J Comb Optim (2010) 19: 200–216
DOI 10.1007/s10878-008-9175-8

Complexity analysis for maximum flow problems with arc reversals

Steffen Rebennack · Ashwin Arulselvan · Lily Elefteriadou · Panos M. Pardalos

Published online: 26 July 2008

Abstract We provide a comprehensive study on network flow problems with arc reversal capabilities. The problem is to identify the arcs to be reversed in order to achieve a maximum flow from source(s) to sink(s). The problem finds its applications in emergency transportation management, where the lanes of a road network could be reversed to enable flow in the opposite direction. We study several network flow problems with the arc reversal capability and discuss their complexity. More specifically, we discuss the polynomial time algorithms for the maximum dynamic flow problem with arc reversal capability having a single source and a single sink, and for the maximum (static) flow problem. The presented algorithms are based on graph transformations and reductions to polynomially solvable flow problems. In addition, we show that the quickest transshipment problem with arc reversal capability and the problem of minimizing the total cost resulting from arc switching costs are $\mathcal{NP}$-*hard*.

This research work was funded by UF Center for Multimodal solutions for Congestion Mitigation (CMS).

S. Rebennack · A. Arulselvan · P.M. Pardalos (✉)
Department of Industrial & Systems Engineering, Center for Applied Optimization, University of Florida, Gainesville, FL 32611, USA
e-mail: pardalos@ufl.edu

S. Rebennack
e-mail: steffen@ufl.edu

A. Arulselvan
e-mail: ashwin@ufl.edu

L. Elefteriadou
Department of Civil & Coastal Engineering, Transportation Research Center, University of Florida, Gainesville, FL 32611, USA
e-mail: elefter@ce.ufl.edu

Keywords Contraflow problem · Dynamic contraflow problem · Complexity analysis

1 Introduction

We study contraflow network problems, wherein we maximize flow in a graph while permitting direction reversals of an arc, resulting in a capacity increase in the direction of switch. The applications are realized in an emergency situation, where people have to be 'evacuated' from a specific area; *i.e.* a football stadium after a game, a city expecting a flood or hurricane, a zone where an unexploded ordnance device has been found, or a region which has been attacked by terrorists. In most of these cases, the evacuees are expected to leave the area of risk, the source(s), towards a safer place, the sink(s). A flow towards the source is undesired during most of these scenarios and we do not expect the evacuees to go in this direction. As a direct consequence, all the arcs that are not a part of any path from the source node(s) to the sink(s) might be left unused. One can even encounter idle arcs during certain scenarios, such as managing a football event, wherein we do have some amount of flow towards the source. These idle arcs could be used to increase the efficiency of evacuation by reversing their directions. The scenarios involving in partial lane reversal capability could be captured with appropriate graph transformations. We discuss several scenarios that may arise during the reconfiguration, which includes permitting only a subset of arcs to be reversed, imposing a switching cost to the arcs involving in the reversals.

There are very few optimization techniques in the literature handling arc reversals. Kim and Shekhar (2005) proposed a simulated annealing procedure for this problem and provided empirical results. They also provide a sketch of the proof that the problem is $\mathcal{NP}$-*complete*. A tabu-based heuristic was proposed by Tuydes and Ziliaskopoulos (2006) for the problem. They focus their study on a specialized version, where they permit lane reversals with partial capacities. Hamza-Lup et al. (2004) proposed a heuristic for this contraflow problem. These techniques and their pitfalls were discussed in (Kim and Shekhar 2005). A few other studies in the literature that are not analytical in nature were also proposed. They rely on simulation-based methods and decision support tools (Theodoulou and Wolshon 2004; Williams et al. 2007).

In this paper, we provide a detailed study of the arc reversal (or contraflow) problems with respect to their computational complexity. The motivation is to introduce the problems formally to provide a basis for further research in this area. As the applications are mainly realized during emergency situations, the dynamic flow problems are of principal interest, but we study static cases as presuppositions and also for the sake of completeness of the study. In Sect. 2, we provide a brief background of the network flow problems and explain the terminology used in the rest of the paper. We then provide a discussion of static flow problems in Sect. 3. A polynomial time algorithm through a graph transformation is introduced for the static maximum flow problem with arc reversal capability. The result is evident and it is useful in Sect. 4.1 in showing that the dynamic maximum contraflow problem with single source and single sink is polynomially solvable. We show in Sect. 4.2 that the decision version of the multiple sources and multiple sinks version of the problem is $\mathcal{NP}$-*complete* through a reduction from 3-SATISFIABILITY (3SAT). In Sect. 4.2.2, we show that the problem becomes $\mathcal{NP}$*-complete* by having just two sources or sinks. In addition, we discuss the inability of the graph transformation that was employed earlier to provide feasible solutions. We finally show in Sect. 5 that the problem of finding the minimum total cost, incurred due to an arc switching cost, to identify the arcs to be reversed is $\mathcal{NP}$*-hard*, even in the static case.

2 Background

The basic terminologies and definitions that are predominantly used in the network flows literature and that are essential for the rest of the paper are explained in this section.

Definition 1 (Static feasible flow) Given is a graph $G = (V, A)$ with capacities $c_e \in \mathbb{Z}^+$ for all arcs $e \in A$. A static flow, characterized by the function $f : A \to \mathbb{R}^+$, with value v, from $s \in V$ to $t \in V$ is feasible, if

$$f_e \le c_e, \quad \forall e \in A \tag{1}$$

$$\sum_{(i,j)\in A} f_{i,j} - \sum_{(j,k)\in A} f_{j,k} = \begin{cases} v, & j = s, \\ 0, & \forall j \in V \backslash \{s \cup t\}, \\ -v, & j = t. \end{cases} \tag{2}$$

We call node s as the 'source', node t as the 'sink' and rest of the nodes as 'intermediate' or 'transhipment' nodes.

Equation (1) ensures that the flow f_e along each arc $e \in A$ meets the capacity constraints; as we assume all lower bounds on the flow to be 0. In (2), the net flow out of s is v and the net flow in t is $-v$. For all intermediate nodes, the net flow is 0 which is also referred to as flow conservation. The definition of a feasible flow generalizes in a natural way for the case of multiple sources and multiple sinks.

