs, MWIS- d is NP-complete (while MWIS is well-known to be solvable in polynomial time for bipartite graphs (Ahuja et al. 1993)). In addition, Kalra et al. (2017) give a $\frac{1}{2}$ -approximation algorithm for MWIS- d on bipartite graphs. Such approximation results can be used to obtain constant factor approximation algorithm for a single bin, which in turn leads to approximation results for special cases of max-TPC using Fleischer et al. (2011).

Likewise, we can use the results of a generalization of SAP, called k -SAP. In k -SAP we are given a set of bins, a set of items I , a profit for assigning an item to a certain bin and a packing constraint for each bin. Now each item $i \in I$ may be assigned to at most k_i different bins, but at most once to each bin. The goal is to find an assignment of items to bins that maximizes total profit.

Again, assume that we are given a β -approximation algorithm for finding the maximum profit assignment for a single bin, and let $k = \min_{i \in I} k_i$. Bender et al. (2015) prove that there exists a $\beta(1 - \frac{1}{e^k})$ -approximation algorithm based on Randomized LP-Rounding.

4.1.1 One-sided max-TPC

Let us first consider only one-sided constraints. The above discussion will give us the following result.

Theorem 6 *Suppose there is a β_j -approximation for finding a maximum weighted independent set of size at most d_j on the conflict graph G_{F_j} induced by F_j for each $j \in D$, let $\beta = \min_{j \in D} \beta_j$ and let $\gamma = \max\{\beta(1 - \frac{1}{e}), (\frac{\beta}{\beta+1} - \epsilon)\}$.*

Then there exists a γ -approximation algorithm for max-TPC(\emptyset, F_j, ∞) with supply $s_i = 1$ for each supply node $i \in S$.

As mentioned before, suppose that in each of the conflict graphs G_{F_j} induced by F_j for each $j \in D$ we can find a maximum weight independent set of size at most d_j in polynomial time (i.e. $\beta = 1$ and $\gamma = (1 - \frac{1}{e})$), then we have a $(1 - \frac{1}{e})$ -approximation algorithm. For example, this is the case when for all $j \in D$ we have that demand d_j is constant, i.e. not part of the input.

We can also apply the above result to max-TPC(\emptyset, F^B, ∞), where the conflict graph is a complete bipartite graph $G = (R, B, E)$ (the Red–Blue transportation problem). To find the most profitable assignment for a single demand node with demand d_j we compute the most profitable assignment with supply nodes in R , the most profitable assignment with supply nodes in B and choose the most profitable assignment of the two. Hence, we can find a maximum weighted independent set of size at most d_j in polynomial time [i.e. $\beta = 1$ and $\gamma = (1 - \frac{1}{e})$]. From the theorem above it follows we have an $(1 - \frac{1}{e})$ -approximation algorithm, thereby improving the $1/2$ approximation factor found by Vancroonenburg et al. (2014).

Recall that one of the applications of the Red–Blue transportation problem is the assignment of male and female patients to hospital rooms, while keeping in mind that a room is either male-only or female-only. Suppose now, for example, that a male and female patient would like to share a room, because they are married. This gives rise to a TPC instance with a conflict graph which is bipartite, but not complete. As mentioned before, finding an Independent Set of size at most d_j of maximum weight in a bipartite graph is NP-hard and a greedy $1/2$ -approximation exists (Kalra et al. 2017) (i.e. $\beta = \frac{1}{2}$ and $\gamma = \frac{1}{3} - \epsilon$). Thus, Theorem 6 implies the existence of a $(\frac{1}{3} - \epsilon)$ approximation algorithm for max-TPC(\emptyset, F_j, ∞) with bipartite conflict graphs and $s_i = 1$ for all $i \in S$.

In case the supply nodes have a supply of $s_i \geq 1$ for all $i \in S$, and all edge capacities equal 1, we can use existing results for k -SAP to derive approximation results.

