Discrete Applied Mathematics 158 (2010) 1441–1455

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Network flow interdiction on planar graphs

R. Zenklusen *

Institute for Operations Research, ETH Zurich, Switzerland

A R T I C L E I N F O

Article history:
Received 22 July 2008
Received in revised form 15 December 2009
Accepted 12 April 2010
Available online 14 May 2010

Keywords:
Network flow interdiction
Network robustness
Planar graphs
Planar duality
Pseudo-polynomial algorithms

A B S T R A C T

The network flow interdiction problem asks to reduce the value of a maximum flow in a given network as much as possible by removing arcs and vertices of the network constrained to a fixed budget. Although the network flow interdiction problem is strongly NP-complete on general networks, pseudo-polynomial algorithms were found for planar networks with a single source and a single sink and without the possibility to remove vertices. In this work, we introduce pseudo-polynomial algorithms that overcome various restrictions of previous methods. In particular, we propose a planarity-preserving transformation that enables incorporation of vertex removals and vertex capacities in pseudo-polynomial interdiction algorithms for planar graphs. Additionally, a new approach is introduced that allows us to determine in pseudo-polynomial time the minimum interdiction budget needed to remove arcs and vertices of a given network such that the demands of the sink node cannot be completely satisfied anymore. The algorithm works on planar networks with multiple sources and sinks satisfying that the sum of the supplies at the sources equals the sum of the demands at the sinks. A simple extension of the proposed method allows us to broaden its applicability to solve network flow interdiction problems on planar networks with a single source and sink having no restrictions on the demand and supply. The proposed method can therefore solve a wider class of flow interdiction problems in pseudo-polynomial time than previous pseudo-polynomial algorithms and is the first pseudo-polynomial algorithm that can solve non-trivial planar flow interdiction problems with multiple sources and sinks. Furthermore, we show that the k-densest subgraph problem on planar graphs can be reduced to a network flow interdiction problem on a planar graph with multiple sources and sinks and polynomially bounded input numbers.

1. Introduction

In this paper, we are interested in minimizing the maximum flow of a network by removing arcs and vertices constrained to some interdiction budget. This problem is mainly known as *network interdiction* or *network flow interdiction*; sometimes the term *network inhibition* is used. One can either allow or disallow partial removal of arcs (removing half of an arc corresponds to reduce its capacity to half of the original value). However, the techniques and results do not substantially differ on this issue. We are interested in the case without partial arc removal. The problem of finding the *k most vital arcs* of a flow network is a special case of the network flow interdiction problem where k arcs have to be removed such that the maximum flow is reduced as much as possible. An important class of problems closely related to network interdiction is *stochastic network interdiction*. These are interdiction problems where one or more of the components of the network interdiction problem are not known with certainty. See [4,9] for more information on stochastic network interdiction.

* Fax: +41 44 632 1025.
E-mail address: rico.zenklusen@ifor.math.ethz.ch.

0166-218X/$ – see front matter
doi:10.1016/j.dam.2010.04.008

Network interdiction and related problems appear in various areas such as drug interdiction [20], military planning [8], protecting electric power grids against terrorist attacks [17] and hospital infection control [2]. The network flow interdiction problem was shown to be strongly NP-complete on general graphs and weakly NP-complete when restricted to planar graphs [15,20]. Different algorithms for finding exact solutions were proposed [8,13,16,20], which are mainly based on branch and bound procedures. In [3] a pseudo-approximation was presented. Earlier work includes [19].

When dealing with planar graphs with a single source and sink it was shown that by using planar duality, pseudo-polynomial algorithms for the network flow interdiction problem can be constructed when only arc removals are allowed [15]. Two of the major drawbacks of these algorithms (apart from the fact that they can only be applied on planar graphs) are the restrictions that vertex removals are not allowed and that the network must have exactly one source and one sink. Vertex removal can easily be formulated as arc removal by a standard technique of doubling vertices, and multiple sources and sinks are generally handled by the introduction of a supersource and supersink [1,6]. Unfortunately, these transformations destroy planarity and make it impossible to profit from the currently known specialized interdiction algorithms for planar graphs.

In this work, we are interested in the development of pseudo-polynomial algorithms for planar graphs that overcome various restrictions of previous methods. We propose a planarity-preserving transformation that enables incorporation of vertex removals and vertex capacities in pseudo-polynomial interdiction algorithms for planar graphs. We hereby answer a question raised in [15] asking how vertex capacities can be handled. The proposed algorithm can easily be transformed into a fully polynomial approximation scheme (FPAS) by using the rounding and scaling technique presented in [21].

Additionally, a pseudo-polynomial algorithm is introduced for the problem of determining the minimum interdiction budget needed to make it impossible to satisfy the demand of all sink nodes. The algorithm works on planar networks with multiple sources and sinks satisfying that the sum of the supplies at the sources equals the sum of the demands at the sinks. This problem is a generalization of the problem of determining whether a flow network is $n-k$ secure, i.e., any removal of k of its components does not impact the value of the maximum flow. A simple adaption of the proposed method allows us to broaden its applicability to solve interdiction problems on planar networks with a single source and sink without restriction on the demand and supply. The proposed method can therefore solve a wider class of interdiction problems in pseudo-polynomial time than previous pseudo-polynomial algorithms and is the first pseudo-polynomial algorithm that can solve non-trivial planar interdiction problems with multiple sources and sinks.

It is not known whether network flow interdiction on planar networks with multiple sources and sinks is a strongly NP-complete problem. To link the planar network flow interdiction problem with multiple sources and sinks to a more classical combinatorial problem we show that the k-densest subgraph problem on planar graphs can be reduced to a planar network flow interdiction problem with polynomially bounded numbers as input. However, it is not known if either of these problems can be solved in polynomial time.

The paper is organized as follows. We begin by giving some definitions and notations in Section 2. In Section 3, we give an overview of known complexity results on network flow interdiction and show how the k-densest subgraph problem on planar graphs can be reduced to a planar network flow interdiction problem with small input numbers. Section 4 presents an extension of currently known algorithms for network flow interdiction problems on undirected networks were only arc removals are allowed to the case of directed networks. We present in Section 5 a pseudo-polynomial algorithm for network flow interdiction on planar networks with a single source and sink that can handle vertex interdiction and vertex capacities. In Section 6 a pseudo-polynomial algorithm is presented that can be used for solving some network flow interdiction problems with multiple sources and sinks. Furthermore we show how the previously presented technique for modelling vertex interdiction and vertex capacities can be adapted to be used in the proposed algorithm for problems with multiple sources and sinks.

2. Preliminaries

2.1. Definitions and notations

Let (V, E) be a directed graph where V is the set of vertices, E is the set of arcs and for every arc $e \in E$, $u(e) \in \{0, 1, \ldots\}$ denotes its capacity. Two special nodes $s, t \in V, s \neq t$ designate the source node and sink node, respectively (the generalization to multiple sources and sinks is straightforward). We call the network $G = (V, E, u, s, t)$ a *flow network*. For $V', V'' \subseteq V$ we denote by (V', V'') the set of all arcs from V' to V''. Furthermore, for $V' \subseteq V$ we denote by $\omega^+(V')$ and $\omega^-(V')$ the set of all arcs exiting V' and entering V', respectively, i.e., $\omega^+(V') = (V', V \setminus V')$ and $\omega^-(V') = (V \setminus V', V')$. We also use the notation ω_G^+ and ω_G^- to specify the underlying graph G. For any subset V' of V, we denote by $G[V']$ the subgraph of G induced by V'. For any subset V' of V we denote by $[V', V \setminus V']$ the *cut* defined by V'. The *value of the cut* $[V', V \setminus V']$ is $\sum_{e \in \omega^+(V')} u(e)$. In the more general setting when every arc $e \in E$ has an additional lower bound $l(e)$ on the arc flow, the value of the cut $[V', V \setminus V']$ is defined by $\nu([V', V \setminus V']) = \sum_{e \in \omega^+(V')} u(e) - \sum_{e \in \omega^-(V')} l(e)$. The notation $\nu_G([V', V \setminus V'])$ is used to specify the underlying network G. A cut $[V', V \setminus V']$ in G is called *elementary* if $G[V']$ is connected. For two distinct vertices $s, t \in V$, a cut $[V', V \setminus V']$ is called an s–t cut if $s \in V', t \notin V'$.

A function $f : E \to \mathbb{R}$ is called a *flow* in G (or simply flow if there is no danger of ambiguity) if it satisfies the following constraints:

(i) $0 \leq f(e) \leq u(e) \quad \forall\, e \in E$
(ii) $\sum_{e\in\omega^+(v)} f(e) - \sum_{e\in\omega^-(v)} f(e) = 0 \quad \forall\, v \in V \setminus \{s, t\}$
(iii) $\sum_{e\in\omega^+(s)} f(e) - \sum_{e\in\omega^-(s)} f(e) \geq 0$.

For a flow f we define its *value* $\nu(f)$ by $\sum_{e\in\omega^+(s)} f(e) - \sum_{e\in\omega^-(s)} f(e)$. A *maximum flow* is a flow f with maximum value. The value of a maximum flow in a flow network G is denoted by $\nu^{\max}(G)$. A flow satisfying condition (ii) for all nodes is called a *circulation*. Additionally, capacities can also be assigned to vertices that are neither sources nor sinks. In this case the capacities are represented by extending the capacity function u to $E \cup (V \setminus \{s, t\})$, i.e., the capacity of $v \in V$ is denoted by $u(v)$. In a network G with capacities on vertices, a flow in G has to satisfy the following additional set of constraints:

(iv) $\sum_{e\in\omega^+(v)} f(e) \leq u(v) \quad \forall\, v \in V \setminus \{s, t\}$.

