Stackelberg Max Closure with Multiple Followers

Citation for published version (APA):

Jungnitsch, K., Peis, B., & Schröder, M. (2022). Stackelberg Max Closure with Multiple Followers. *Mathematics of Operations Research*, *47*(4), 3010-3024. https://doi.org/10.1287/moor.2021.1240

Document status and date:
Published: 01/11/2022

DOI:
10.1287/moor.2021.1240

Document Version:
Publisher's PDF, also known as Version of record

Document license:
Taverne

Mathematics of Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Stackelberg Max Closure with Multiple Followers

Karsten Jungnitsch, Britta Peis, Marc Schröder

MATHEMATICS OF OPERATIONS RESEARCH
Articles in Advance, pp. 1–15
ISSN 0364-765X (print), ISSN 1526-5471 (online)
http://pubsonline.informs.org/journal/moor

Stackelberg Max Closure with Multiple Followers

Karsten Jungnitsch,[a] Britta Peis,[a] Marc Schröder[b,*]

[a] School of Business and Economics, RWTH Aachen University, 52062 Aachen, Germany; [b] School of Business and Economics, Maastricht University, 6211 LK Maastricht, Netherlands
*Corresponding author

Contact: karsten-jungnitsch@t-online.de (KJ); peis@oms.rwth-aachen.de (BP); m.schroder@maastrichtuniversity.nl, https://orcid.org/0000-0002-0048-2826 (MS)

Received: June 4, 2020
Revised: January 25, 2021; October 27, 2021
Accepted: November 12, 2021
Published Online in Articles in Advance: April 1, 2022

MSC2020 Subject Classification: Primary: 68R10; secondary: 05C21, 91A65

https://doi.org/10.1287/moor.2021.1240

Abstract. In a Stackelberg max closure game, we are given a digraph whose vertices correspond to projects from which firms can choose and whose arcs represent precedence constraints. Some projects are under the control of a leader who sets prices in the first stage of the game, while in the second stage, the firms choose a feasible subset of projects of maximum value. For a single follower, the leader's problem of finding revenue-maximizing prices can be solved in strongly polynomial time. In this paper, we focus on the setting with multiple followers and distinguish two situations. In the case in which only one copy of each project is available (limited supply), we show that the two-follower problem is solvable in strongly polynomial time, whereas the problem with three or more followers is NP-hard. In the case of unlimited supply, that is, when sufficient copies of each project are available, we show that the two-follower problem is already APX-hard. As a side result, we prove that Stackelberg min vertex cover on bipartite graphs with a single follower is APX-hard.

Keywords: project selection • Stackelberg games • computational complexity

1. Introduction

The *project selection problem* or, more formally, the *max closure problem*, is one of the most common applications of the max-flow, min-cut theorem. Project selection deals with the trade-off between projects that yield revenue and supporting projects that cost money. Every firm is constantly faced with decisions of this nature.

The max closure problem is formally defined as follows. We are given a set $V = \{1, \dots, n\}$ of possible projects from which the firm can choose. An acyclic directed graph $D = (V, A)$ on the set of projects represents precedence constraints among pairs of projects. That is, an arc $(u, v) \in A$ indicates that project v needs to be selected whenever project u is selected. A subset $C \subseteq V$ of projects is *feasible* if it is a *closure* in D, that is, if none of the precedence constraints is violated. That is, $C \subseteq V$ is a closure in D if $N^+(C) = \{(u, v) \in A \mid u \in C, v \notin C\} = \emptyset$.

Each project $v \in V$ comes with some associated weight $w(v) \in \mathbb{R}$. If $w(v)$ is positive, then v is called a benefit project, and $w(v)$ reflects the gain achieved if project v is selected. We denote the set of benefit projects by $B^+ = \{v \in V \mid w(v) > 0\}$. For a given subset of benefit projects $S \subseteq B^+$, we define $w(S) = \sum_{v \in S} w(v)$. If $w(v)$ is negative, then v is called a cost project, and $|w(v)|$ reflects the cost that needs to be paid if project v is selected. We denote the set of cost projects by $B^- = \{v \in V \mid w(v) \leq 0\}$. For a given subset of cost projects $S \subseteq B^-$, we define $c(S) = \sum_{v \in S} |w(v)|$.