A sequence of distinct nodes $x_1, x_2, \dots, x_n$ of a graph $G = (V, A)$ is called a chain if $(x_i, x_{i+1}) \in A, \forall i = 1, \dots, n$. A chain is also referred to as a directed path. Let P be the set of all chains from s to t. We define another flow function, $h : P \to \mathbb{Z}^+$, in terms of the flow along the chains from s to t. A feasible flow f with value v could be decomposed into a set of chains, P, from s to t, such that

$$v = \sum_{i=1}^{|P|} h_i.$$

The process of obtaining flow along the chains this way is called as 'chain decomposition'. A more detailed account of these terminologies could be found in (Ahuja et al. 1993; Ford and Fulkerson 1962).

In a dynamic graph or network $G = (V, A)$ each arc is associated with a travel time, $t : A \to \mathbb{R}^+$, besides the capacity function. The graph expanded over T time periods, $G^T = (V^T, A^T)$, is obtained by replacing each node by T copies and having nodes v_i^l and $v_j^{l+t_{i,j}}$ connected in $G^T = (V^T, A^T)$ if v_i and v_j are connected in G, for all $l = 0, \dots, T - t_{i,j}$. This concept of a feasible flow can be directly adapted to the dynamic case by ensuring that both (1) and (2) are satisfied for all discrete time steps. Hence, a feasible dynamic flow is a feasible flow in the time expanded graph with the value equal to the sum of the net flows out of all the T copies of s. For more details about time expanded graphs refer, for instance, to (Ahuja et al. 1993, Chap. 19.6).

3 Maximum static contraflow problems

In this section, we provide a polynomial time algorithm solving the maximum contraflow problem in a static graph. The results presented in this section are very basic and straightforward. Nevertheless, we discuss them in detail as this helps us in developing the main results in Sect. 4.

Now, let us define the maximum flow problem with arc reversal capability.

Definition 2 (Maximum Contraflow (MCF))
Instance: Given a directed graph $G = (V, A)$ with source $s^+ \in V$, sink $s^- \in V$ and capacity $c_e \in \mathbb{Z}^+$ on each arc $e \in A$.
Question: What is the maximum flow from node s^+ to node s^- if the direction of the arcs can be reversed?

This problem is also called *maximum flow problem with arc reversal*. Consider now procedure P-MCF. In the first step, an auxiliary graph $\widetilde{G} = (V, \widetilde{A})$ is constructed. The transformation from the original graph G is obtained by summing the capacities of arcs (i, j) and (j, i). This allows us to reduce the MCF problem to the maximum flow problem on the transformed graph in step 2. Step 3 removes cycle flows in the transformed graph. This ensures that the constructed solution of the MCF problem in step 4 is well defined.

We have the prior knowledge that there exists an optimal flow to the maximum flow problem that does not have cycles. Thus, arcs on either direction will never be used in this flow for the maximum flow problem. This is the basic idea of procedure P-MCF that motivates the graph transformation given. This result is straightforward but we can realize its impact in Sect. 4.1.

Theorem 1 (Proof of correctness) *Procedure P-MCF solves the maximum flow problem with arc reversal for graph $G = (V, A)$ optimally.*

Proof The proof consists out of two steps. First, we show that any solution of the procedure P-MCF is feasible for $G = (V, A)$. Second, we show its optimality.

For feasibility, we only have to show that step 4 in the algorithm is well defined; *i.e.* not both arcs (i, j) and (j, i) have to be switched. However, this is ensured by step 3. The optimal solution after the flow decomposition results in a set of paths from source to sink and a set of cycles with positive flows. After the flow decomposition we could cancel the positive flows along all cycles and ensure that there is no flow

Procedure Maximum Contraflow (P-MCF)

1. Construct the transformed graph $\widetilde{G} = (V, \widetilde{A})$ where the arc set is defined as
$$(i, j) \in \widetilde{A}, \quad \text{if } (i, j) \in A \text{ or } (j, i) \in A.$$
The arc capacity function $\widetilde{c}$ is given by
$$\widetilde{c}_{i,j} := c_{i,j} + c_{j,i},$$
for all arcs $(i, j) \in \widetilde{A}$.
2. Solve the maximum flow problem on graph $\widetilde{G}$ with capacity $\widetilde{c}$.
3. Perform flow decomposition into path and cycle flows of the maximum flow resulting from step 2. Remove the cycle flows.
4. Arc $(j, i) \in A$ is reversed, if and only if the flow along arc (i, j) is greater than $c_{i,j}$, or if there is a non-negative flow along arc $(i, j) \notin A$ and the resulting flow is the maximum flow with arc reversal for graph $G = (V, A)$.

End procedure

along any cycle. This ensures that there is either a flow along arc (i, j) or (j, i), but never on both arcs. Hence, the resulting flow from step 4 is a feasible flow with arc reversal for graph $G = (V, A)$.

Now, we prove that the resulting flow is also optimal. Note that any optimal solution to the maximum flow problem with arc reversal on graph $G = (V, A)$ is also a feasible solution to the maximum flow problem on the transformed graph $\widetilde{G} = (V, \widetilde{A})$. As the amount of flow sent from s to t is not changed in steps 3 and 4, the resulting flow is an optimal solution to the maximum flow problem with arc reversal on graph $G = (V, A)$. □

The running time of procedure P-MCF is dominated by solving a maximum flow problem in step 2 and by the flow decomposition in step 3; as steps 1 and 4 can be done in $O(|A|)$. Let us denote the running time for solving the maximum flow problem by $S_1(|V|, |A|)$ and for the flow decomposition problem by $S_2(|V|, |A|)$. Then, the running time of procedure P-MCF is given by $O(S_1(|V|, |A|) + S_2(|V|, |A|))$. Using the highest-label preflow-push algorithm leads to $S_1(|V|, |A|) = O(|V|^2 \cdot \sqrt{A})$, (Cheriyan and Maheshwari 1989). The flow decomposition can be done, for instance, in $O(|V| \cdot |A|)$, (Ahuja et al. 1993). This proves the following theorem.

Theorem 2 (Running time) *Procedure P-MCF solves the maximum contraflow problem in strongly polynomial time.*

We are now able to extend the result above to the case of multiple sources and multiple sinks. This problem is also called *maximum transshipment contraflow (MTCF) problem*.