Theorem 7 *Suppose there is a β_j -approximation for finding a maximum weighted independent set of size at most d_j on the conflict graph G_{F_j} induced by F_j for each $j \in D$. Let k be the minimum of all supply, i.e. $k = \min_{i \in S} s_i$, and let $\beta = \min_{j \in D} \beta_j$.*

Then there exists a $\beta(1 - \frac{1}{e^k})$ -approximation algorithm for max-TPC($\emptyset, F_j, 1$).

4.1.2 Two-sided max-TPC

Recall that we require the conflicts to be one-sided in order to be able to use the results known for SAP. However, we can be slightly less restrictive. Assume that the demand nodes are coloured either red or blue and that each supply node may send flow to either only red demand nodes or only blue demand nodes. One approach is to solve the problem once while

ignoring the blue demand nodes and once while ignoring the red demand nodes, and select the best of both solutions; this is a $1/2 \cdot \gamma$ -approximation algorithm.

More generally, suppose we have two-sided conflicts, but one of the sides has identical conflicts, w.l.o.g. conflict set $C_i = C$ for all supply nodes $i \in S$. If we have a valid q -colouring on the induced conflict graph G_C , then we can split the problem into q one-sided max-TPC problems, which we can solve independently. A valid q -colouring on a graph $G = (V, E)$, is a function $c : V \rightarrow \{1, \dots, q\}$, such that for all $(v, u) \in E$ we have $c(v) \neq c(u)$.

Note that, to find a q -colouring, it need not be sufficient to know that a graph is q -colourable, since it is NP-hard to color a q -colourable graph with $q^{(\log q)/25}$ colours, for a sufficiently large constant q (Khot 2001).

Theorem 8 *Suppose there is a β_j -approximation for finding a maximum weighted independent set of size at most d_j on the conflict graph G_{F_j} induced by F_j for each $j \in D$, and a valid q -colouring on the conflict graph induced by C , let $\beta = \min_{j \in D} \beta_j$ and let $\gamma = \max\{\beta(1 - \frac{1}{e}), (\frac{\beta}{\beta+1} - \epsilon)\}$.*

Then there exists a $1/q \cdot \gamma$ -approximation algorithm for max-TPC(C, F_j, ∞) with supply $s_i = 1$ for each supply node $i \in S$.

Again, we can strengthen these results if we allow supply nodes to have a supply $s_i \geq 1$ for $i \in S$, but restrict that each supply node is assigned at most once to each demand node, i.e. edge capacity $u_{i,j} = 1$. This allows us to use the results for k -SAP.

Theorem 9 *Suppose there is a β_j -approximation for finding a maximum weighted independent set of size at most d_j on the conflict graph G_{F_j} induced by F_j for each $j \in D$, and a valid q -colouring on the conflict graph induced by C . Let k be the minimum of all supply, i.e. $k = \min_{i \in S} s_i$, and let $\beta = \min_{j \in D} \beta_j$.*

Then there exists a $1/q \cdot \beta(1 - \frac{1}{e})$ -approximation algorithm for max-TPC($C, F_j, 1$).

5 Conclusion

In this paper we have introduced the transportation problem with conflicts (TPC), which is a natural generalization of the classical Transportation problem. TPC is able to model several applications that have restrictions on the set of suppliers supplying a demand node and vice versa. Since in general TPC is at least as difficult as maximum (Weighted) Independent Set, we are interested in establishing the complexity of TPC when the restrictions satisfy particular structures that we capture using a conflict graph.

In particular, we have looked at the structure of the conflict graph induced by the conflict sets. We show that one-sided identical TPC remains NP-hard even if its conflict graph is a bipartite graph, planar graph, interval graph or a simple path, thereby answering an open problem stated by Vancroonenburg et al. (2014).

Finally we use results known for the separable assignment problem, and MWIS- d , to find approximation algorithms for special cases of max-TPC.

Acknowledgements This research has been partially funded by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office.