The max closure problem asks for a feasible subset of projects $C \subseteq V$ of maximum value $\sum_{v \in C} w(v)$. It is well-known (Picard [19]) that the max closure problem can be solved in strongly polynomial time via a reduction to a min s-t-cut problem in some auxiliary flow network.

In Stackelberg max closure, a distinguished player, called the leader, controls a subset of the projects on which prices are set in the first stage of the game. All other projects have a fixed weight. Afterward, in the second stage of the game, the followers see these prices as negative weights, and each of the followers selects a max closure in a follower-specific subgraph. The goal of the leader is to find revenue-maximizing prices while taking into account that the followers solve the induced max closure problems.

Stackelberg max closure problems, named after Von Stackelberg [22], belong to the class of combinatorial Stackelberg network pricing games. These games model leader–follower problems in which the items are, for example, the vertices or edges of an underlying directed or undirected graph, and the followers each solve an optimization problem by selecting a feasible subset of items at minimum or maximum cost from a predefined feasibility space consisting of, for example, matchings, spanning trees, origin–destination paths, vertex covers, or closures. The Stackelberg max closure problem is a natural variant that has not been considered before but is motivated by very common business decisions.

1.1. StackMaxClosure

In this paper, we focus on Stackelberg max closure games (StackMaxClosure) with multiple followers $K = \{1, \ldots, k\}$. We are given a digraph $D = (V, A)$ whose vertex set V is partitioned into red (priceable) and blue (fixed price) vertices $V = R \cup B$ together with fixed weights $\boldsymbol{w} : B \to \mathbb{R}$. Each follower $i \in K$ is associated with a follower-specific subgraph $D_i = (V_i, A_i)$. We denote by R_i the priceable vertices and by B_i the fixed price vertices of follower $i \in K$. The set of feasible strategies of follower $i \in K$ corresponds to the collection $\mathcal{F}_i \subseteq 2^{V_i}$ of closures in D_i.

We assume throughout this paper that the digraphs of two different followers may only intersect in priceable vertices. That is, we assume $B_i \cap B_j = \emptyset$ for $i, j \in K$ with $i \neq j$. Notice that, under this assumption, all kinds of situations can be modeled in which firms are interested in follower-specific benefit projects that might depend on the same-cost projects. Imagine, for example, two construction firms that are both interested in building certain houses (the benefit projects), and the construction depends on common resources, such as machines or certain areas that need to be rented or bought for construction.

For the setting with at least two followers, we need to distinguish between situations depending on how many copies of each project are available. We study the following two extreme situations:

- Limited supply: There is only one copy of each project $v \in R$ available. Notice that every hardness result for this setting automatically extends to the more general setting in which different projects are available in different amounts.
- Unlimited supply: There are sufficiently many copies of the projects available so that, in principle, each project can be sold to all followers. Notice that all copies of the same project need to be sold for the same price. Otherwise, if the leader is allowed to set follower-specific prices, the multifollower problem with unlimited supply essentially reduces to solving one single-follower problem for each follower separately.

1.2. Limited Supply

We model the limited supply situation, in which we restrict to the setting in which only one copy of each project is available as follows. In the first stage of the game, the leader's problem is twofold: the leader has to decide on both (i) an assignment $\boldsymbol{x}$ that assigns each $v \in R$ to one of the followers in $K(v) = \{i \in K \mid v \in R_i\}$ and (ii) prices $\boldsymbol{p} : R \to \mathbb{R}_+$ maximizing the revenue given the constraint that projects $v \in R$ can only be sold to follower $x(v) \in K(v)$. Afterward, in the second stage of the game, based on $\boldsymbol{x}$ and $\boldsymbol{p}$, the followers $i \in K$ simultaneously select a feasible closure $C_i^* \in \mathcal{F}_i(\boldsymbol{x})$, where $\mathcal{F}_i(\boldsymbol{x})$ denotes the set of feasible closures with respect to (w.r.t.) assignment $\boldsymbol{x}$ of maximum value. Thus, the leader's problem in a Stackelberg max closure problem with limited supply reads

$$\pi^*(D) = \max_{\boldsymbol{x}, \boldsymbol{p}} \left\{ \sum_{i \in K} \sum_{v \in C_i^* \cap R} p(v) \mid C_i^* \in \arg\max_{C_i \in \mathcal{F}_i(\boldsymbol{x})} \operatorname{val}(C_i, \boldsymbol{p}) \;\forall i \in K \right\}, \tag{1}$$

where the value of a closure $C_i \in \mathcal{F}_i(\boldsymbol{x})$ is defined as $\operatorname{val}(C_i, \boldsymbol{p}) = \sum_{v \in C_i \cap B_i} w(v) - \sum_{v \in C_i \cap R_i} p(v)$ for all $i \in K$, all $\boldsymbol{x}$, and all $\boldsymbol{p}$.