Corollary 1 *The static version of the maximum contraflow problem with multiple sources and multiple sinks is polynomially solvable.*

Corollary 1 can be realized through a simple reduction. Let S^+ and S^- be the set of sources and the set of sinks, respectively. Then, add a 'super-source' u^+ and a 'super-sink' v^- together with the arcs (u^+, s^+), for all $s^+ \in S$, with arc capacities equal their respective surplus and (s^-, v^-) for all $s^- \in S^-$ with their arc capacities equal their respective deficits. For more details, refer to (Ford and Fulkerson 1962).

Recognize that we basically showed in this section that the maximum contraflow problem is equivalent to a maximum flow problem on an undirected (modified) graph. This could be seen in the graph transformation provided in step 1 of procedure P-MCF with arcs having same capacities in either directions.

4 Maximum dynamic contraflow problems

In this section, we discuss the *maximum dynamic contraflow (MDCF) problem*. The maximum dynamic flow problem was studied by Ford and Fulkerson (1962), where they try to maximize the flow sent from source to sink within a given time horizon T. Unlike the static case, in the dynamic network flow problem the flow over an arc can be repeated over time. FORD and FULKERSON proved that this problem is equivalent to solving a minimum cost flow problem with the arc costs as travel times on the arcs. Then the optimal flow on the arcs from source to sink is decomposed into a set of paths or chains. These chains are then *temporally repeated* over time to obtain the required dynamic flow. In other words, there is always a temporally repeated chain flow that is equivalent to the maximum dynamic flow. A more formal definition is provided below (Ford and Fulkerson 1962, p. 147).

Definition 3 (Temporally repeated flow) Let P be the set of paths obtained from the chain decomposition of the optimal minimum cost flow. Then the maximum dynamic flow is given by

$$\sum_{i \in P} (T + 1 - t_i) h_i,$$

where h_i is the flow along the i^{th} path and t_i is the time taken to travel the i^{th} path.

In this section, we first study the single source and single sink dynamic flow problem having arc reversal capability. We provide an algorithm employing a similar kind of graph transformation as procedure P-MCF and discuss its proof of correctness together with its worst case running time analysis. This implies that the *quickest contraflow (QCF) problem* is also polynomially solvable. In the *quickest flow problem*, the time to send a given flow from source to sink is minimized. Burkard et al. (1993) gave a strongly polynomial time algorithm for this problem.

Hoppe (1995) studied the multiple sources and multiple sinks version of this problem—also called the *quickest transshipment problem*—where the time taken to send the supply at the sources to the sinks while satisfying their demands is minimized. In static network flows, the multiple sources and multiple sinks are handled by adding a 'super-source' and a 'super-sink'. Then they are connected to the sources and sinks respectively, see Corollary 1. However, this solution procedure is not applicable in a dynamic case anymore. For the same reason, the dynamic contraflow problem with multiple sources and multiple sinks is $\mathcal{NP}$*-complete*. We provide an example illustrating this together with a proof of its $\mathcal{NP}$*-completeness*.

4.1 Single source and single sink

Let us extend the MCF problem of Sect. 3 to the dynamic case.

Definition 4 (Maximum dynamic contraflow (MDCF))
Instance: Given a directed graph $G = (V, A)$ with source node $s^+ \in V$, sink node $s^- \in V$, capacity $c_e \in \mathbb{Z}^+$ and transmission time $t_e \in \mathbb{Z}^+$ on each arc $e \in A$ with $t_{i,j} = t_{j,i}$ if $(i, j), (j, i) \in A$, and an overall time horizon $T \in \mathbb{Z}^+$.
Question: Determine the maximum amount of flow that can be send in T units of time from source s^+ to sink s^-, if the direction of the arcs can be reversed at time 0.

Note: In this case, if we choose to switch an arc, it remains switched from time 0 to T. The case where we allow switching of arcs back and forth in time is trivial as the quickest transhipment contraflow problem, with this assumption, reduces to the quickest transhipment problem through the graph transformation suggested in procedure P-MDCF and hence is polynomially solvable.

Definition 4 states that in a MDCF problem, the graph is allowed to be asymmetric with respect to the arc capacities. However, whenever both directions of an arc are included in the graph, then the traveling time of these two arcs must be the same. This assumption implies that the switching of an arc only changes the capacities of the arcs but does not alter their traveling time.

The following theorem reveals the usefulness of temporally repeated flows in the context of single source and single sink network flow problems, (Ford and Fulkerson 1962, Theorem 9.1).

Theorem 3 *There is a temporally repeated dynamic flow that is maximal over all dynamic flows for T periods.*

The flow to be temporally repeated could then be determined by just solving a minimum cost flow problem. Let us denote its running time by $S_3(|V|, |A|)$. Using, for instance, the minimum mean cycle-canceling algorithm leads to a strongly polynomial running time of $O(|V|^2 \cdot |A|^3 \cdot \log(|V|))$, (Goldberg and Tarjan 1989).

Before we proceed to the next lemma, we need to know that utilizing the concept of time expanded graphs in a solution algorithm leads to a pseudo-polynomial running time. In this case, the running time depends on $|T|$, rather than $\log(|T|)$ which would then lead to a pseudo-polynomial running time. Nevertheless, we use the concept of time expanded graphs in Theorem 4.

Consider now procedure P-MDCF. We show in Theorem 4 that it solves the MDCF problem correctly. The main differences of procedure P-MCF and P-MDCF is given

Procedure Maximum Dynamic Contraflow (P-MDCF)

1. Construct the transformed graph $\widetilde{G}=(V,\widetilde{A})$ where the arc set is defined as

$$(i,j)\in\widetilde{A},\quad \text{if } (i,j)\in A \text{ or } (j,i)\in A.$$

The arc capacity function $\widetilde{c}$ is given by

$$\widetilde{c}_{i,j}:=c_{i,j}+c_{j,i}$$

and the traveling time is

$$\widetilde{t}_{i,j}\left(=\widetilde{t}_{j,i}\right):=\begin{cases} t_{i,j}, & \text{if } (i,j)\in A,\\ t_{j,i}, & \text{otherwise,}\end{cases}$$

for all arcs $(i,j)\in\widetilde{A}$.