In the context of network flow interdiction, with every arc and node of the network $p \in V \cup E$ an *interdiction cost* $c(p) \in \{0, 1, \ldots\} \cup \{\infty\}$ is associated (with $c(s) = c(t) = \infty$). The network $G = (V, E, u, s, t, c)$ is called an *interdiction network*. An interdiction network has *unit interdiction costs* if $c(p) \in \{1, \infty\}\ \forall\ p \in V \cup E$. For some set $R \subseteq V \cup E$ we denote by $G \setminus R$ the subgraph of G obtained by removing the arcs and vertices contained in R (when removing a vertex, all arcs adjacent to this vertex are removed, too). For some given budget $B \in \{0, 1, \ldots\}$, a set $R \subseteq V \cup E$ is called an *interdiction set* if its cost does not exceed B, i.e., $c(R) = \sum_{r\in R} c(r) \leq B$. The *network flow interdiction problem* asks to find an interdiction set R that minimizes the value of a maximum flow on the graph $G \setminus R$. The value of this minimum maximum flow corresponding to budget B is denoted by $\nu_B^{\max}(G)$ (we therefore have $\nu^{\max}(G) = \nu_0^{\max}(G)$). An interdiction set R is called *optimal* if it minimizes the maximum flow with respect to the given budget. Furthermore, an optimal interdiction set R is called *efficient* if its interdiction cost $c(R)$ is minimum among all optimal interdiction sets and it is called *minimal* when removing any arc from the interdiction set results in a non-optimal interdiction set. We define the *network flow security* problem to be the problem of finding the minimal budget necessary to decrease the maximum flow by at least one unit, i.e., $\min\{B \in \{0, 1, 2, \ldots\} \mid \nu_B^{\max}(G) < \nu^{\max}(G)\}$.

The above definitions and problems can easily be extended to interdiction networks with multiple sources and sinks with fixed supply/demand. In this case an interdiction network is given by $G = (V, E, u, S, T, c, d)$ where $S, T \subseteq V$ with $S \cap T = \emptyset$ are the sets of sources and sinks and the function $d : V \to \mathbb{Z}$ is the demand/supply function which satisfies $d(s) < 0\ \forall s \in S, d(t) > 0\ \forall t \in T$ and $\mathrm{d}(v) = 0\ \forall\ v \in V \setminus (S \cup T)$. We call a flow network *balanced*, if the sum of the supplies equals the sum of the demands, i.e., $-d(S) = d(T)$. A flow network is called *demand-satisfiable* if there exists a flow in the network that satisfies all demands. Flow networks which are balanced and demand-satisfiable are called *supply networks*. A flow that satisfies all demands is called a *saturating flow*.

To simplify notations, a circuit $\mathcal{C}$ in G will be represented by the set of arcs it contains. When considering a planar graph, we typically assume that a planar embedding of the graph is fixed. For further graph-theoretical terms used in this paper and not further specified in this section we refer to [18]. In particular, we want to highlight that in this paper circuits and paths are by definition node-disjoint in contrast to walks and closed walks.

2.2. Symmetry between capacities and interdiction costs

An interesting property of the network flow interdiction problem is that we have a symmetrical relation between capacities and interdiction costs in the following sense. Let $G = (V, E, u, s, t, c)$ be an interdiction network where, to simplify explanations, we assume that G does not allow vertex interdiction and does not contain vertex capacities. We will show that the natural decision problem that asks to determine for some fixed budget $B \in \{0, 1, \ldots\}$ and some fixed $K \in \{0, 1, \ldots\}$ whether $\nu_B^{\max}(G) \leq K$ can be solved by determining the solution of a decision problem of the same type on the same network with the difference that the roles of capacities and costs are exchanged.

By the max-flow min-cut theorem we have that $\nu_B^{\max}(G) \leq K$ if and only if there exists an s–t cut $[V', V \setminus V']$ that satisfies

$$\min\{u(\omega^+(V') \setminus R) \mid R \subseteq E, c(R) \leq B\} \leq K. \tag{1}$$

An s–t cut $[V', V \setminus V']$ satisfies (1) if and only if it satisfies

$$\min\{c(\omega^+(V') \setminus A) \mid A \subseteq E, u(A) \leq K\} \leq B \tag{2}$$

because of the following observation. Suppose that an s–t cut $[V', V \setminus V']$ satisfies (1) and let $R \subseteq E$ with $c(R) \leq B$ be a set attaining the minimum in (1). Since the set $A = \omega^+(V') \setminus R$ satisfies $u(A) \leq K$ and $c(\omega^+(V') \setminus A) \leq B$, Inequality (2) is satisfied. Conversely suppose that an s–t cut $[V', V \setminus V']$ satisfies (2). Let $A \subseteq E$ with $u(A) \leq K$ be a set attaining the minimum in (2). Then since the set $R = \omega^+(V') \setminus A$ satisfies $c(R) \leq B$ and $u(\omega^+(V') \setminus R) \leq K$, Inequality (1) is satisfied. Since the Inequality (2) is of the same form as the Inequality (1) with the roles of capacities and interdiction costs exchanged, we have finally shown the symmetric relationship between capacities and interdiction costs in network flow interdiction problems.

All pseudo-polynomial algorithms presented in this work are polynomial in the capacities and pseudo-polynomial in the interdiction costs. The above observation shows that by exchanging the roles of capacities and interdiction costs many of these algorithms can easily be transformed into pseudo-polynomial algorithms that have a running time which is polynomial in the interdiction costs and pseudo-polynomial in the capacities. However, the above discussion is done only for the network flow interdiction problem and does not imply that a network flow security problem can be transformed into another network flow security problem where the roles of costs and capacities are exchanged. Thus, the presented symmetry cannot be used to exchange the roles of costs and capacities in the algorithms introduced in Section 6 for solving network flow security problems with multiple sources and sinks.

3. Complexity

3.1. Previous results

We associate the following natural decision problems to the network flow interdiction problem and the network flow security problem, respectively.

NFI($\boldsymbol{G}$, $\boldsymbol{B}$, $\boldsymbol{K}$)(Decision version of network flow interdiction problem). *Given an interdiction network G, some interdiction budget $B \in \{0, 1, 2, \ldots\}$ and a value $K \in \{0, 1, 2, \ldots\}$, decide whether $\nu_B^{\max}(G) \leq K$.*

NFS($\boldsymbol{G}$, $\boldsymbol{B}$)(Decision version of network flow security problem). *Given an interdiction network G and an interdiction budget $B \in \{0, 1, 2, \ldots\}$, decide whether $\nu_B^{\max}(G) < \nu^{\max}(G)$.*

It is easy to observe that the NFS problem is a special case of NFI by choosing $K = \nu^{\max}(G) - 1$. Conversely, when working on a class of interdiction networks with a single source or sink, the NFI problem can be reduced to a NFS problem by the following construction. Suppose we have a single source s (the case of a single sink is analogous). We introduce a new vertex s' which replaces s as source and add a non-removable arc from s' to s with capacity equal to $K + 1$. The NFS problem on the modified interdiction network is then equivalent to the NFI problem on the initial interdiction network.

The following theorem was shown in [20] by reducing a maximum clique problem to an NFI Problem.

Theorem 1 (*[20]*)**.** NFI *is strongly NP-complete even when the underlying interdiction network is restricted to unit interdiction costs.*

Furthermore there is a simple reduction from the binary knapsack problem (c.f. [7] for more information on the binary knapsack problem) to an interdiction problem on a graph with only two vertices implying the following theorem [20].

Theorem 2 (*[20]*)**.** NFI *is NP-complete on planar graphs even when restricted to a single source and sink.*

Since there exists a pseudo-polynomial algorithm for network flow interdiction problems on planar graphs with a single source and sink [15], this class of problems is not strongly NP-complete. When working on interdiction networks with a single source or a single sink, we have by the aforementioned reducibility of the NFI problem to an NFS problem that Theorems 1 and 2 apply also to the NFS problem.

It is not known whether the class of interdiction problems on planar graphs with multiple sources and sinks is strongly NP-complete. Furthermore, the presented reduction from the NFI problem to the NFS problem is no longer possible on this class of networks. In Section 6 we introduce a pseudo-polynomial algorithm for solving the NFS problem on planar supply networks with multiple sources and sinks. However, this algorithm does not seem to generalize in a simple way to the NFI problem.

3.2. Relation between planar network flow interdiction and the k-densest subgraph problem in planar graphs

In the following we show that finding dense subgraphs of a given size on planar graphs can easily be modelled as a planar network flow interdiction problem with multiple sources and sinks. This result links the planar network flow interdiction problem to a more classical combinatorial problem. The problem of finding a densest subgraph of size k is often called the *k-densest subgraph problem* or the k-clustering problem and is formally defined as follows. Given an undirected graph $G = (V, E)$ and $k \in \{0, 1, \ldots, |V|\}$, find an induced subgraph of G over k vertices with a maximum number of edges. Whereas the k-densest subgraph problem is known to be NP-complete on a wide variety of graph classes [5], its complexity for the class of planar graphs is still open. A slight modification of the problem obtained by imposing that the subgraph must be connected was shown to be NP-complete on planar graphs [10].

Theorem 3. *The k-densest subgraph problem on a planar graph can be reduced in polynomial time to a network flow interdiction problem on a planar graph.*

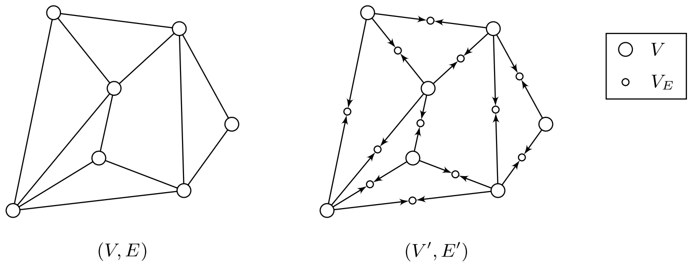

Fig. 1. Topology of the auxiliary graph (V', E') used in the proof of Theorem 3.

Proof. Let $G = (V, E)$ be a planar undirected graph. Consider the following planar interdiction network $G' = (V', E', u, S, T, c, d)$. The underlying graph (V', E') is obtained from G by subdividing all edges, i.e., on every edge $e \in E$, a new node v_e is added. We thus obtain a bipartite planar graph where each edge has one endpoint in V and the other one in the set of newly added vertices V_E (hence $V' = V \cup V_E$). By directing all edges from V to V_E, we get E' (cf. Fig. 1). The sets containing the sources and sinks are defined as follows: $S = V, T = V_E$. All arcs have unit capacity, every source has a supply equal to its outdegree, i.e., $d(s) = -|\omega(s)|\ \forall s \in S$ and every sink has unit demand. Furthermore, all arcs and all vertices of V_E are non-removable (they have an interdiction cost of ∞) and the vertices in V have an interdiction cost equal to one. For some fixed budget $B \in \{0, 1, 2 \ldots\}$, an optimal interdiction set in G' corresponds exactly to the vertices of a B-densest subgraph in G because of the following observation. For some fixed interdiction set R, the decrease of flow by removing the vertices in R corresponds to the number of sinks for which both neighbors are in R. This corresponds to the number of edges in G that have both endpoints in R. $\square$

4. Planar duality and current pseudo-polynomial algorithms

Planarity is a very helpful property when dealing with interdiction problems since the problem seems to have a simpler form when restated on the planar dual of the original interdiction network. We first introduce the planar dual of an interdiction network, which can be seen as a generalization of the classical planar dual. In a second step we propose a pseudo-polynomial algorithm for planar network flow interdiction with a single source and a single sink and without vertex removals. This algorithm is a direct generalization of an algorithm introduced in [15], which was designed only for undirected networks. Furthermore, the algorithm we present does not allow partial arc removals whereas the algorithm presented in [15] did allow it. However, this makes no significant difference since the technique applies easily to both types of arc removals. The extensions we propose in the following sections will overcome some restrictions of the algorithm presented in this section.