References

- Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). *Network flows: Theory, algorithms, and applications*. Englewood Cliffs: Prentice Hall.
- Bar-Noy, A., & Kortsarz, G. (1998). Minimum color sum of bipartite graphs. *Journal of Algorithms*, 28(2), 339–365.
- Bender, M., Thielen, C., & Westphal, S. (2015). Packing items into several bins facilitates approximating the separable assignment problem. *Information Processing Letters*, 115(6–8), 570–575.
- Cao, B. (1992). Transportation problem with nonlinear side constraints a branch and bound approach. *Zeitschrift für Operations Research*, 36(2), 185–197.
- Cao, B., & Uebe, G. (1995). Solving transportation problems with nonlinear side constraints with tabu search. *Computers & Operations Research*, 22(6), 593–603.
- Chen, C., Zheng, L., Srinivasan, V., Thomo, A., Wu, K., & Sukow, A. (2016). Conflict-aware weighted bipartite B-matching and its application to E-commerce. *IEEE Transactions on Knowledge and Data Engineering*, 28(6), 1475–1488.
- Darmann, A., Pferschy, U., Schauer, J., & Woeginger, G. J. (2011). Paths, trees and matchings under disjunctive constraints. *Discrete Applied Mathematics. 8th Cologne/Twente Workshop on Graphs and Combinatorial Optimization (CTW 2009)*, 159(16), 1726 – 1735.
- Fleischer, L., Goemans, M. X., Mirrokni, V. S., & Sviridenko, M. (2011). Tight approximation algorithms for maximum separable assignment problems. *Mathematics of Operations Research*, 36(3), 416–431.
- Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. New York: W. H. Freeman & Co.
- Goossens, D., Polyakovskiy, S., Spieksma, F. C. R., & Woeginger, G. J. (2012). Between a rock and a hard place: The two-to-one assignment problem. *Mathematical Methods of Operations Research*, 76(2), 223–237.
- Goossens, D. R., & Spieksma, F. C. R. (2009). The transportation problem with exclusionary side constraints. *4OR*, 7(1), 51–60.
- Håstad, J. (1996). Clique is hard to approximate within $n^{1-\epsilon}$. *Proceedings of 37th Conference on Foundations of Computer Science*, pp. 627–636.
- Kalra, T., Mathew, R., Pal, S. P., & Pandey, V. (2017). Maximum weighted independent sets with a budget. In D. Gaur & N. Narayanaswamy (Eds.), *CALDAM 2017, LNCS 10156*, pp. 254–266. Cham, Springer.
- Khot, S. (2001). Improved inapproximability results for maxclique, chromatic number and approximate graph coloring. *Proceedings of 42nd IEEE Symposium on Foundations of Computer Science*, pp. 600–609.
- Marx, D. (2005). A short proof of the NP-completeness of minimum sum interval coloring. *Operations Research Letters*, 33(4), 382–384.
- Pferschy, U., & Schauer, J. (2013). The maximum flow problem with disjunctive constraints. *Journal of Combinatorial Optimization*, 26(1), 109–119.
- Ratner, D., & Warmuth, M. (1990). The $(n^2 - 1)$ -puzzle and related relocation problems. *Journal of Symbolic Computation*, 10(2), 111–137.
- Salavatipour, M. (2000). On sum coloring of graphs. Master's Thesis, Department of Computer Science, University of Toronto.
- Sun, M. (1998). A tabu search heuristic procedure for solving the transportation problem with exclusionary side constraints. *Journal of Heuristics*, 3(4), 305–326.
- Sun, M. (2002). The transportation problem with exclusionary side constraints and two branch-and-bound algorithms. *European Journal of Operational Research*, 140(3), 629–647.
- Syarif, A., & Gen, M. (2003). Solving exclusionary side constrained transportation problem by using a hybrid spanning tree-based genetic algorithm. *Journal of Intelligent Manufacturing*, 14(3–4), 389–399.
- Szkaliczki, T. (1999). Routing with minimum wire length in the dogleg-free manhattan model is NP-complete. *SIAM Journal on Computing*, 29(1), 274–287.
- Vancroonenburg, W., Della Croce, F., Goossens, D., & Spieksma, F. C. R. (2014). The red–blue transportation problem. *European Journal of Operational Research*, 237(3), 814–823.