2. Generate a dynamic, temporally repeated flow on graph $\widetilde{G}$ with capacity $\widetilde{c}$ and traveling time $\widetilde{t}$.
3. Perform flow decomposition into path and cycle flows of the flow resulting from step 2. Remove the cycle flows.
4. Arc $(j,i)\in A$ is reversed, if and only if the flow along arc (i,j) is greater than $c_{i,j}$, or if there is a non-negative flow along arc $(i,j)\notin A$ and the resulting flow is the maximum flow with arc reversal for graph $G=(V,A)$.

End procedure

in step 2. For the dynamic problem, we need temporally repeated flows. This ensures that only one of the arcs (i,j) or (j,i) is used in the flow. This enables us to use the same flipping rule for the arcs as in procedure P-MCF.

In order to show the correctness of procedure P-MDCF, we need the following lemma.

Lemma 1 *The maximum amount of flow in the single source and single sink maximum dynamic contraflow problem for graph $G=(V,A)$ is less than the optimal flow in the maximum contraflow problem for the corresponding time expanded graph $G^T=(V^T,A^T)$.*

Proof The result follows directly from the observation that every feasible flow to the maximum dynamic contraflow problem has an equivalent feasible flow to the maximum contraflow problem of the time expanded graph. □

Please note that Lemma 1 holds good for more than one source and one sink. However, in general, equality holds only for the case of a single source and a single sink, as we will see in the following theorem. We are now ready to prove the correctness of procedure P-MDCF.

Theorem 4 (Proof of correctness) *Procedure P-MDCF solves the maximum dynamic contraflow problem for graph $G = (V, A)$ optimally.*

Proof The concept of this proof is similar to the proof of Theorem 1. First, we prove that all the steps in procedure P-MDCF are well defined and result in a feasible solution. Second, we show optimality.

For feasibility, the proof follows directly from the fact that the constructed flows are temporally repeated and hence, there is only a flow in one direction of two nodes, and never in both directions at the same time as well as at different time periods. After canceling the flows along the cycles, we have flows either on arc (i, j) or on (j, i) but not on both. This ensures that the flow is less than the reversed capacities on all the arcs at all time units. This also ensures the feasibility. In other words, we now have established the fact

$$[G = (V, A)]_{MDCFopt} \geq [\widetilde{G} = (V, \widetilde{A})]_{MDFopt},$$

by the argument that every feasible flow of the dynamic flow problem in the transformed graph $\widetilde{G} = (V, \widetilde{A})$ is feasible to the maximum dynamic contraflow problem in the graph $G = (V, A)$. Our proof is complete if we show that

$$[G = (V, A)]_{MDCFopt} \leq [\widetilde{G} = (V, \widetilde{A})]_{MDFopt}.$$

To see this, first note that the maximum contraflow in graph $G^T = (V^T, A^T) \geq$ maximum dynamic contraflow in graph $G = (V, A)$, from Lemma 1. Hence we have,

$$[G = (V, A)]_{MDCFopt} \leq [G^T = (V^T, A^T)]_{MCFopt}.$$

By Theorem 1 we have that the maximum contraflow problem in graph $G^T = (V^T, A^T)$ is equivalent to the maximum flow problem in the graph $\widetilde{G}^T = (V^T, \widetilde{A}^T)$, where the arc set $\widetilde{A}^T$ is defined as

$$(i, j) \in \widetilde{A}^T, \quad \text{if } (i, j) \in A^T \text{ or } (j, i) \in A^T,$$

and the arc capacity function $\widetilde{c}$ is given by

$$\widetilde{c}^t_{i,j} := c^t_{i,j} + c^t_{j,i}.$$

Thus,

$$[G^T = (V^T, A^T)]_{MCFopt} = [\widetilde{G}^T = (V^T, \widetilde{A}^T)]_{MFopt}.$$

By Theorem 3, the maximum flow in the time expanded graph $\widetilde{G}^T = (V^T, \widetilde{A}^T)$ can be obtained by a temporally repeating a chain flow of a static graph $\widetilde{G} = (V, \widetilde{A})$. Hence we have the fact,

$$[\widetilde{G}^T = (V^T, \widetilde{A}^T)]_{MFopt} = [\widetilde{G} = (V, \widetilde{A})]_{MDFopt}. \qquad \square$$

Just like procedure P-MCF, running time dominating are steps 2 and 3 for procedure P-MDCF. This results in a worst case running time of $O(S_2(|V|, |A|) + S_3(|V|, |A|))$; which is strongly polynomial.

Theorem 5 (Running time) *Procedure P-MDCF solves the maximum flow problem in strongly polynomial time.*

For given excess b, the *quickest contraflow problem* determines the minimum time horizon T needed by any feasible flow.

Corollary 2 *The quickest contraflow problem can be solved in a strongly polynomial time.*

One way to realize Corollary 2 is through the work by Burkard et al. for the quickest flow problem, (Burkard et al. 1993). First, obtain an upper bound on the quickest time and second, perform a binary search by repeatedly solving the minimum dynamic contraflow problem. Such a bound can be obtained in polynomial time, for instance, by computing a path from source to sink and temporally repeating flow along the path until all supply at the source is sent to the sink. However, this leads to a weakly polynomial algorithm. A strongly polynomial algorithm could be obtained through a parametric search suggested by Megiddo (1979), Burkard et al. (1993).

4.2 Multiple sources and multiple sinks

Let us start with the definition of the multiple sources and multiple sinks version of the MDCF problem.

Definition 5 (Dynamic Transshipment Contraflow (DTCF))
Instance: A directed graph $G = (V, A)$, a set of sources $S^+ \subset V$, a set of sinks $S^- \subset V$, arc capacities $c_e \in \mathbb{Z}^+$ and transmission time $t_e \in \mathbb{Z}^+$ for each arc $e \in A$ with $t_{i,j} = t_{j,i}$ if $(i, j), (j, i) \in A$, and an overall positive integer time bound T.
Question: Is there a feasible dynamic flow within time horizon T, allowing each arc to be revered once at time 0?

Note that the DTCF problem is a decision problem corresponding to the maximum dynamic contraflow problem with multiple sources and multiple sinks.