1.3. Unlimited Supply

In the setting with unlimited supply, in the first stage of the game, the leader assigns prices $\boldsymbol{p} : R \to \mathbb{R}_+$ to all priceable vertices. Afterward, in the second stage of the game, the followers $i \in K$ simultaneously each select a feasible closure $C_i^* \in \mathcal{F}_i$ of maximum value $\operatorname{val}(C_i^*, \boldsymbol{p})$. Thus, the leader's problem in a Stackelberg max closure problem with unlimited supply reads

$$\pi^*(D) = \max_{\boldsymbol{p}} \left\{ \sum_{i \in K} \sum_{v \in C_i^* \cap R} p(v) \mid C_i^* \in \arg\max_{C_i \in \mathcal{F}_i} \operatorname{val}(C_i, \boldsymbol{p}) \;\; \forall i \in K \right\}. \tag{2}$$

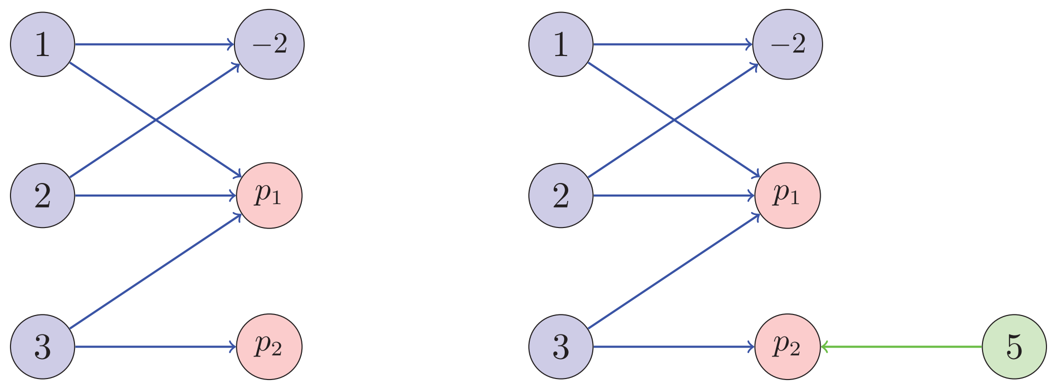
To avoid technicalities, we impose the standard assumption in Stackelberg games that, when each follower faces multiple optimal solutions, ties are broken in favor of the leader.

Example 1. Consider the following two examples illustrated in Figure 1. In the left figure, there is only one follower, and the leader can make a revenue of four by, for example, setting the prices to two on both priceable projects, and this is the best possible. To be more precise, every optimal solution has the following form: $p_1 + p_2 = 4$ with $p_1 \geq 1$ and $p_2 \geq 0$.

In the right figure, there are two followers. How should the leader set the prices so as to maximize revenue?

If we consider the setting in which each priceable project can only be sold to one of the followers, the leader first has to decide to whom to give which priceable project. In this particular example, it would be optimal to assign priceable project 1 to follower 1 for a price of one and priceable project 2 to follower 2 for a price of five,

Figure 1. (Color online) Left: StackMaxClosure with a single follower. Right: StackMaxClosure with two followers.



yielding an optimal revenue of six. Notice that once project 2 is given to follower 2, follower 1 cannot complete the benefit project with a weight of three.

If, instead, two copies of each priceable project are available, the leader can make a revenue of seven by setting $p_1 = 1$ and $p_2 = 3$. For this pricing, follower 1 selects priceable projects 1 and 2 as part of the optimal closure, and follower 2 selects priceable project 2 as part of the optimal closure.