4.1. Planar duality in the context of interdiction networks

The classical planar dual, which is also called *geometric dual* or simply *dual*, of a directed graph is constructed on the base of a planar embedding by placing a vertex in each face of the original graph and connecting two vertices by an arc if they correspond to faces in the original graph sharing an arc. This gives a natural one-to-one correspondence between arcs in the original graph and arcs in the dual graph (dual arcs) as well as faces in the original graph and vertices in the dual graph, and vice versa. By convention, the dual arcs are oriented such that they cross the corresponding original arcs from right to left. See [12] for more details.

We extend the notion of planar duality to networks with lower and upper bounds on the arc flows and interdiction costs on the arcs. Even though the network given in a network flow interdiction problem does not contain lower bounds on the arc flows, we consider them here since in later sections auxiliary networks are used that contain lower bounds on the arc flows. Let $G = (V, E, l, u, c)$ be a directed planar network where for every arc $e \in E$, $l(e), u(e), c(e) \in \{0, 1, 2, \ldots\}$ correspond to the lower bound on the arc flow, the capacity and interdiction cost of arc e ($l(e) \leq u(e)\ \forall e \in E$). We define the dual $G^* = (V^*, E^*, \lambda^*, c^*)$ of the network G in the following way. The graph (V^*, E^*) is the planar dual of the graph (V, E) in the classical sense with the single difference that for every arc in the dual we add a reverse arc. For every arc $e \in E$ we denote by e^D the corresponding dual arc (as in the classical sense) and by e^D_R the reverse arc of e^D (cf. Fig. 2). The function $\lambda^* : E^* \to \mathbb{Z}$ is an integral length function in the network G^*, defined by $\lambda^*(e^D) = u(e)$ and $\lambda^*(e^D_R) = -l(e)\ \forall e \in E$. The cost function c^* is defined by $c^*(e^D) = c(e)$, $c^*(e^D_R) = 0\ \forall e \in E$.

For every cut $[V', V \setminus V']$ in G we denote its corresponding dual arcs by $\mathcal{C}^*(V') = \{e^D \in E^* \mid e \in (V', V \setminus V')\} \cup \{e^D_R \in E^* \mid e \in (V \setminus V', V')\}$. Note that the set $\mathcal{C}^*(V')$ is a set of edge-disjoint, non-overlapping circuits in (V^*, E^*), where non-overlapping is defined as follows. Let $\mathcal{C}^*_1, \mathcal{C}^*_2$ be two circuits in G^* and $V_1, V_2 \subseteq V$ be the vertices in V surrounded in counterclockwise sense by $\mathcal{C}^*_1$ and $\mathcal{C}^*_2$, respectively. We say that $\mathcal{C}^*_1, \mathcal{C}^*_2$ *do not overlap* if $V_1 \cap V_2 = \emptyset$. The following proposition highlights the correspondence between elementary cuts in the network G and circuits in its dual G^*.

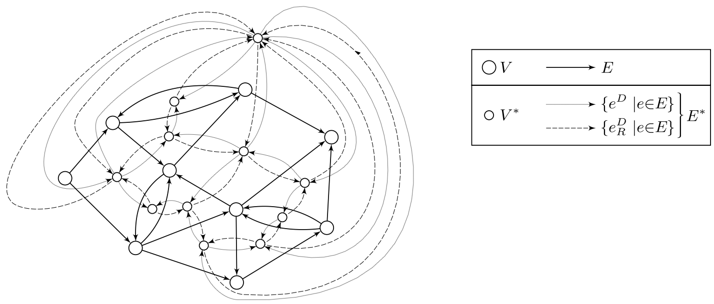

Fig. 2. Example dual graph (V^*, E^*) drawn over a given original graph (V, E).

Proposition 1. *The function that associates with every elementary cut $[V', V \setminus V']$ its corresponding dual arcs $\mathcal{C}^*(V')$ is a one-to-one mapping between elementary cuts in G and circuits in G^*. Furthermore, the value of an elementary cut in G is equal to the length of its corresponding circuit in G^*, i.e., for any elementary cut $[V', V \setminus V']$ in G, we have*

$$\nu([V', V \setminus V']) = \sum_{e^* \in \mathcal{C}^*(V')} \lambda^*(e^*).$$

Proof. The one-to-one property follows easily by observing that for any circuit $\mathcal{C}_0^*$ in G^*, the set V' of all vertices being surrounded in counterclockwise sense by $\mathcal{C}_0^*$ satisfies $\mathcal{C}^*(V') = \mathcal{C}_0^*$. The equality between the value of a cut in G and the sum of the lengths of the corresponding dual arcs follows directly from the definition of λ^*. $\square$

In particular, when dealing with a flow network $G = (V, E, l, u, s, t)$ with a single source s and a single sink t, one can easily check that elementary s–t cuts in G correspond to counterclockwise s–t separating circuits in G^*, where a circuit is called *counterclockwise s–t separating* if it is a circuit surrounding in counterclockwise sense s and separating s from t, i.e., s and t do not lie in the same of the two faces defined by the circuit.

Since the capacities on G are nonnegative, we have that there is a minimum s–t cut in G which is elementary. Proposition 1 thus implies that for every minimal cut in G, there exists a corresponding circuit in the dual G^* with length equal to the value of the cut. Therefore, a minimum s–t cut in G can be found by finding for a counterclockwise s–t separating circuit in G^* with minimum length.

In the following we discuss how this correspondence described by Proposition 1 can be extended for solving network flow interdiction problems on planar graphs with a single source and a single sink in pseudo-polynomial time.

4.2. A pseudo-polynomial algorithm for single source, single sink network flow interdiction on planar graphs without vertex removal

We now construct a pseudo-polynomial algorithm for solving the network flow interdiction problem on planar directed graphs with a single source s and a single sink t and without vertex removal, which is a direct generalization of an algorithm presented in [15] (which was designed for undirected networks). This algorithm nicely illustrates the techniques currently used for creating pseudo-polynomial network flow interdiction algorithms on planar graphs. Given is an interdiction network $G = (V, E, u, s, t, c)$ with non-removable vertices, i.e., $c(r) = \infty\ \forall r \in V$, and interdiction budget B. For every interdiction set $R \subseteq E$ we fix a minimum s–t cut in $G \setminus R$ that we denote by $[V_R, V \setminus V_R]$. We therefore have $\nu^{\max}(G \setminus R) = \nu([V_R, V \setminus V_R]) - \sum_{e \in R \cap \omega^+(V_R)} u(e)$. Note that an efficient optimal interdiction set R must satisfy $R \subseteq \omega^+(V_R)$ since otherwise the interdiction set $R' = R \cap \omega^+(V_R)$ would reduce the maximum flow by the same value as R and has lower interdiction cost. The *reduced value* of an s–t cut C (with respect to the budget B) is defined as the minimum value of C in $G \setminus R$ over all interdiction sets R. Note that the problem to find for some given s–t cut C an interdiction set R that minimizes the value of C in $G \setminus R$is a binary knapsack problem. By the max-flow min-cut theorem, we have that the network flow interdiction problem is equivalent to finding an s–t cut with minimum reduced value. Since we have no lower flow bounds on the arcs, we can restrict our search to minimum s–t cuts that are elementary, since for every s–t cut $[V', V \setminus V]$ we can define an elementary s–t cut $[V'', V \setminus V'']$ where V'' is the set of vertices in the connected component of $G[V']$ that contains s. One can easily check that $\omega^+(V'') \subset \omega^+(V')$ implying that the value of the cut $[V'', V \setminus V'']$ is smaller or equal than the value of $[V', V \setminus V']$.

The main idea of the algorithm is to find an efficient optimal interdiction set R by finding a corresponding elementary s–t cut with minimal reduced value. This is done by translating the problem into the dual. For any set of arcs $U^* \subseteq E^*$, we define its *reduced length* (with respect to B) by

$$\lambda_B^*(U^*) = \min \left\{ \sum_{e^* \in U^* \setminus X^*} \lambda^*(e^*) \mid X^* \subseteq U^*, \sum_{e^* \in X^*} c^*(e^*) \le B \right\}.$$

Similarly, for a walk W^* in G^* along the arcs $(e_1^*, e_2^*, \ldots, e_k^*)$, we define

$$\lambda_B^*(W^*) = \min \left\{ \sum_{i \in \{1,2,\ldots,k\} \setminus I} \lambda^*(e_i^*) \mid I \subseteq \{1, 2, \ldots, k\}, \sum_{i \in I} c^*(e_i^*) \leq B \right\}.$$

By the correspondence between elementary s–t cuts in G and s–t separating counterclockwise circuits in G^* as highlighted in Section 4.1, we have that the problem of finding an s–t cut in G with minimal reduced value is equivalent to finding an s–t separating counterclockwise circuit with minimal reduced length in the dual. Such circuits can be described in the following way. Let P be any path in the graph G from vertex s to vertex t, we define $P^D = \{e^D \in E^* \mid e \in P\}$ and $P_R^D = \{e_R^D \in E^* \mid e \in P\}$. For any set of arcs $U^* \subseteq E^*$ we define its parity with respect to P by $p_P(U^*) = |U^* \cap P^D| - |U^* \cap P_R^D|$. By a result of [14] we have that for every circuit $\mathcal{C}^*$ in G^*, two consecutive crossings on P alternate between left–right crossing and right–left ones. This implies that every circuit $\mathcal{C}^*$ in G satisfies $p_P(\mathcal{C}^*) \in \{-1, 0, 1\}$. Furthermore, it is easy to observe that a circuit $\mathcal{C}^*$ in G^* has the properties to be counterclockwise s–t separating if and only if $p_P(\mathcal{C}^*) = 1$. We therefore have to solve the following problem.