4.2.1 DTCF is $\mathcal{NP}$-complete in the strong sense

In this section, we proof that the DTCF problem is $\mathcal{NP}$-*complete*. A sketch of the proof outline was given in (Kim and Shekhar 2005). However, we provide a rigorous proof. Also, the proof has some differences though we provide the reduction from the same problem, *3SAT*, (Garey and Johnson 1979, p. 46):

Definition 6 (3SAT)
Instance: Collection $C = \{c_1, c_2, \ldots, c_m\}$ of clauses on a finite set U of variables such that $|c_1| = 3$ for $1 \leq i \leq m$.
Question: Is there a truth assignment for U that satisfies all the clauses in C?

3SAT is known to be $\mathcal{NP}$-*complete* in the strong sense, see (Garey and Johnson 1979, Theorem 3.1).

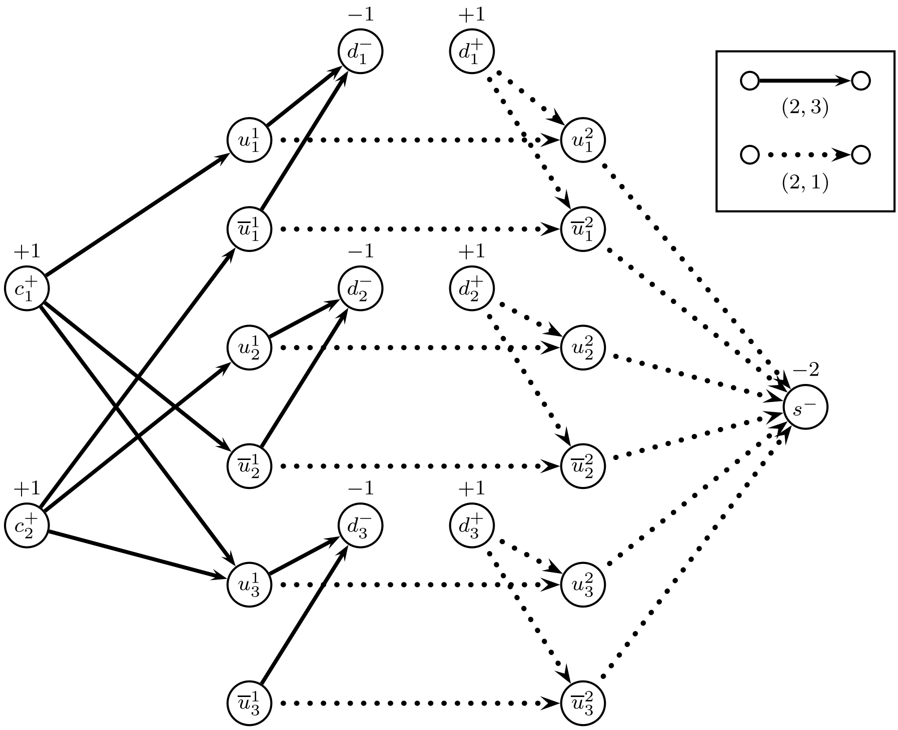


Fig. 1 Transformed graph G_{3SAT} corresponding to 3SAT instance $C = \{\{u_1, \overline{u}_2, u_3\}, \{\overline{u}_1, u_2, u_3\}\}$

For an instance of 3SAT, construct a graph $G_{3SAT} = (V, A)$ for DTCF as follows. For each clause c_i we have one source node c_i^+ with a surplus of 1. Each variable $u_j \in U$, is presented by six nodes in the graph: two for each literal, named u_j^1, u_j^2, $\overline{u}_j^1$ and $\overline{u}_j^1$ respectively, one source node with surplus 1, d_j^+, and one sink node with deficit -1, d_j^-. Finally, there is one node with deficit $-|C|$, named s^-. This sums up to $|V| = |C| + 6|U| + 1$ nodes. Each clause node c_i^+ is connected to the nodes with superscript 1 representing its literals, taking 3 time units. For each j, the node u_j^1 is connected to its copy, u_j^2, with transshipment time of 1. Nodes d_j^+ are connected to u_j^2 and $\overline{u}_j^2$ with transshipment time 1, while nodes d_j^- are connected to u_j^1 and $\overline{u}_j^1$ having a transshipment time of 1. Finally, each second copy (superscript 2) of the literals is connected to the sink s^- taking a time of 1. All arcs have a capacity of $|C|$. This leads to $|A| = 3|C| + 8|U|$ arcs in graph G_{3SAT}. One such graph transformation is shown in Fig. 1.

The proof of the validity of the transformation is based on the following key observation.

Lemma 2 *In any feasible flow f in the graph G_{3SAT} within time $T = 5$, there is a flow of value* 1 *from node d_j^+ to node d_j^-, for all j.*

Proof Let us fix index j and assume that the flow to node d_j^+ is integral. If the flow to node d_j^- does not come from node d_j^+, then it can only come from exactly one of the nodes c_i or d_k^+ with $k \neq j$. However, in both cases, the flow arrives at node d_j^- earliest at time 6, or 7 respectively. This proofs the lemma for the case of integer flows. The case of fractional flow is similar: If some fraction of the flow to node d_j^- comes from a different node then d_j^+, then the flow arrives after time $T = 5$. □

Lemma 2 implies that for a feasible flow, at least one of the arcs (u_j^1, u_j^2) or $(\overline{u}_j^1, \overline{u}_j^2)$ has been switched for all j – with other words, at most one of the two arcs (u_j^1, u_j^2) and $(\overline{u}_j^1, \overline{u}_j^2)$ keep their direction in any feasible flow with time bound $T = 5$. Now, we are able to proof the following lemma.

Lemma 3 *An instance of* 3*SAT is a 'YES' instance, if and only if the transformed graph* G_{3SAT} *is a 'YES' instance for DTCF with overall time bound* $T = 5$.

Proof "⇒" Let 3SAT have the feasible assignment $u_j = a_j$ for all variables, with $a_j \in \{0, 1\}$. Then, reverse the arcs (u_j^1, u_j^2) if $a_j = 0$, and reverse arc $(\overline{u}_j^1, \overline{u}_j^2)$ otherwise. Now, for all j, send one unit of flow from d_j^+ to d_j^- along the reversed arc. As only one of the arcs (u_j^1, u_j^2) or $(\overline{u}_j^1, \overline{u}_j^2)$ has been switched, we can send flow from any of the nodes c_i^+ through any non-switched arc, dependent on the assignment of the literals. This leads to a feasible flow for DTCF within time $T = 5$.