1.4. Connection to StackVC

Given an undirected graph $G = (V, E)$, a subset $C \subseteq V$ is called a *vertex cover* if each edge in E has at least one of its endpoints in C. In a Stackelberg vertex cover (StackVC) with a single follower, the vertex set V is partitioned into red (priceable) and blue (fixed price) vertices $V = R \cup B$ together with fixed weights $\boldsymbol{w} : B \to \mathbb{R}$. Given prices $\boldsymbol{p} : R \to \mathbb{R}_+$, the follower selects a minimum-weight vertex cover.

Both the max closure problem and the min vertex cover problem in bipartite graphs can be solved by means of max-flow computations (König [15], Picard [19]). Briest et al. [7] and Baïou and Barahona [2] show that the same is true for the Stackelberg bipartite vertex cover problem (StackBipartiteVC) with a single follower and the priceable vertices on one side of the partition. We use their algorithm to prove that the Stackelberg max closure problem with a single follower can also be solved in polynomial time (Theorem 1).

1.5. Related Literature

Stackelberg network pricing games were first introduced by Labbé et al. [16], who, motivated by tolling problems in road networks, introduced a two-stage game in which a leader sets tolls so as to maximize revenue after which each of the followers chooses a shortest path. They show that the toll-setting problem can be written as a linear bilevel program and is NP-complete by a reduction from the directed Hamiltonian path problem.

Combinatorial Stackelberg network pricing games are known to be notoriously hard. Most of the results are hardness and/or approximation results. Roch et al. [20] extend the hardness results for Stackelberg games with shortest path followers and prove that the problem with unrestricted prices is NP-hard. Joret [13] shows that the problem is even APX-hard, and Briest et al. [8] give a first explicit approximation threshold. For a more detailed survey on Stackelberg games with shortest path followers, see Van Hoesel [21]. Cardinal et al. [10] prove that the Stackelberg minimum spanning tree problem with a single follower and only two different fixed costs is APX-hard by a reduction from the set cover problem. Briest et al. [7] show that Stackelberg vertex cover in bipartite graphs with priceable vertices on one side, three followers, and unlimited supply is weakly NP-hard by a reduction from the partition problem. The complexity status of the same problem with two followers remained open. We show that this problem is APX-hard. Böhnlein et al. [6] consider Stackelberg packing games instead of covering games and show that the Stackelberg bipartite matching problem is APX-hard.

Roch et al. [20] present a first polynomial-time logarithmic approximation algorithm for Stackelberg games with shortest path followers. Bilò et al. [4] and Cabello [9] give a polynomial time algorithm for Stackelberg shortest path tree games with a fixed number of priceable edges. Balcan et al. [3] and Briest et al. [7] provide a simple logarithmic approximation algorithm for Stackelberg games with shortest path followers. Böhnlein et al. [5] extend the analysis of this simple algorithm beyond the combinatorial setting. Cristi and Schröder [11] study the difference in revenue between positive and unrestricted prices.

Positive results are usually only obtained for particular classes of problems. Briest et al. [7] study Stackelberg vertex cover in bipartite graphs with a single follower and priceable vertices on one side and prove that they can be solved by means of a sequence of maximum flow computations in polynomial time. Later, Baïou and Barahona [2] consider the same problem and propose a different preflow algorithm with improved running time. An interesting open problem, stated in the final remarks of Baïou and Barahona [2], is whether a Stackelberg vertex cover in bipartite graphs with one follower is polynomially solvable, that is, if the priceable vertices are not necessarily on one side of the partition. We answer this question negatively. Thus, although the min vertex cover in bipartite graphs is solvable in strongly polynomial time, its Stackelberg version turns out to be NP-hard even for a single follower.

1.6. Contribution

In Section 2, we show that the algorithm of Briest et al. [7] can be used to solve Stackelberg max closure with a single follower via a sequence of max-flow computations in an auxiliary flow network. Insights of this algorithm and its analysis turn out to be useful for proving the existence of a strongly polynomial-time algorithm for the Stackelberg max closure problem with two followers and limited supply (Theorem 2).

Section 3 is devoted to complexity results. We show that the Stackelberg bipartite vertex cover problem with a single follower is APX-hard (Theorem 4). This way, we answer an open question raised in Baïou and Barahona [2].

Afterward, we show that the Stackelberg max closure problem is APX-hard for two followers and unlimited supply (Theorem 6). Finally, we show that the Stackelberg max closure problem with three followers and limited supply is NP-hard (Theorem 7).