Problem 1.

$$\operatorname{argmin}\{\lambda_B^*(\mathcal{C}^*) \mid \mathcal{C}^* \text{circuit in } G^* \text{with } p_P(\mathcal{C}^*) = 1\}.$$

Consider the following relaxation of Problem 1.

Problem 2.

$$\operatorname{argmin}\{\lambda_B^*(W^*) \mid W^* \text{closed walk in } G^* \text{with } p_P(W^*) = 1\}.$$

A solution to Problem 1 can easily be obtained on the base of a solution W^* of Problem 2 by the following observation. W^* can be partitioned into a disjoint union of circuits $\mathcal{C}_1^*, \mathcal{C}_2^*, \ldots, \mathcal{C}_k^*$. Furthermore, by modularity of the parity function p_P and the fact that $p_P(W^*) = 1$, we have $1 = p_P(W^*) = \sum_{i=1}^k p_P(\mathcal{C}_i^*)$. As the parity of each circuit is in $\{-1, 0, 1\}$, there is some index $i \in \{1, 2, \ldots, k\}$ with $p_P(\mathcal{C}_i^*) = 1$. From $\mathcal{C}_i^* \subseteq W^*$ follows that $\lambda_B^*(\mathcal{C}_i^*) \leq \lambda_B^*(W^*)$. By optimality of W^* for Problem 2 we thus have $\lambda_B^*(\mathcal{C}_i^*) = \lambda_B^*(W^*)$, and by the fact that Problem 2 is a relaxation of Problem 1 follows that $\mathcal{C}_i^*$ is an optimal solution for Problem 1. More generally, the above reasoning shows that minimal solutions of Problem 2 correspond to solutions of Problem 1 and vice versa. We will solve Problem 2 by solving a sequence of problems of the following type, where v^* is some fixed vertex in V^*.

Problem 3.

$$\operatorname{argmin}\{\lambda_B^*(W^*) \mid W^* \text{closed walk in } G^* \text{containing vertex } v^*, p_P(W^*) = 1\}.$$

Solving Problem 3 for all vertices $v^* \in V^*$ and choosing among those solutions the closed walk with minimum reduced value solves Problem 2. However, as a closed walk solving Problem 2 must pass at least once by an arc in P^D, it suffices to solve Problem 3 for all vertices in G^* with at least one outgoing arc in P^D. Therefore, at most $|P|$ instances of Problem 3 have to be solved to get a solution to Problem 2.

Problem 3 can be formulated as a multi-objective shortest path problem with the three objectives budget, length and parity on a network $\overline{G}$ defined as follows. $\overline{G} = (V^*, \overline{E})$ is obtained from the graph $G^* = (V^*, E^*)$ by doubling every arc. For every arc $e^* \in E^*$, we denote by $e^{*,1}, e^{*,2}$ the two corresponding parallel arcs in $\overline{E}$. With every arc $e^{*,i} \in \overline{E}$ we associate a parity value $\overline{p}_P(e^{*,i}) = p_P(e^*)$, a length

$$\overline{\lambda}(e^{*,i}) = \begin{cases} \lambda^*(e^*) & \text{if } i = 1 \\ 0 & \text{if } i = 2 \end{cases}$$

and a budget value

$$\overline{c}(e^{*,i}) = \begin{cases} 0 & \text{if } i = 1 \\ c^*(e^*) & \text{if } i = 2. \end{cases}$$

We define the following natural correspondence between walks in $\overline{G}$ and walks in G^*. With every walk $\overline{W}$ in $\overline{G}$, we associate a walk $W^*(\overline{W})$ in G^* which is obtained by replacing each arc $e^{*,i}$ of $\overline{W}$, where $i \in \{1, 2\}$, with e^*. Conversely let W^* be a walk in G^* going along the arcs $(e_1^*, e_2^*, \ldots, e_k^*)$. We denote by $R^*(W^*) \subseteq \{e_1^*, e_2^*, \ldots, e_k^*\}$ an interdiction set in G^* with respect to the budget B that satisfies $\lambda_B^*(W^*) = \lambda^*(W^*) - \lambda^*(R(W^*))$. Such a set $R^*(W^*)$ exists by definition of λ_B^* and can be determined by solving a binary knapsack problem. The walk $\overline{W}(W^*)$ in $\overline{G}$ that corresponds to the walk W^* is defined by the sequence of arcs $(\overline{e}_i)_{i \in \{1,2,\ldots,k\}}$ where

$$\overline{e}_i = \begin{cases} e_i^{*,1} & \text{if } e_i^* \notin R(W^*) \\ e_i^{*,2} & \text{if } e_i^* \in R(W^*). \end{cases}$$

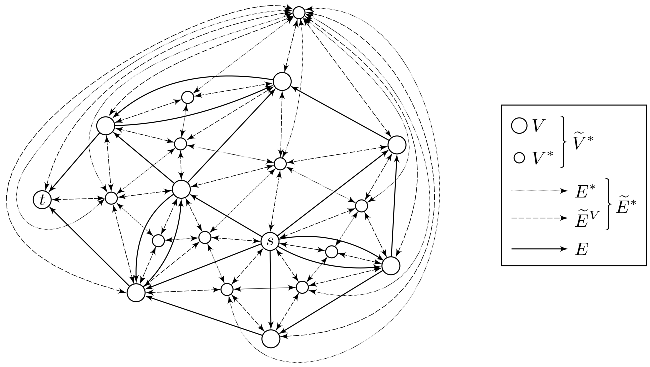

Fig. 3. Topology of the auxiliary graph $(\widetilde{V}^*, \widetilde{E}^*)$ used for modelling vertex interdiction. To simplify the drawing, lines with arrowheads on both sides represent two oppositely directed arcs.

The doubling of the arcs for the network $\overline{G}$ represents that for a walk W^* in G^*, every arc in W^* either contributes to the reduced length $\lambda^*_B(W^*)$ (this corresponds to the arcs in $\overline{E}$ with superscript one) or will not be considered in $\lambda^*_B(W^*)$ as it will be removed (this corresponds to the arcs in $\overline{E}$ with superscript two). Therefore, a walk $\overline{W}$ in $\overline{G}$ can be seen as a representation of the path $W^*(\overline{W})$ and a set $R^* \subseteq W^*(\overline{W})$ of arcs to be interdicted. It is thus easy to verify that the introduced correspondence between walks in G^* and $\overline{G}$ satisfies the following property.

Property 1. (a) *Let $\overline{W}$ be a walk in $\overline{G}$. We have that $W^*(\overline{W})$ satisfies $p_P(W^*(\overline{W})) = \overline{p}_P(\overline{W})$ and $\lambda^*_{\overline{c}(\overline{W})}(W^*) \le \overline{\lambda}(\overline{W})$.*

(b) *Let W^* be a walk in G^* and B some fixed budget. Then $\overline{W}(W^*)$ is a walk in $\overline{G}$ with $\overline{c}(\overline{W}) \le B$, $\overline{p}_P(\overline{W}) = p_P(W^*)$ and $\overline{\lambda}(\overline{W}) = \lambda^*_B(W^*)$.*

Let $v^* \in V^*$ be some fixed vertex and $\overline{W}$ a closed walk in $\overline{G}$ containing v^*, having parity equal to one, a budget value bounded by B and with minimal length among all those closed walks. By Property 1, $W^*(\overline{W})$ is then a solution to Problem 3. Finding such a $\overline{W}$ is therefore a multi-objective shortest path problem in $\overline{G}$ which can be transformed by standard techniques into a classical (single-objective) shortest path problem with positive edge-weights that can finally be solved in $O(B|P|n\log(n))$ time (c.f. [22]). Since Problem 2 can be solved by solving at most $|P|$ instances of Problem 3 we get an overall complexity of $O(B|P|^2 n\log(n))$ for solving Problem 2. By exchanging the roles of budget and length, an algorithm is obtained with running time $O(\nu_B^{\max}(G)|P|^2 n\log(n))$.

5. Incorporating vertex interdiction and vertex capacities

In this section we show how vertex interdiction and vertex capacities can be incorporated into the method presented in the previous section by adapting the dual network. Therewith, we answer a question raised in [15]. We begin by introducing the possibility of vertex interdiction and observe afterwards how vertex capacities can be added to the model. The role of the dual network G^* will be replaced by a modified dual $\widetilde{G}^*$ which allows for modelling vertex interdiction basically as arc interdiction. A similar technique was used in [11] for modelling vertex capacities in planar flow problems. The modified dual network will be introduced in Section 5.1 for networks that may have lower bounds on the arc flows since in some of the algorithms to be presented auxiliary networks are used that contain lower bounds on the arc flows. Furthermore, a correspondence between elementary cuts in the original network and circuits in the modified dual network will be established in the context of vertex interdiction. In Section 5.2 we show how a pseudo-polynomial algorithm can be obtained for network flow interdiction problems with the possibility of vertex interdiction and vertex capacities by transforming the problem to the modified dual network.

5.1. A modified dual network for vertex interdiction

Let $G = (V, E, l, u, c)$ be a network with lower bound l and upper bound u imposed on the arc flows and with interdiction costs defined by c. We define a modified dual network $\widetilde{G}^* = (\widetilde{V}^*, \widetilde{E}^*, \widetilde{\lambda}^*, \widetilde{c}^*)$ as an extended version of $G^* = (V^*, E^*, \lambda^*, c^*)$ as follows. For every $v^* \in V^*$ we denote by $f(v^*)$ the face of G corresponding to v^*. The vertex set of the network $\widetilde{G}^*$ is $V^* \cup V$. The arc set $\widetilde{E}^*$ is defined by $\widetilde{E}^* = E^* \cup \widetilde{E}^V$ where $\widetilde{E}^V$ contains the two arcs (v, v^*) and (v^*, v) for every pair of $v \in V$ and $v^* \in V^*$ where $f(v^*)$ is a face adjacent to v (cf. Fig. 3). The length function $\widetilde{\lambda}^*$ and the cost function $\widetilde{c}^*$ are extensions of λ^* and c^* on the arcs $\widetilde{E}^*$ defined as follows.

$$\begin{aligned}
\widetilde{\lambda}^*(e^*) &= \lambda^*(e^*) && \forall e^* \in E^* \\
\widetilde{\lambda}^*(v, v^*) &= 0 && \forall (v, v^*) \in \widetilde{E}^V \cap V \times V^* \\
\widetilde{\lambda}^*(v^*, v) &= \infty && \forall (v^*, v) \in \widetilde{E}^V \cap V^* \times V
\end{aligned}$$

$$\begin{aligned} \widetilde{c^*}(e^*) &= c^*(e^*) && \forall e^* \in E^* \\ \widetilde{c^*}(v, v^*) &= 0 && \forall (v, v^*) \in \widetilde{E^V} \cap V \times V^* \\ \widetilde{c^*}(v^*, v) &= c(v) && \forall (v^*, v) \in \widetilde{E^V} \cap V^* \times V. \end{aligned}$$

For some subset of arcs $\widetilde{U^*} \subseteq \widetilde{E^*}$ we define their reduced length in $\widetilde{G^*}$ with respect to the budget B by

$$\widetilde{\lambda}^*_B(\widetilde{U^*}) = \min \left\{ \sum_{\widetilde{e^*} \in \widetilde{U^*} \setminus X^*} \widetilde{\lambda^*}(\widetilde{e^*}) \mid X^* \subseteq \widetilde{U^*}, \sum_{\widetilde{e^*} \in X^*} \widetilde{c^*}(\widetilde{e^*}) \le B \right\}.$$

Furthermore, for a circuit $\widetilde{C^*}$ in $\widetilde{G^*}$ we denote by $V^{\circlearrowleft}_{\widetilde{C^*}}$ the subset of vertices in V that are surrounded in counterclockwise sense by $\widetilde{C^*}$. The construction of $\widetilde{G^*}$ is motivated by the following correspondence between the reduced value of elementary cuts in G and reduced costs of circuits in $\widetilde{G^*}$.