"⇐" We have to show, that any feasible flow f for DTCF needing (at most) 5 units of time leads to a 'YES' instance of the 3SAT. We assign the following value to each variable $u_j \in U$ as

$$u_j := \begin{cases} 0, & \text{if arc } (u_j^1, u_j^2) \text{ is reversed in flow } f, \\ 1, & \text{otherwise.} \end{cases} \tag{3}$$

We have to show that this is a satisfying truth assignment for the 3SAT instance. Now, assume that clause c_i is not a truth assignment. One unit of flow is send from node c_i^+ to node s^- through one of the nodes u_j^1 or $\overline{u}_j^1$ with $u_j \in c_i$ or $\overline{u}_j \in c_i$. Notice that this flow cannot go through any other node c_k^+ with $1 \leq k \leq m$ and $k \neq i$. Lemma 2 implies that the corresponding value of variable u_j has been set; *i.e.* $u_j = 1$ if the flow passes node u_j^1, or $\overline{u}_j = 1$ if it passes through node $\overline{u}_j = 1$. This leads to a contradiction. □

The second part of the proof of Lemma 3 together with Lemma 2 give the idea of the transformation from *3SAT*. First, we have to send one unit of flow from each of the nodes d_j^+ to d_j^-. This ensures that (at least) one of the arcs between the copies of the literals has to be reversed. The arc which has not been switched can then be used for the flow of the nodes c_j^+, allowing the clauses to have a truth assignment. Hence, the value of the literals is reflected by the switching of the arcs.

Theorem 6 *DTCF is* $\mathcal{NP}$*-complete in the strong sense.*

Proof DTCF $\in \mathcal{NP}$, as a non-deterministic algorithm needs only guess the set of arcs to be reversed together with a flow f and check if the flow is feasible with time bound $T = 5$; which can all be done in polynomial time. Lemma 3 states that the given transformation G_{3SAT} from *3SAT* to DTCF is valid. As the cardinality of the node set and the arc set of the constructed graph is $O(|C|)$, the transformation is polynomial in the input size of *3SAT*. □

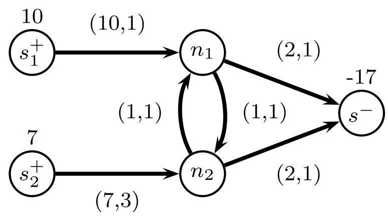
We want to mention that the transformation from *3SAT* can easily be changed to the general SAT by only changing the appropriate arcs from the clause nodes to the nodes representing the literals.

4.2.2 *What makes DTCF so tough to solve?*

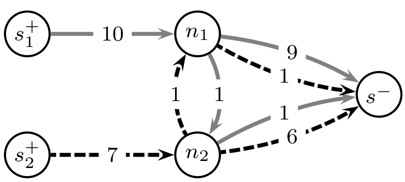
Ford and Fulkerson introduced the idea of temporally repeated chain flows of a static flow. This enabled them to solve the maximum dynamic flow problem with one source and one sink. The fundamental principle is that there is always an optimal dynamic flow which uses only one direction of an arc, but never both. They call this a *standard chain decomposition*. This property allows us to solve the maximum dynamic flow problem in strongly polynomial time. We exploit this property in Sect. 4.1 to solve the MDCF problem.

The concept of standard chain decomposition is not sufficient for some well known dynamic flow problems (Hoppe 1995; Hajek and Ogier 1984; Orlin 1983). An example is given in Fig. 2. Graph $G = (V, A)$ shown in Fig. 2(a) has a feasible flow with time $T = 6$ as illustrated in Fig. 2(b). The dashed and gray lines show the two flows from nodes s_1^+ and s_2^+ to node s^-, respectively. Analyzing the graph reveals that there is no feasible flow within time horizon $T = 6$ using only one of the arcs (n_1, n_2) or (n_2, n_1); this can be seen, for instance, by considering the flow trough the cut separating s^- from the rest of the graph.

However, it was still possible to solve the maximum dynamic flow problem with multiple sources and multiple sinks in the time-expanded graph; resulting in a pseudo-polynomial running time algorithm. However, Hoppe was able to provide a polynomial time algorithm for the dynamic transshipment problem (Hoppe 1995). He introduced the concept of *non-standard chain decomposition*, allowing flow in either



(a) Graph $G = (V, A)$



(b) Feasible flow with $T = 6$ using both arcs (n_1, n_2) and (n_2, n_1)

Fig. 2 A tough instance of DTCF

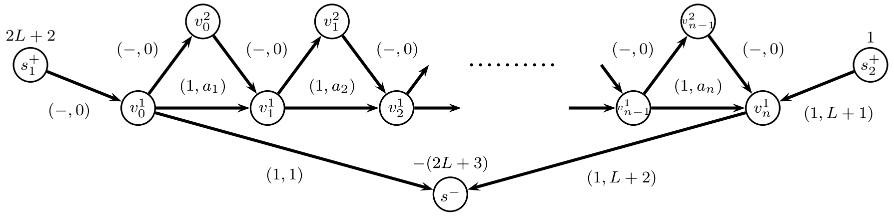


Fig. 3 Instance for DTCF with time bound $T = 2L + 2$ resulting from PARTITION

directions of an arc at different time steps – if both directions of an arc are present in the graph.

Loosely speaking, the procedures P-MCF and P-MDCF reverse the arcs on the fly and they are blind whether they reverse an arc or not. This does not cause any problems in the context of static flows or single source and single sink dynamic flows, as in a standard chain decomposition, one can always derive an optimal solution using only one the arcs during the whole time horizon. However, in the case of multiple sources and multiple sinks, the potential of using both arcs leads to the problem that we have to know if an arc has been reversed or not. But exactly this memory and the tradeoff of reversing the arc now or at a later time makes the problem $\mathcal{NP}$-*complete*. Consider Fig. 2 again. Applying the idea of procedures P-MDCF to this problem leads to the following result: At time 1, we would switch arc (n_2, n_1) in order to increase the capacity and at time point 3, we would switch it back again; resulting in a flow needing only $T = 5$ time steps.