Recall that the Stackelberg max closure problem with two followers and limited supply is polynomial solvable, whereas the Stackelberg max closure problem with two followers and unlimited supply is APX-hard. This may seem counterintuitive as the leader in the limited supply setting has to make two decisions: an assignment of projects to followers and prices of priceable projects, whereas the leader in the unlimited supply setting has to make only one decision: prices. However, because we can solve single follower instances in polynomial time, the solution to the pricing problem in the limited supply setting straightforwardly follows from the assignment of projects. Our results, thus, imply that the assignment problem is computationally easier than the pricing problem for two followers.

2. Efficient Algorithms for StackMaxClosure

In this section, we design and analyze a strongly polynomial time algorithm for StackMaxClosure with two followers and limited supply. We assume that only one copy of each project is available. As a warm-up, we start by describing the algorithm that solves the single-follower Stackelberg max closure problem. The algorithm for the single follower case and its analysis turn out to be useful for the design and analysis of the algorithm for the two-follower setting presented in Section 2.3.

Hochbaum [12] shows that every max closure problem on a digraph D can be transformed into a bipartite digraph with arcs only from benefit to cost projects (by making use of transitivity) without changing the solution. We, thus, may restrict our attention to bipartite instances without loss of generality. So, in the remainder of this paper, we assume that the instance is bipartite with arcs only from benefit to cost projects.

2.1. StackMaxClosure with a Single Follower

It is well-known (Picard [19]) that the max closure problem with underlying digraph $D = (V, A)$ and weights $w(v) \in \mathbb{R}$ for all $v \in V$ can be solved efficiently by computing a min s-t-cut in the auxiliary flow network $\hat{D} = (V \cup \{s, t\}, \hat{A})$ with capacities $\boldsymbol{c} : \hat{A} \to \mathbb{R}_+$ defined as follows. The auxiliary flow network $(\hat{D}, \boldsymbol{c})$ is constructed from instance $(D, \boldsymbol{w})$ according to the following rules:

- A source s and a sink t are added to the vertex set, that is, $\hat{V} = V \cup \{s, t\}$.
- An arc $(u, v) \in \hat{A}$ with capacity $c(u, v) = \infty$ is added for each arc $(u, v) \in A$.
- An arc (s, v) is added to $\hat{A}$ with capacity $c(s, v) = w(v)$ for each $v \in B^+$.
- An arc (v, t) is added to $\hat{A}$ with capacity $c(v, t) = -w(v)$ for each $v \in B^-$.

Observe that, if the capacity of an s-t-cut C in $\hat{D}$ is finite, then it induces a closure $C \setminus \{s\}$. Moreover, the capacity of such an s-t-cut C in $\hat{D}$ is equal to the sum of all benefit projects minus the value of the corresponding closure $C \setminus \{s\}$ in D. Hence, a minimum s-t-cut $C^* \cup \{s\} \subseteq V \cup \{s\}$ in $\hat{D}$ corresponds to an optimal closure C^* in D and vice versa.

Analogue to the algorithm in Briest et al. [7], we use the construction of $\hat{D}$ to describe the algorithm that solves StackMaxClosure with a single follower for any given instance consisting of a digraph $D = (V, A)$ with vertex set partitioned into red and blue vertices $V = R \cup B$ and fixed weights $\boldsymbol{w} : B \to \mathbb{R}$. Because vertices in $v \in R$ are not assigned a fixed weight $w(v)$, we need the additional rule for the construction of $\hat{D}$:

- For each $v \in R$, an arc (v, t) is added to $\hat{A}$ with capacity $c(v, t) = p_v$, where p_v is initially zero.

The algorithm, first introduced and shown to run in polynomial time by Briest et al. [7], proceeds as follows.