Theorem 4. *Let $V' \subset V$, $\emptyset \neq V' \neq V$ be a proper subset of the vertices such that $G[V']$ is connected and let $B \in \{0, 1, \ldots\}$. We have the following equality:*

$$\min\{\nu_{G \setminus R}([V', V \setminus V']) \mid R \subseteq E \cup (V \setminus V'), c(R) \le B\} = \min\{\widetilde{\lambda}^*_B(\widetilde{C^*}) \mid \widetilde{C^*} \text{ circuit in } \widetilde{G^*}, V^{\circlearrowleft}_{\widetilde{C^*}} = V'\}.$$

Proof. ($\geq$) We begin by proving that the optimization problem on the left side has an optimal value which is greater or equal to the one on the right side. Let $R \subseteq E \cup (V \setminus V')$ be an interdiction set and let $U^+ = \omega^+_{G \setminus R}(V')$, $U^- = \omega^-_{G \setminus R}(V')$ and $U = U^+ \cup U^-$. Using these definitions we can rewrite the capacity of the cut $[V', V \setminus V']$ in $G \setminus R$ as $\nu_{G \setminus R}([V', V \setminus V']) = u(U^+) - l(U^-)$. Since V' is the set of vertices of a connected component in $G \setminus (R \cup U)$, there exists a circuit $\widetilde{C^*}$ in $\widetilde{G^*}$ with $V^{\circlearrowleft}_{\widetilde{C^*}} = V'$ and consisting only of arcs that are either adjacent to vertices in R, are dual arcs of arcs contained in $R \cup U^+$ or are reverse arcs of dual arcs of U^-. We therefore have as desired

$$\widetilde{\lambda}^*_B(\widetilde{C^*}) \le u(U^+) - l(U^-) = \nu_{G \setminus R}([V', V \setminus V']).$$

($\leq$) Let $\widetilde{\mathcal{C}^*}$ be a circuit in $\widetilde{G^*}$ with $V^{\circlearrowleft}_{\widetilde{C^*}} = V'$ and satisfying $\widetilde{\lambda}^*_B(\widetilde{\mathcal{C}^*}) < \infty$ (when the reduced dual length is equal to ∞, the result follows trivially). Let $\widetilde{U^*}$ be a solution of

$$\underset{\widetilde{X^*} \subseteq \widetilde{\mathcal{C}^*}}{\operatorname{argmin}} \left\{ \sum_{\widetilde{e^*} \in \widetilde{\mathcal{C}^*} \setminus \widetilde{X^*}} \widetilde{\lambda^*}(\widetilde{e^*}) \mid \widetilde{c^*}(\widetilde{X^*}) \le B \right\}.$$

We assume without loss of generality that there is no arc $\widetilde{e^*} \in \widetilde{U^*}$ with $\widetilde{\lambda^*}(\widetilde{e^*}) \le 0$ since by removing those arcs from $\widetilde{U^*}$ we still get a solution to the above minimization problem. By definition of the reduced dual length we have $\widetilde{\lambda}^*_B(\widetilde{\mathcal{C}^*}) = \widetilde{\lambda^*}(\widetilde{\mathcal{C}^*}) - \widetilde{\lambda^*}(\widetilde{U^*})$. In the following we show how an interdiction set $R \subseteq E \cup (V \setminus V')$ satisfying $\nu_{G \setminus R}([V', V \setminus V']) \le \widetilde{\lambda}^*_B(\widetilde{C^*})$ can be derived from $\widetilde{U^*}$. Let $V_{\widetilde{\mathcal{C}^*}}$ be the subset of vertices in V through which the circuit $\widetilde{\mathcal{C}^*}$ passes and let $U^* = \widetilde{U^*} \cap E^*$. Because $\widetilde{\lambda}^*_B(\widetilde{\mathcal{C}^*}) < \infty$ we have that all arcs of $\widetilde{\mathcal{C}^*}$ entering one of the vertices in $V_{\widetilde{\mathcal{C}^*}}$ are contained in $\widetilde{U^*}$ since their length is ∞. The cost of $\widetilde{U^*}$ can therefore be decomposed into a term corresponding to the interdiction of arcs and one corresponding to the interdiction of vertices as follows.

$$\widetilde{c^*}(\widetilde{U^*}) = c^*(U^*) + c(V_{\widetilde{\mathcal{C}^*}}). \tag{3}$$

Let $U = \{e \in E \mid e^D \in U^*\}$ and we define $R = V_{\widetilde{\mathcal{C}^*}} \cup U$. Notice that since $\widetilde{U^*}$ does not contain arcs with non-positive lengths we have that there is no arc $e \in E$ with $e^D_R \in U^*$ since all reverse arcs of dual arcs have non-positive length. By (3) and the definition of $\widetilde{U^*}$ we have $c(R) = c(V_{\widetilde{\mathcal{C}^*}}) + c^*(U^*) = \widetilde{c^*}(\widetilde{U^*}) \le B$ showing that R is an interdiction set with respect to the budget B. Let $E^*_{\widetilde{\mathcal{C}^*}} = \widetilde{\mathcal{C}^*} \cap E^*$, $E^+_{\widetilde{\mathcal{C}^*}} = \{e \in E \mid e^D \in E^*_{\widetilde{\mathcal{C}^*}}\}$, $E^-_{\widetilde{\mathcal{C}^*}} = \{e \in E \mid e^D_R \in E^*_{\widetilde{\mathcal{C}^*}}\}$ and $E_{\widetilde{\mathcal{C}^*}} = E^+_{\widetilde{\mathcal{C}^*}} \cup E^-_{\widetilde{\mathcal{C}^*}}$. Since $\widetilde{\mathcal{C}^*}$ is a circuit with $V^{\circlearrowleft}_{\widetilde{C^*}} = V'$, there is no arc from a vertex in V' to a vertex in $V \setminus V'$ in the network $G \setminus (V_{\widetilde{\mathcal{C}^*}} \cup E_{\widetilde{\mathcal{C}^*}})$. Hence, removing the arcs $E_{\widetilde{\mathcal{C}^*}} \setminus U = (E^+_{\widetilde{\mathcal{C}^*}} \setminus U) \cup E^-_{\widetilde{\mathcal{C}^*}}$ from $G \setminus R$ destroys all paths from V' to $V \setminus V'$ and thus implies $\nu_{G \setminus R}([V', V \setminus V']) \le u(E^+_{\widetilde{\mathcal{C}^*}} \setminus U) - l(E^-_{\widetilde{\mathcal{C}^*}})$. The result is finally obtained by observing that $u(E^+_{\widetilde{\mathcal{C}^*}} \setminus U) - l(E^-_{\widetilde{\mathcal{C}^*}}) = u(E^+_{\widetilde{\mathcal{C}^*}}) - l(E^-_{\widetilde{\mathcal{C}^*}}) - u(U) = \widetilde{\lambda^*}(\widetilde{\mathcal{C}^*}) - \widetilde{\lambda^*}(\widetilde{U^*}) = \widetilde{\lambda}^*_B(\widetilde{\mathcal{C}^*})$. □

Theorem 4 will be used for transforming an interdiction problem on G that allows vertex interdiction to a problem of finding appropriate circuits in $\widetilde{G^*}$. Notice that even though the theorem just states equality between the optimal values of the two indicated optimization problems, the proof of Theorem 4 shows how the solution of one problem can be transformed to a solution of the other problem. When using Theorem 4 to reduce an interdiction problem to a problem of finding an appropriate circuit in the modified dual, we can therefore build a solution to the interdiction problem on the basis of the obtained circuit in the modified dual.

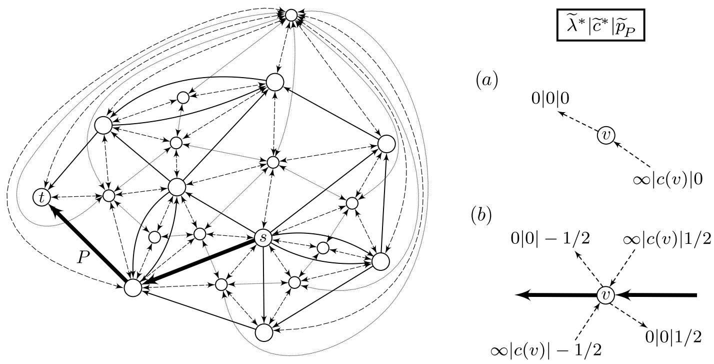

Fig. 4. Schematic description of how $\widetilde{\lambda}^*$, $\widetilde{c}^*$ and $\widetilde{p}_P$ are defined on arcs of $\widetilde{E}^V$ depending on whether they are adjacent to a node v that (a) does not lie on the path P or (b) does lie on P.

5.2. *Solving interdiction problems with vertex interdiction and vertex capacities*

In the following we present a pseudo-polynomial algorithm for the network flow interdiction problem on planar networks with a single source and sink that can handle vertex interdiction and vertex capacities. In a first step, we consider an interdiction problem allowing only vertex interdiction but no vertex capacities. Afterwards, we show how vertex capacities can be incorporated into the algorithm.