In Sect. 4.2.1, we showed that DTCF is $\mathcal{NP}$-*complete*. The reduction from *3SAT* involves $|C| + |U|$ source nodes and $|U| + 1$ sink nodes. In the following, we show that there is no polynomial time algorithm for the DTCF problem having only two sources and one sink (or one source and two sinks), unless $\mathcal{P} = \mathcal{NP}$. In other words, allowing only one more source or sink to DTCF makes the problem $\mathcal{NP}$-*complete*.

We do not go into full detail here, but rather provide the idea of a reduction from PARTITION, which is motivated by the key observation of Lemma 2 and the $\mathcal{NP}$-*completeness* proof by Melkonian (2007). Given is a finite set A and a size $a_i \in \mathbb{Z}^+$ for each $i \in A$. The PARTITION problem decides whether there is a subset $A' \subseteq A$ such that $\sum_{i \in A'} a_i = \sum_{j \in A' \setminus A} a_j$, or not. PARTITION is known to be $\mathcal{NP}$-*complete* (in the weak sense), see (Garey and Johnson 1979, Theorem 3.5, Chapter 4.2). Let $\sum_{i \in A} a_i = 2L$ with $L \in \mathbb{Z}^+$. We construct an instance of the DTCF with two source nodes s_1^+, s_2^+ and one sink node s^-, as shown in Fig. 3. The idea of this transformation is that the flow at node s_2^+ has to pass through node v_0^1 to reach node s^-, and one unit of flow from node s_1^+ has to travel though node v_n^1 to node s^-. This is indeed true as otherwise the total time bound of $T = 2L + 2$ would be exceeded. The flow through the nodes v_0^1 to v_n^1 and back gives the assignment to set A'; *i.e.* $i \in A'$ if and only if arc $(v_{i_1}^1, v_i^1)$ is not reversed in the graph.

5 Contraflow problems with arc switching Cost

To allow the switching of an arc in order to increase the capacity in one direction results from the application in evacuation scenarios. However, in practice, you might not be able to switch certain arcs. For instance, in evacuation scenarios, certain streets are reserved for emergency vehicles but can also be used by (a limited number of) other travelers; *i.e.* this can be modeled by reducing the capacity of this arc and blocking it from being reversed. In addition, the switching of an arc is highly costly; *i.e.* in order to switch the direction of a highway, we have to set up police blocks on each entry to the highway. Hence, it is natural to ask what are the minimum cost incurred in switching the arcs allowing a certain (minimum) amount of flow. This leads to the following problem.

Definition 7 (Fixed switching cost contraflow (FSCF))
Instance: A directed graph $G = (V, A)$ with a set of sources S^+, a set of sinks S^-, excess $b \in \mathbb{Z}^{|V|}$, arc capacities c_e and arc-switching cost b_e^f for each arc $e \in A$.
Question: Find a feasible flow f in G with minimal total cost, if the direction of the arcs can be reversed with (fixed) cost b^f.

Note that FSCF is a static problem with multiple sources and multiple sinks. The fixed cost b_e^f occur, whenever arc e is reversed. This definition allows to model the situation described above: Whenever an arc cannot be reversed, then its cost can be assigned a high value; *i.e.* Big M. As the cost of switching can differ for each arc, we can distinguish between the effort of reversing an arc; *i.e.* reversing a highway or an alleyway involves different cost or resources.

The fixed switching-cost contraflow problem has the following interesting value. One can solve the MTCF problem and determine the optimal flow in the graph, see Corollary 1. Later, one can apply the FSCF problem to determine the minimal cost implied by the switching of arcs, while still pushing the optimal amount of flow through the graph.

Notice that the FSCF problem has a similar structure as the *minimum concave-cost network flow* problems. These problems ask to find a feasible flow while minimizing the total cost which are in this case the sum of concave-costs induced by using of the arcs. For an exact definition and an overview about this problem, please see the survey by Guisewite and Pardalos (1990). We can basically assume the concave-cost per arc to consist of fixed cost, occurring whenever this particular arc is used, and a variable cost, depending on how much flow is send trough this arc, see (Kim and Pardalos 2000). Fixing the variable cost to zero leads to a special problem called *minimum cost fixed flow* (MCFF) problem. Krumke et al. prove that this problem is $\mathcal{NP}$*-hard* in the strong sense even on series-parallel graphs, (Krumke et al. 1998, Theorem 14). Series-parallel graphs have a very special structure and are defined recursively, see (Gross and Yellen 2003; Bern et al. 1987). Furthermore, Krumke et al. show that the minimum cost fixed flow problem is equivalent to the following problem, (Krumke et al. 1998, Theorem 8):

Definition 8 0/1-minimum improvement flow (MIF)
Instance: A graph $G = (V, A)$ with sink node s^+, source node s^-, excess $b \in \mathbb{Z}^{|V|}$,

arc capacities $c_e \in \mathbb{Z}^+$, maximum capacities $C_e \in \mathbb{Z}^+$, $C_e \geq c_e$ and capacity improvement cost $b'_e \in \mathbb{Z}$.
Question: Determine an improvement strategy $d : A \to \{0, C_e - c_e\}$ with minimum cost $\sum_{e\in A} d_e b'_e$, such that the graph with the improved capacity $u_e + d_e, \forall e \in A$, allows a feasible flow f from s^+ to s^-.

The definition given here is slightly different then the one in the paper by Krumke et al. (1998, Definition 7). Basically, we assume all data to be positive integral. The improvement strategy function d is a 0-1 decision if additional capacity is used or not; independent of how much additional capacity is used. The cost for this additional capacity for arc e is fix at value $(C_e - c_e)b'_e$. In order to prove that FSCF is strongly $\mathcal{NP}$*-hard*, we show that it is equivalent to MIF.

Theorem 7 *Fixed switching-cost contraflow is equivalent to* 0/1*-minimum improvement flow.*

Proof Without loss of generality, we can assume the FSCF problem to have single source and single sink. Recognize that the graph transformation provided for Corollary 1 works here.