Algorithm 1 (Algorithm for StackMaxClosure with a Single Follower)

Construct the flow network $\hat{D}$ as described with $p_v = 0$ for all $v \in R$.
Compute an s-t-flow $\boldsymbol{f}^*$ of max value.
while *there is a vertex* $v \in R$ *such that increasing* p_v *yields an augmenting path* P **do**
  Increase p_v and $\boldsymbol{f}^*$ along P as much as possible
end
return $p_v^* = p_v$ for all $v \in R$

Let $\boldsymbol{f}^*$ denote the maximum flow at termination of Algorithm 1 and $\hat{D}_{f^*} = (\hat{V}, \hat{A}_{f^*})$ be the residual network associated with $\boldsymbol{f}^*$. That is, for each arc $(u, v) \in \hat{A}$, the residual network contains a forward arc $(u, v) \in \hat{A}_{f^*}$ if the residual capacity $c^*(u, v) := c(u, v) - f^*(u, v)$ is positive, and whenever $f^*(u, v) > 0$, a backward arc (v, u) with residual capacity $c^*(v, u) := f^*(u, v)$. Remember that the set of all vertices from which t is not reachable in the residual network $\hat{D}_{f^*}$, that is, $C^* \cup \{s\} = \{v \in \hat{V} \mid \exists \text{ no v-t-path in } \hat{D}_{f^*}\}$, is an s-t-cut of min capacity in $\hat{D}$ and C^* is an optimal closure in D for capacities corresponding to prices $\boldsymbol{p}^*$. Even though C^* need not be the unique optimal closure with respect to $\boldsymbol{p}^*$, it follows from our tie-breaking rule being in favor of the leader that the follower always selects an optimal closure, such as C^*, containing all priceable projects in R. In the remainder, we assume that the follower picks the closure C^*.

Theorem 1 (cf. Briest et al. [7]). *Algorithm 1 returns optimal prices $\boldsymbol{p}^*$ for StackMaxClosure with a single follower. Moreover, all priceable vertices belong to the optimal closure C^* selected by the follower, and the leader's revenue is equal to $\pi^*(D) = c_\infty - c_0$, where c_∞ and c_0 denote the value of the maximum flow in $\hat{D}$ if $c(v, t) = \infty$ for all $v \in R$ and $c(v, t) = 0$ for all $v \in R$, respectively.*

Observe that $c_\infty - c_0$ denotes the difference in revenue of the follower when all priced projects are unavailable and when all priced projects are available for free. The proof of Theorem 1 goes along the same lines as in Briest et al. [7] and can be found in the appendix.

2.2. Structural Properties

To solve the Stackelberg max closure problem for two followers in the setting in which only one copy of each project is available, we need a couple of structural properties of the optimal closure C^* selected by the follower and the revenue obtained by the leader.

For the proof of our main Theorem 2, we need Lemma 1, which characterizes the optimal revenue of the leader. In order to do so, we need some additional notation. Given a subset of fixed-price projects $S \subseteq B$, let us denote by $f(S)$ the value of the maximum flow in the auxiliary network $\hat{D}$ restricted to the subgraph induced by $S \cup \{s, t\}$, and for every $T \subseteq V$, let $N(T) = \{v \in V \mid (v, u) \text{ or } (u, v) \text{ in } A \text{ for any } u \in T\}$ denote the set of all neighbors of T in D. Moreover, for $T \subseteq V$, let $w(T) := \sum_{v \in T} w(v)$.

Lemma 1. *Let C^* be the optimal closure in D (as defined in Section 2.1). Then,*

$$\pi^*(D) = w(C^* \cap B^+ \cap N(R)) + f(C^* \cap ((B^+ \setminus N(R)) \cup B^-)) - f(C^* \cap B).$$

Proof. The value of the maximum flow in $\hat{D}$ if $c(v, t) = \infty$ for all $v \in R$ is equal to

$$c_\infty = w(B^+ \cap N(R)) + f((B^+ \setminus N(R)) \cup B^-).$$

The value of the maximum flow in $\hat{D}$ if $c(v, t) = 0$ for all $v \in R$ is equal to

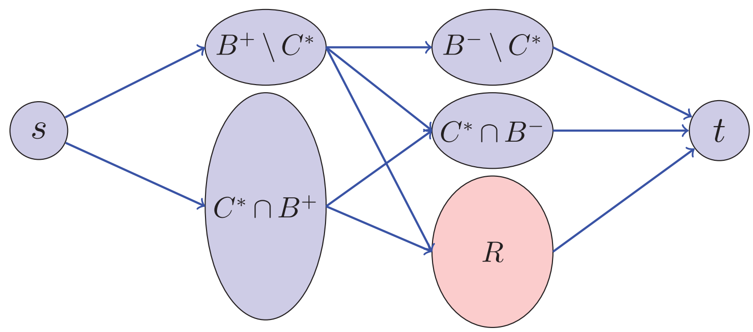
$$c_0 = f(B).$$

By Theorem 1,

$$\pi^*(D) = c_\infty - c_0 = w(B^+ \cap N(R)) + f((B^+ \setminus N(R)) \cup B^-) - f(B).$$