Let $G = (V, E, u, s, t, c)$ be an interdiction network allowing vertex interdiction but without vertex capacities and let B be some fixed budget. The interdiction problem on G can be formulated as the problem of finding a tuple (R, V') where R is an interdiction set, $V' \subset V \setminus R$ with $s \in V'$ and $t \notin V'$ and such that $\nu_{G\setminus R}([V', V \setminus V'])$ is minimal. Since we have no lower bounds on the arc flows, we can assume that an optimal solution (R, V') satisfies that the cut $[V', V \setminus V']$ is elementary in $G \setminus R$, since otherwise, V' can be replaced by the subset $V'' \subset V'$ that consists of the set of vertices in the connected component of $G[V']$ that contains s. Using Theorem 4 the interdiction problem can be reformulated on $\widetilde{G}^*$ as the problem of finding an s–t separating counterclockwise circuit with minimum reduced value.

For characterizing s–t separating counterclockwise circuits, we introduce an adapted version of the parity function. As in the previous section, let P be a path in G from s to t. Since in the graph $\widetilde{G}^*$ it is possible to cross P at a vertex, we have to take this possibility into account in the parity function. We therefore define a parity function $\widetilde{p}_P$ which is an extension of p_P on the subsets of $\widetilde{E}^*$ in the following way. For every vertex $v \in V \setminus \{s, t\}$ which lies on the path P we define the parity of arcs in $\widetilde{E}^*$ that are adjacent to v as follows. For every arc $\widetilde{e}^* = (v, v^*) \in \widetilde{E}^*$ which leaves P to the left side and every arc $\widetilde{e}^* = (w^*, v) \in \widetilde{E}^*$ which enters P on the right side, we set $\widetilde{p}_P(\widetilde{e}^*) = 1/2$. Similarly, for every arc $\widetilde{e}^* = (v^*, v) \in \widetilde{E}^*$ which enters P from the left side and every arc $\widetilde{e}^* = (v, w^*)$ which leaves P to the right side we set $\widetilde{p}_P(\widetilde{e}^*) = -1/2$. For all other arcs in $\widetilde{E}^* \setminus E^*$ we set $\widetilde{p}_P = 0$. Finally, for any set $\widetilde{U}^* \subseteq \widetilde{E}^*$ we define its parity by $\widetilde{p}_P(\widetilde{U}^*) = \sum_{\widetilde{e}^* \in \widetilde{U}^*} \widetilde{p}_P(\widetilde{e}^*)$. By this definition of the parity function we have, as desired, that the parity of a given walk corresponds to the difference between the number of times the walk crosses the path P from right to left and the number of times the walk crosses P from left to right. We thus have as in the previous section that a circuit $\widetilde{\mathcal{C}}^*$ in $\widetilde{G}^*$ has the properties to be counterclockwise s–t separating if and only if $\widetilde{p}_P(\widetilde{\mathcal{C}}^*) = 1$. Fig. 4 illustrates how $\widetilde{\lambda}^*$, $\widetilde{c}^*$ and $\widetilde{p}_P$ are defined on the arcs of $\widetilde{E}^V$.

Similar as in the previous section, an s–t separating counterclockwise circuit can be found by formulating a corresponding multi-objective shortest path problem by adapting the network $\widetilde{G}^*$. We therefore get a pseudo-polynomial algorithm for solving the network flow interdiction problem on planar graphs with a single source and sink and with the possibility to interdict arcs and vertices. The asymptotic complexity remains the same as in the previous section because the size of the graph $\widetilde{G}^*$ is only at most a constant factor larger than G^*, and the same is true for the number of times we have to solve subproblems of the type of Problem 2.

Upper capacities on the vertices can be introduced in the same way as shown in [11] by slightly modifying the network $\widetilde{G}^*$ as follows. Suppose that some vertex $v \in V$ has an upper capacity $u(v) \in \{1, 2, \ldots\}$. The modified network is built from the network $\widetilde{G}^*$ by setting length of every arc $(v^*, v) \in \widetilde{E}^*$, where $v^* \in V^*$, to $u(v)$. Therefore, in the calculation of the reduced dual length for a given s–t separating counterclockwise circuit $\widetilde{C}^*$ in $\widetilde{G}^*$, an arc in $\widetilde{C}^*$ that enters some vertex $v \in V$ does not need anymore to be interdicted, which corresponds to interdicting the vertex v, but can also be "in the cut" and contribute $u(v)$ to the reduced dual length. In [11] a justification of this construction in the case without interdiction (respectively by setting $B = 0$) is given which easily extends to the case with interdiction. To simplify the presentation of further results, we restrict ourselves in the following to flow networks without vertex capacities. Vertex capacities can easily be added by the above construction.

6. Pseudo-polynomial algorithm for network flow security on supply networks

In this section we present a pseudo-polynomial algorithm for the network flow security problem on planar supply networks with multiple sources and sinks. At first sight, network flow security on planar supply networks seems to be a

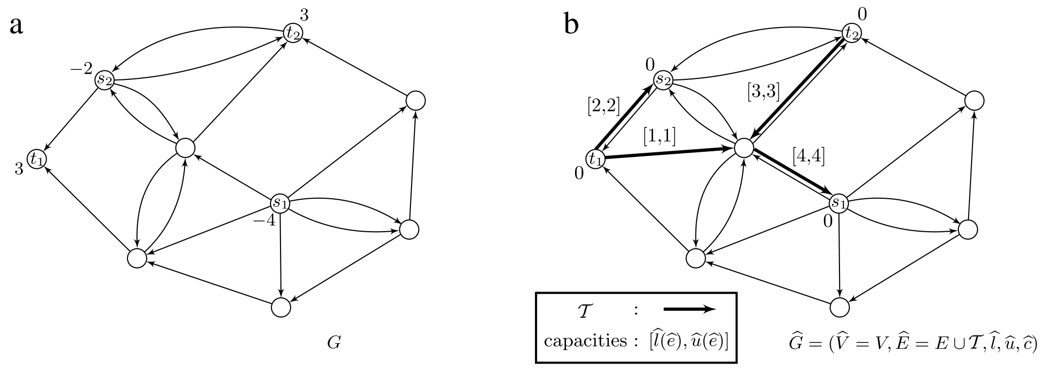

Fig. 5. Illustration of how the auxiliary arcs $\mathcal{T}$ are added. In (a) an initial graph G is shown containing two sources s_1, s_2 with demands $d(s_1) = -4$, $d(s_2) = -2$ and two sinks t_1, t_2 with demands $d(t_1) = 3$, $d(t_2) = 3$. In (b) a possible way of how the tree $\mathcal{T}$ can be added is shown and the resulting lower and upper bounds imposed on the arc flows of the arcs in $\mathcal{T}$ are indicated.

rather special case of network flow interdiction on planar networks. However a pseudo-polynomial algorithm for network flow security on planar supply networks can easily be transformed into a pseudo-polynomial algorithm for network flow interdiction on planar networks with a single source and sink. This can be done by using a reduction from network flow interdiction to network flow security analogous to the one presented in Section 3 as follows. Let G be a planar interdiction network with a single source s and a single sink t and B some fixed interdiction budget (without loss of generality we suppose that s and t have infinite supply and demand). For some fixed $K \in \mathbb{N}$ we can decide whether $\nu_B^{\max}(G) < K$ by solving the network flow security problem with budget B on the network G where we set the supply of s and the demand of t to the value K. If it is possible to reduce the maximum flow in this network with respect to the budget B then $\nu_B^{\max}(G) < K$ and otherwise $\nu_B^{\max}(G) \geq K$. A binary search over K allows therefore to determine the value of $\nu_B^{\max}(G)$ with a polynomial number of calls to the algorithm for network flow security on planar supply networks.

The method we propose for network flow security on planar supply networks works in two steps. In a first step, the underlying network flow problem is transformed into a circulation problem by using a technique introduced in [14]. This step will be explained in Section 6.1. In Section 6.2 we present how the transformed problem can be solved in pseudo-polynomial when vertex interdiction is not allowed. Since the proposed algorithm differs significantly from the approach presented in Section 4.2, the method for incorporating vertex interdiction has to be adapted. How this can be done is presented in Section 6.3.

6.1. Transformation to circulation problem

In this section we briefly present how the problem of testing whether a balanced flow problem with multiple sources and sinks is a supply network can be transformed into a circulation problem. This transformation was introduced in [14]. Let $G = (V, E, u, S, T, d)$ be a balanced planar flow network with upper bounds on the arc flows designated u, source set $S \subseteq V$ and sink set $T \subseteq V \setminus S$ and with demand/supply function d. The problem we want to solve is the following.

Problem 4. Does there exist a saturating flow in G?

The main idea is to send the flow from the sinks to the sources over auxiliary arcs to obtain a residual network $\widehat{G}$. The problem of finding a saturating flow in G is then equivalent to finding a circulation in $\widehat{G}$ that neutralizes the flow which was sent on the auxiliary arcs by sending the same amount of flow in the reverse direction. This can easily be formulated as a circulation problem in $\widehat{G}$ with lower bounds imposed on the flow on auxiliary arcs.

The network $\widehat{G} = (\widehat{V} = V, \widehat{E}, \widehat{l}, \widehat{u})$ is defined as follows. Let $\mathcal{T}'$ be a set of new edges that form an undirected tree over the vertices in V spanning the sources and sinks and that can be added to G without destroying planarity. For every edge $\{v, u\} \in \mathcal{T}'$ we denote by $V_{\mathcal{T}'}(v, u) \subseteq V$ the set of vertices in the connected component of $\mathcal{T}' \setminus \{v, u\}$ that contains v. We orient the edges in $\mathcal{T}'$ to obtain $\mathcal{T}$ in the following way. For $\{v, u\} \in \mathcal{T}'$we orient the edge from v to u if $d(V_{\mathcal{T}'}(v, u)) > 0$, otherwise we orient the edge from u to v. The set $\widehat{E}$ is defined to be $E \cup \mathcal{T}$. Furthermore, the lower and upper bounds $\widehat{l}$ and $\widehat{u}$ are extensions of l and u on the set $\widehat{E}$ defined by $\widehat{l}(v, u) = \widehat{u}(v, u) = d(V_{\mathcal{T}'}(v, u))\ \forall (v, u) \in \mathcal{T}$ (c.f. Fig. 5 for an illustration of the above construction).

As noted in [14] we have the following theorem.