"⇒" Given an instance of FSCF for graph $G = (V, A)$ with arc capacity c_e and arc-switching cost b^f_e. Construct an instance of MIF for graph $\overline{G} = (\overline{V}, \overline{A})$ as follows. If there is an arc $(i, j) \in A$ and $(j, i) \notin A$, then $(i, j), (j, i) \in \overline{A}$ with $\overline{c}_{i,j} = \overline{C}_{i,j} = \overline{C}_{j,i} := c_{i,j}$, $\overline{b}'_{i,j} = \overline{c}_{j,i} := 0$, and $\overline{b}'_{j,i} := b^f_{i,j}/c_{i,j}$ respectively. For the case that $(i, j), (j, i) \in A$, we define $(i, j), (j, i) \in \overline{A}$ with $\overline{c}_{i,j} := c_{i,j}$, $\overline{C}_{i,j} = \overline{C}_{j,i} := c_{i,j} + c_{j,i}$, $\overline{b}'_{i,j} := b^f_{j,i}/(c_{i,j} + c_{j,i})$, $\overline{c}_{j,i} := c_{j,i}$, and $\overline{b}'_{j,i} := b^f_{i,j}/(c_{i,j} + c_{j,i})$ respectively. By applying the cycle reduction principle used in Sect. 3 and 4.1, we can see that this transformation is indeed valid.

"⇐" Given an instance of MIF for graph $G = (V, A)$ with c_e, C_e and b'_e, construct an instance of FSCF for graph $\overline{G} = (\overline{V}, \overline{A})$ as follows. For any arc $(i, j) \in A$, we have the three arcs $(i, j), (i, i'), (j, i') \in \overline{A}$. Define $\overline{c}_{i,j} := c_{i,j}$, $\overline{b}^f_{i,j} = \overline{b}^f_{i,i'} := M$, $\overline{c}_{i,i'} = \overline{c}_{j,i'} = C_{i,j} - c_{i,j}$ and $\overline{b}^f_{j,i'} := b'_{i,j}(C_{i,j} - c_{i,j})$, where M is a big number preventing to switch the corresponding arc in an optimal solution. □

Recognize that having fixed cost for arc reversals makes the problem $\mathcal{NP}$*-hard*, even in the static case. One reason is, for instance, the previously mentioned observation, that the procedure P-MCF is 'blind' for the arc reversal decisions. Adding a time component to FSCF makes it practically even more difficult to solve. The time component reveals also the differences between the (dynamic) fixed switching-cost contraflow problem and the (dynamic) 0/1-minimum improvement flow problem: MIF affects only a particular arc (i, j), while in FSCF also the reverse arc (j, i) is affected, if both arcs are contained in the graph.

6 Conclusions

This paper formally introduces the contraflow problem that has applications in emergency transportation management. Several classic network flow problems are studied, including static and dynamic networks. A polynomial time algorithm for the dynamic contraflow problem with single source and single sink is given, together with an $\mathcal{NP}$*-completeness* proof for the dynamic transhipment contraflow problem. The hardness of the contraflow problem with arc reversal cost is also indicated.

Acknowledgements The authors thank Stanislav Busygin and Balachandran Vaidyanathan for their careful proof reading and various helpful suggestions.

References

Ahuja RK, Magnati TL, Orlin JB (1993) Network flows: theory, algorithms, and applications. Prentice Hall, Englewood Cliffs

Bern MW, Lawler EL, Wong AL (1987) Linear-time computation of optimal subgraph of decomposable graphs. J Algorithms 8:216–235

Burkard R, Dlaska K, Klinz B (1993) The quickest flow problem. Math Methods Oper Res 37(1):31–58

Cheriyan J, Maheshwari SN (1989) Analysis of preflow push algorithm for maximum network flow. SIAM J Comput 18:1057–1086

Ford FR, Fulkerson DR (1962) Flows in networks. Princeton University Press, Princeton

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

Goldberg AV, Tarjan RE (1989) Finding minimum-cost circulations by cancelling negative cycles. J ACM 36:873–886

Gross JL, Yellen J (2003) Handbook of graph theory. Discrete mathematics and its applications. CRC, New York

Guisewite GM, Pardalos PM (1990) Minimum concave-cost network flow problems: applications, complexity, and algorithms. Ann Oper Res 25:75–100

Hajek B, Ogier RG (1984) Optimal dynamic routing in communication networks with continuous traffic. Networks 14:457–487

Hamza-Lup GL, Hua K, Lee M, Peng R (2004) Enhancing intelligent transportation systems to improve and support homeland security. In: Proceedings of the 7th international IEEE conference on intelligent transportation systems, pp 250–255

Hoppe BE (1995) Efficient dynamic network flow algorithms. PhD thesis, Cornell University. http://www.math.tu-berlin.de/~skutella/hoppe_thesis.ps.gz

Kim D, Pardalos PM (2000) Dynamic slope scaling and trust interval techniques for solving concave piecewise linear network flow problems. Networks 35(3):216–222

Kim S, Shekhar S (2005) Contraflow network reconfiguration for evaluation planning: a summary of results. In: Proceedings of the 13th annual ACM international workshop on geographic information systems, pp 250–259

Krumke S, Noltemeier H, Schwarz S, Wirth H, Ravi R (1998) Flow improvement and network flows with, fixed costs. In: Proceedings of the international conference of operations research (0R'98), Zürich, pp 158–167

Megiddo N (1979) Combinatorial optimization with rational objective functions. Math Oper Res 4:414–424

Melkonian V (2007) Flows in dynamic networks with aggregate arc capacities. Inf Process Lett 101(1):30–35

Orlin JB (1983) Maximum-throughput dynamic network flows. Math Program 27:214–231

Theodoulou G, Wolshon B (2004) Alternative methods to increase the effectiveness of freeway contraflow evacuation. J Transp Res Board 1865:48–56

Tuydes H, Ziliaskopoulos A (2006) Tabu-based heuristic approach for optimization of network evacuation contraflow. Transp Res Record 1964:157–168

Williams B, Tagliaferri A, Meinhold S, Hummer J, Rouphail N (2007) Simulation and analysis of freeway lane reversal for coastal hurricane evacuation. J Urban Plng Devel 133(1):61–72