To see that we can restrict attention to vertices in the optimal closure C^*, notice that, at the end of Algorithm 1, no flow is sent from a benefit project $u \in B^+ \setminus C^*$ to a cost project $v \in C^* \cap B^-$ nor to a priceable project $v \in R$. See Figure 2. If there were such flow, there would be a path from project v to t in the final residual network $\hat{D}_{f^*}$ by

Figure 2. (Color online) The auxiliary flow network $\hat{D}$.



using the arc to u and continuing along the u-t path, and hence, v would not belong to $C^* \cup \{s\} = \{v \in \hat{V} \mid \exists \text{ no v-t-path in } \hat{D}_{f^*}\}$: a contradiction to $v \in C^*$.

The previous observation shows that, even if we restrict attention to the subgraph $(B^+ \setminus C^*) \cup B^- \cup \{s,t\}$, which does not include R, the maximum flow can all be sent from $B^+ \setminus C^*$ to $B^- \setminus C^*$. This implies that also, if prices are zero, we send $f((B^+ \setminus C^*) \cup B^-)$ flow units from $B^+ \setminus C^*$ to $B^- \setminus C^*$. Because $\pi^*(D) = c_\infty - c_0$, the flow that is sent from $B^+ \setminus C^*$ to $B^- \setminus C^*$ cancels out and does not impact the optimal revenue. Hence, the desired result follows. □

2.3. StackMaxClosure with Two Followers and Limited Supply

We now consider StackMaxClosure with two followers and limited supply. That is, each priceable project can only be bought by one of the followers. In this setting, the leader's problem is twofold: (i) an assignment $\boldsymbol{x}: R_1 \cap R_2 \to \{1,2\}$ of the shared priceable projects to the two followers and (ii) prices $\boldsymbol{p}: R_1 \cup R_2 \to \mathbb{R}_+$ maximizing the revenue given the constraint that projects $v \in R_1 \cap R_2$ can only be sold to follower $x(v) \in \{1,2\}$.

We show that we can find an optimal solution in strongly polynomial time. Consider an instance of our problem with two followers $i \in \{1,2\}$ with underlying digraphs $D_i = (V_i, A_i)$, the vertex sets $V_i = R_i \cup B_i$ being partitioned into priceable (red) and fixed price (blue) vertices, and fixed prices $\boldsymbol{w}: B_1 \cup B_2 \to \mathbb{R}$. Recall that only priceable vertices can be shared, that is, $B_1 \cap B_2 = \emptyset$. We may split the fixed price vertices into benefit projects $B_i^+ = \{v \in B_i \mid w(v) > 0\}$ and cost projects $B_i^- = B_i \setminus B_i^+$, $i \in \{1,2\}$.

2.3.1. Finding Optimal Prices for a Given Assignment *x*. We first note that the leader's problem essentially reduces to the problem of finding an optimal assignment $\boldsymbol{x}: R_1 \cap R_2 \to \{1,2\}$. Note that any assignment $\boldsymbol{x}: R_1 \cap R_2 \to \{1,2\}$ partitions the instance given by the two digraphs $D = (D_1, D_2)$ and weights $\boldsymbol{w}$ into two single-follower instances $D_i(\boldsymbol{x}) = D_i[R_i(\boldsymbol{x}) \cup B_i(\boldsymbol{x})]$, where $R_i(\boldsymbol{x}) = \{v \in R_i \mid \text{ if } v \in R_1 \cap R_2 \text{ then } x(v) = i\}$ and $B_i(\boldsymbol{x}) = \{v \in B_i \mid \text{ for all } u \in R_1 \cap R_2,$ if $x(u) \neq i$ then $(v,u) \neq A_i\}$, $i \in \{1,2\}$. Given an assignment $\boldsymbol{x}$, let us denote by $\pi^*(D,\boldsymbol{x})$ the optimal revenue of the leader in D with respect to weights $\boldsymbol{w}$ and assignment $\boldsymbol{x}$. Observe that $\pi^*(D,\boldsymbol{x})$ can be written as $\pi^*(D,\boldsymbol{x}) = \pi_1^*(D,\boldsymbol{x}) + \pi_2^*(D,\boldsymbol{x})$. Because, for a given assignment $\boldsymbol{x}$, the optimal price vector $\boldsymbol{p}^*$ and the associated max closures $C_i^*(\boldsymbol{x})$, $i = 1,2$, can be found in strongly polynomial time by Algorithm 1, optimal prices for a given assignment $\boldsymbol{x}$ can be found in strongly polynomial time.