Theorem 5. *There exists a saturating flow in G if and only if there exists a circulation in $\widehat{G}$.*

The following theorem can easily be obtained as a consequence of the max-flow min-cut theorem (c.f. [1]).

Theorem 6. *There exists a saturating flow in G if and only if there is no elementary cut with negative value in $\widehat{G}$.*

Additionally, whether a planar flow network admits a circulation can be determined by checking whether its dual contains negative circuits as follows. Let $\widehat{G}^* = (\widehat{V}^*, \widehat{E}^*, \widehat{\lambda}^*)$ be the dual network of $\widehat{G}$ as defined in Section 4, with the difference that we have no dual costs $\widehat{c}^*$ as we deal with a standard flow network and not an interdiction network. By Proposition 1 we have that $\widehat{G}$ has no elementary cut with negative value if and only if $\widehat{G}^*$ contains no negative circuits. Combining this result with Theorem 6 we finally get the following theorem.

Theorem 7. *There exists a saturating flow in G if and only if $\widehat{G}^*$ has no negative circuits.*

In the following, pseudo-polynomial algorithms for interdiction problems will be presented that are based on generalized versions of the above theorems.

6.2. Network security on planar graphs with multiple sources and sinks

By putting together the techniques for network flow interdiction on planar graphs with a single source and sink as presented in Section 4 and the results of the previous subsection, we can easily build a pseudo-polynomial algorithm for network flow security on planar supply networks without vertex interdiction as follows. Let B be a fixed budget and $G = (V, E, u, S, T, d, c)$ be a planar interdiction network without vertex removal whose underlying flow network is a supply network. Let $(\widehat{V}, \widehat{E}, \widehat{l}, \widehat{u})$ be the auxiliary network corresponding to (V, E, u, S, T, d) as defined in Section 6.1. We extend this auxiliary network to $\widehat{G} = (\widehat{V}, \widehat{E}, \widehat{l}, \widehat{u}, \widehat{c})$ where $\widehat{c}$ is an extension of c on the set $\widehat{E}$, satisfying $\widehat{c}(\widehat{e}) = \infty\ \forall \widehat{e} \in \widehat{E} \setminus E$. Let $\widehat{G}^* = (\widehat{V}^*, \widehat{E}^*, \widehat{\lambda}^*, \widehat{c}^*)$ be its corresponding dual network as defined in Section 4.

Interdicting an arc in G corresponds to setting its capacity to 0, which corresponds in the dual network $\widehat{G}^*$ to setting the length of the corresponding dual arc to 0. Hence by Theorem 7 we have that there is an interdiction set $R \subseteq E$ such that it is not possible to find a circulation in $G \setminus R$ if and only if $\widehat{G}^*$ contains a circuit with negative reduced value. To check whether such a circuit exists, we determine the circuit with minimum reduced cost in $\widehat{G}^*$. By using techniques analogous to the ones presented in Section 4 this problem can be reduced to solving multi-objective shortest path problems with the difference that we do not have to take parity into account. Again, using standard techniques for solving multi-objective shortest path problems, the problem of finding a circuit with minimum reduced cost in $\widehat{G}^*$ can be solved in $O(Bn^3)$ (c.f. [22]).

6.3. Generalization to the case with vertex interdiction

Again let $G = (V, E, u, S, T, d, c)$ be a planar interdiction network that is balanced, demand-satisfiable and allows arc and vertex removal (except for sources and sinks) and let B be a fixed budget. As in the case without vertex removal, we begin by reformulating the problem as an interdiction problem for circulations. Let $\widehat{G} = (\widehat{V} = V, \widehat{E}, \widehat{l}, \widehat{u}, \widehat{c})$ be the auxiliary graph as defined in Section 6.2 and as before we denote by $\mathcal{T} = \widehat{E} \setminus E$ the added tree arcs. We now discuss how arc and vertex removal in G translates to $\widehat{G}$. A removal of an arc in G simply corresponds to removing the same arc in $\widehat{G}$. However, vertex removal cannot be translated in such a direct way since the arcs in $\widehat{E} \setminus E$ are auxiliary arcs which should not be removed by a vertex removal. This is the main difference compared to the interdiction problems discussed in Section 5 and the reason why the method has to be adapted. For any interdiction set $R \subseteq V \cup E$, we denote by $\widehat{G}(R)$ the graph obtained from $\widehat{G}$ by removing all arcs contained in R and all arcs in E being adjacent to a vertex in R. The following theorem is the counterpart of Theorem 6 in the context of interdiction.

Theorem 8. *For any interdiction set R in G we have the following equivalence: There exists a saturating flow in $G \setminus R$ if and only if there is no elementary cut in $\widehat{G}(R)$ with value < 0.*

Proof. Let $G(R)$ be the network obtained from G by removing all arcs in R and all arcs adjacent to vertices in R. We trivially have that there is a saturating flow in $G \setminus R$ if and only if there is a saturating flow in $G(R)$. The network $\widehat{G}(R)$ can easily be obtained from $G(R)$ by applying the construction introduced in Section 6.1. Applying Theorem 6 finally proves the claim. □

In the following, we show how the problem of finding an interdiction set R and an elementary cut in $\widehat{G}(R)$with negative value can be mapped onto the problem of finding a circuit with negative reduced length in an adapted dual network $\widetilde{G}^*$. In a first step we assume that all vertices that are adjacent to auxiliary arcs, i.e. the arcs $\mathcal{T}$, cannot be interdicted. This restriction will be lifted in a second step.

6.3.1. Special case: all vertices adjacent to arcs in $\mathcal{T}$ cannot be interdicted

Assuming that no edge of $\mathcal{T}$ is adjacent to a vertex with finite interdiction cost we have for any interdiction set R in G, $G(R) = G \setminus R$. Let $\widetilde{G}^* = (\widetilde{V}^*, \widetilde{E}^*, \widetilde{\lambda}^*, \widetilde{c}^*)$ be the modified dual network for the network $\widehat{G}$ as introduced in Section 5.[1] The following theorem shows in the current context how the network security problem can be transformed to the dual. Since $\mathcal{T}$ does not touch vertices with finite interdiction cost, the transformation can be done in an analogous way as in Section 5.

[1] Using the notation $\widetilde{\widetilde{G}}^*$ instead of $\widetilde{G}^*$ would be more consistent. However, to simplify notations we chose the second form.

Theorem 9. *If all vertices that are adjacent to an arc in $\mathcal{T}$ cannot be interdicted, the following statements are equivalent.*

(i) *There exists an interdiction set R in G such that there is no saturating flow in $G \setminus R$.*
(ii) *There is a circuit $\widetilde{C}^*$ in $\widetilde{G}^*$, with negative reduced value, i.e., $\widetilde{\lambda}^*_B(\widetilde{C}^*) < 0$.*

Proof. By Theorem 8 and the property that for any interdiction set R in G we have $\widehat{G}(R) = \widehat{G} \setminus R$ we have that point (i) is equivalent to

(i′) *There exists an interdiction set R in G such that there is an elementary cut in $\widehat{G} \setminus R$ with value < 0.*

By applying Theorem 4 we have that statement (i′) is equivalent to

(i″) $\min\{\widetilde{\lambda}^*_B(\widetilde{C}^*) \mid \widetilde{C}^* \text{ circuit in } \widetilde{G}^*, \emptyset \neq V^{\circlearrowleft}_{\widetilde{C}^*} \neq V\} < 0$.

The proof will finally be completed by showing that any circuit $\widetilde{C}^*$ in $\widetilde{G}^*$ with either $V^{\circlearrowleft}_{\widetilde{C}^*} = \emptyset$ or $V^{\circlearrowleft}_{\widetilde{C}^*} = V$ satisfies $\widetilde{\lambda}^*_B(\widetilde{C}^*) \geq 0$ which shows the equivalence between statement (i″) and statement (ii). Let $\widetilde{C}^*$ be a circuit in $\widetilde{G}^*$ and let $\widetilde{U}^*_{\mathcal{T}}$ be the subset of arcs in $\widetilde{C}^*$ corresponding to dual arcs of $\mathcal{T}$ or their reverse arcs, i.e.,

$$\widetilde{U}^*_{\mathcal{T}} = \{\widetilde{e}^* \in \widetilde{C}^* \mid \exists e \in \mathcal{T} \text{ s.t. } e^D = \widetilde{e}^* \text{ or } e^D_R = \widetilde{e}^*\}.$$

Because all arcs in $\widetilde{U}^*_{\mathcal{T}}$ have infinite interdiction cost, the reduced length of $\widetilde{C}^*$ can be rewritten as follows:

$$\widetilde{\lambda}^*_B(\widetilde{C}^*) = \widetilde{\lambda}^*_B(\widetilde{C}^* \setminus \widetilde{U}^*_{\mathcal{T}}) + \widetilde{\lambda}^*(\widetilde{U}^*_{\mathcal{T}}). \tag{4}$$

By the definition of $\widetilde{\lambda}^*$, the arcs in $\widetilde{G}^*$ with negative lengths are given by $\{e^D_R \mid e \in E, l(e) > 0\}$. Since the only arcs in $\widehat{G}$ that may have positive lower bounds on the arc flows are those in $\mathcal{T}$, we have $\widetilde{\lambda}^*(\widetilde{e}^*) \geq 0$ for all arcs $\widetilde{e}^* \in \widetilde{C}^* \setminus \widetilde{U}^*_{\mathcal{T}}$ and thus

$$\widetilde{\lambda}^*_B(\widetilde{C}^* \setminus \widetilde{U}^*_{\mathcal{T}}) \geq 0. \tag{5}$$

In the following we show that $\widetilde{\lambda}^*(\widetilde{U}^*_{\mathcal{T}}) = 0$ if the circuit $\widetilde{C}^*$ satisfies $V^{\circlearrowleft}_{\widetilde{C}^*} \in \{\emptyset, V\}$. Combined with (4) and (5) this result will imply as desired $\widetilde{\lambda}^*_B(\widetilde{C}^*) \geq 0$ if $V^{\circlearrowleft}_{\widetilde{C}^*} \in \{\emptyset, V\}$. Let $V_{\mathcal{T}}$ be the subset of vertices in $\widehat{G}$ that are adjacent to arcs in $\mathcal{T}$ and let $\widehat{G}_{\mathcal{T}}$ be the subnetwork of $\widehat{G}$ over the vertices $V_{\mathcal{T}}$ and the arcs $\mathcal{T}$. Thus, $\widehat{G}_{\mathcal{T}}$ is a network whose underlying graph consists of the tree defined by $\mathcal{T}$. One can easily check that the value of any cut $[V', V_{\mathcal{T}} \setminus V']$ in $\widehat{G}_{\mathcal{T}}$ can be expressed as follows:

$$\nu_{G_{\mathcal{T}}}([V', V_{\mathcal{T}} \setminus V']) = \sum_{v \in V'} \mathrm{d}(v).$$

Since $\widetilde{U}^*_{\mathcal{T}}$ are the dual arcs corresponding to the arcs in the cut $[V^{\circlearrowleft}_{\widetilde{C}^*} \cap V_{\mathcal{T}}, V_{\mathcal{T}} \setminus V^{\circlearrowleft}_{\widetilde{C}^*}]$ we have

$$\widetilde{\lambda}^*(\widetilde{U}^*_{\mathcal{T}}) = \nu_{G_{\mathcal{T}}}([V^{\circlearrowleft}_{\widetilde{C}^*} \cap V_{\mathcal{T}}, V_{\mathcal{T}} \setminus V^{\circlearrowleft}_{\widetilde{C}^*}]) = \sum_{v \in V^{\circlearrowleft}_{\widetilde{C}^*} \cap V_{\mathcal{T}}} \mathrm{d}(v). \tag{6}$$

Since the set $V_{\mathcal{T}}$ contains by construction all sources and sinks, we have $\mathrm{d}(v) = 0$ for all $v \in V^{\circlearrowleft}_{\widetilde{C}^*} \setminus V_{\mathcal{T}}$. Therefore, (6) can be rewritten as

$$\widetilde{\lambda}^*(\widetilde{U}^*_{\mathcal{T}}) = \sum_{v \in V^{\circlearrowleft}_{\widetilde{C}^*}} \mathrm{d}(v).$$

By the above equation, we finally have $\widetilde{\lambda}^*(\widetilde{U}^*_{\mathcal{T}}) = 0$ if $V^{\circlearrowleft}_{\widetilde{C}^*} \in \{\emptyset, V\}$. □

The problem of testing whether there is a circuit with minimum reduced length in $\widetilde{G}^*$ can be solved by the same techniques that were used for the network flow security problem without vertex interdiction in Section 6.2. Since the network $\widetilde{G}^*$ is only by a constant factor larger than the network $\widehat{G}^*$ which was used in Section 6.2 we get the same running time of $O(Bn^3)$.

6.3.2. General case: arcs in $\mathcal{T}$ may be adjacent to vertices with finite interdiction costs

We finally consider the general case where the auxiliary arcs $\mathcal{T}$ may be adjacent to vertices with finite interdiction costs. This case will be reduced to the previous one by modifying the network G in such a way that the auxiliary arcs $\mathcal{T}$ can be chosen such that they are not adjacent to vertices with finite interdiction costs. More precisely, before introducing the auxiliary arcs $\mathcal{T}$, some arcs of the network G will be subdivided such that the tree $\mathcal{T}$ can bypass vertices with finite interdiction costs. This idea was also used in [11] in a different context. Note that an arc $e = (u, w)$ in the network G can easily be subdivided by introducing a new vertex v_e with infinite interdiction cost and replacing e by two arcs (u, v_e) and (v_e, w) with the same capacity and interdiction cost as e. One can easily check that the network security problem on a network with subdivided arcs is equivalent to the original one. Notice that when subdividing a set of arcs with the same

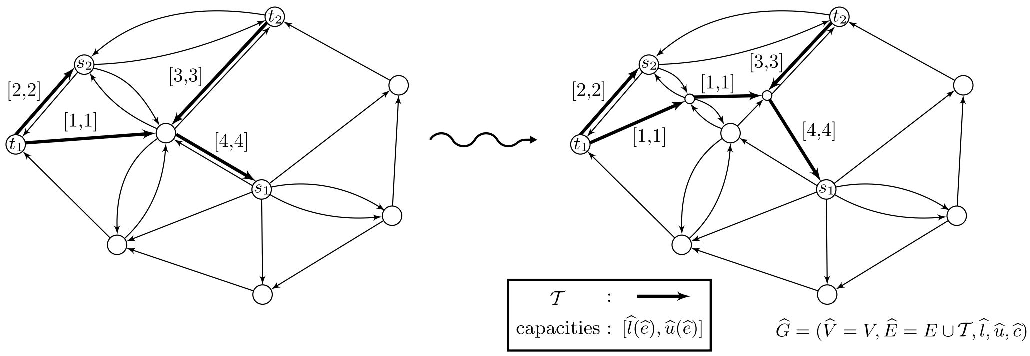

Fig. 6. Illustration showing how some initial candidate for $\mathcal{T}$ (bold arcs on the left image) can be transformed by edge subdivisions into a tree $\mathcal{T}$ (bold arcs on the right image) such that all vertices that are adjacent to arcs in $\mathcal{T}$ are either sources, sinks or artificial vertices which were introduced for subdividing arcs. The artificial vertices on the right image are indicated as small circles.

endpoints, it suffices to introduce only one artificial vertex instead of one for each arc. By subdividing arcs of G, it is easy to choose the artificial arcs $\mathcal{T}$ such that all vertices that are adjacent to $\mathcal{T}$ are either artificial vertices or sources and sinks (c.f. Fig. 6). Since artificially added vertices as well as sources and sinks cannot be interdicted, we are back in the previous case. Furthermore, since the size of the resulting graph $\widehat{G}$ is still bounded by $O(|V|)$, the network security problem can thus be solved in $O(Bn^3)$ time as in the previous case.

Vertex capacities can be introduced in the same way as presented in Section 5. Here too, we have to ensure that no auxiliary arc is adjacent to a capacitated vertex that is neither a source nor a sink. However, when choosing $\mathcal{T}$ such that all vertices adjacent to $\mathcal{T}$ are either sources, sinks or vertices that were artificially added through subdivisions, this condition is automatically satisfied.

7. Conclusions

We proposed a planarity-preserving transformation that allows the inclusion of vertex removals and vertex capacities in pseudo-polynomial interdiction algorithms for planar graphs. Additionally, a pseudo-polynomial algorithm was introduced for the network security problem on planar supply networks. This is the first pseudo-polynomial algorithm that can be used to solve non-trivial interdiction problems with multiple sources and sinks. Since the network security problem on supply networks can be seen as a generalization of the interdiction problem with a single source and sink, the introduced approach can solve a broader range of interdiction problems compared to previous pseudo-polynomial algorithms. We showed how the method for incorporating vertex removals and vertex capacities can be adapted for the new algorithm. However, the proposed algorithm does not seem to extend easily to general network flow interdiction problems with multiple sources and sinks. Thus, it is still open whether network flow interdiction on planar graphs with multiple sources and sinks is solvable in pseudo-polynomial time. We showed that the k-densest subgraph problem on planar graphs can be polynomially reduced to a network flow interdiction problem on a planar graph with multiple sources and sinks. The algorithms presented in this paper can easily be adapted to the case when multiple resources are needed for removing arcs and nodes, still remaining pseudo-polynomial. The main purpose of the algorithms presented in this paper was to show that various interdiction problems on planar graphs can be solved in pseudo-polynomial time. We expect, however, that it should be possible to speed up the proposed algorithms by using more elaborate techniques as, for example, nested dissection.

References

[1] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.
[2] N. Assimakopoulos, A network interdiction model for hospital infection control, Computers in Biology and Medicine 17 (6) (1987) 413–422.
[3] C. Burch, R. Carr, S. Krumke, M. Marathe, C. Phillips, E. Sundberg, A decomposition-based pseudoapproximation algorithm for network flow inhibition, in: Network Interdiction and Stochastic Integer Programming, Springer, 2003, pp. 51–68. (Chapter 3).
[4] K.J. Cormican, D.P. Morton, R.K. Wood, Stochastic network interdiction, Operations Research 46 (2) (1998) 184–197.
[5] D.G. Corneil, Y. Perl, Clustering and domination in perfect graphs, Discrete Applied Mathematics 9 (1984) 27–29.
[6] L.R. Ford, D.R. Fulkerson, Maximal flow through a network, Canadian Journal of Mathematics 8 (1956) 399–404.
[7] M.R. Garey, D.S. Johnson, Computers and Intractability; A Guide to the Theory of NP-Completeness, W.H. Freeman & Co., New York, NY, USA, 1990.
[8] P.M. Ghare, D.C. Montgomery, W.C. Turner, Optimal interdiction policy for a flow network, Naval Research Logistics Quarterly 18 (1971) 37–45.
[9] U. Janjarassuk, J. Linderoth, Reformulation and sampling to solve a stochastic network interdiction problem, Networks 52 (3) (2008) 120–132.
[10] J.M. Keil, T.B. Brecht, The complexity of clustering in planar graphs, Journal of Combinatorial Mathematics and Combinatorial Computing 9 (1991) 155–159.
[11] S. Khuller, J. Naor, Flow in planar graphs with vertex capacities, Algorithmica 11 (3) (1994) 200–225.
[12] E.L. Lawler, Combinatorial Optimization: Networks and Matroids, Holt, Rinehart and Winston, 1976.
[13] A.W. McMasters, T.M. Mustin, Optimal interdiction of a supply network, Naval Research Logistics Quarterly 17 (3) (1970) 261–268.
[14] G.L. Miller, J. Naor, Flow in planar graphs with multiple sources and sinks, SIAM Journal on Computing 24 (5) (1995) 1002–1017.
[15] C.A. Phillips, The network inhibition problem, in: STOC'93: Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, ACM Press, New York, NY, USA, 1993, pp. 776–785.

[16] H.D. Ratliff, G.T. Sicilia, S.H. Lubore, Finding the n most vital links in flow networks, Management Science 21 (5) (1975) 531–539.
[17] J. Salmeron, K. Wood, R. Baldick, Analysis of electric grid security under terrorist thread, IEEE Transaction on Power Systems 19 (2) (2004) 905–912.
[18] A. Schrijver, Combinatorial Optimization, Polyhedra and Efficiency, Springer, 2003.
[19] R. Wollmer, Removing arcs from a network, Operations Research 12 (6) (1964) 934–940.
[20] R.K. Wood, Deterministic network interdiction, Mathematical and Computer Modeling 17 (2) (1993) 1–18.
[21] R. Zenklusen, Combinatorial methods for analyzing network security and reliability. Ph.D. Thesis, ETH Zurich, No 18109, 2008.
[22] R. Zenklusen, Extensions to network flow interdiction on planar graphs, 2008. arxiv.org/abs/0801.1737v1.