2.3.2. The Easy Case: No Fixed Price Cost Projects. Let us first consider a simple special case of our problem without any fixed-price cost projects, that is, where $B_1^- = B_2^- = \emptyset$. Here, the leader's problem can be solved in strongly polynomial time via a reduction to MAX INDEPENDENT SET IN BIPARTITE GRAPHS. Because, for a given graph $G = (V,E)$, a vertex set $S \subseteq V$ is independent if and only if $V \setminus S$ is a vertex cover, the problem of finding a max independent set is equivalent to the problem of finding a min vertex cover. The latter problem in bipartite graphs can be solved, again, via a min s-t-cut computation in some auxiliary flow network (König [15]).

Lemma 2. *If $B_1^- = B_2^- = \emptyset$, the leader's problem can be polynomially reduced to* MAX INDEPENDENT SET IN BIPARTITE GRAPHS.

Proof. As described, any assignment $\boldsymbol{x}: R_1 \cap R_2 \to \{1,2\}$ partitions the instance into two single-follower Stackelberg max closure problems that can be solved separately. Recall that $\pi_i^*(D,\boldsymbol{x})$ denotes the maximum revenue achievable for the leader in the single-follower instance $D_i(\boldsymbol{x})$ resulting from assignment $\boldsymbol{x}$ for $i \in \{1,2\}$. Note that, in the special case in which all cost projects are priceable, $\pi_i^*(D,\boldsymbol{x})$ for $i \in \{1,2\}$ is equal to the overall weight of all benefit projects in B_i^+, which do not depend on any priceable project assigned to the other follower, that is,

$$\pi_i^*(D,\boldsymbol{x}) = \sum \{w(v) \mid v \in B_i^+ \setminus N_i(R_j(\boldsymbol{x}))\},$$

where $i \neq j$ and $N_i(S) = \{v \in V_i \mid (v,u) \text{ or } (u,v) \text{ in } A_i \text{ for any } u \in S\}$ denotes the set of all neighbors of any vertex set $S \subseteq V_i$ in D_i. As a consequence, the leader's problem for the case in which $B_1^- = B_2^- = \emptyset$ reduces to the problem of finding an optimal assignment $\boldsymbol{x}$ maximizing

$$\sum\{w(v) \mid v \in B_1^+ \setminus N_1(R_2(\boldsymbol{x}))\} + \sum\{w(v) \mid v \in B_2^+ \setminus N_2(R_1(\boldsymbol{x}))\}. \tag{3}$$

Such an optimal assignment $\boldsymbol{x}$ can be found by finding a max independent set in the undirected bipartite graph $G = (B_1^+ \cup B_2^+, E)$ on vertex set $B_1^+ \cup B_2^+$ and edges $\{u,v\} \in E$ for any pair $(u,v) \in B_1^+ \times B_2^+$ having a common neighbor in $R_1 \cap R_2$. Note that any max independent set S^* in G w.r.t. the given fixed weights $\boldsymbol{w}: B_1^+ \cup B_2^+ \to \mathbb{R}_+$ corresponds to an optimal solution $\boldsymbol{x}^*$ of the leader's problem with revenue $\pi^*(D, \boldsymbol{x}^*) = w(S^*)$ for the following reason. By construction, whenever two benefit projects $u \in B_1^+$ and $v \in B_2^+$ share a priceable project as a neighbor, only one of $\{u, v\}$ is selected into S^*. The priceable projects associated with edge $\{u, v\}$ are then assigned to the follower whose endpoint is selected into S^*, resulting in assignment $\boldsymbol{x}^*$. Because the revenue is equal to the overall weight of all benefit projects assigned to each follower, the result follows. □

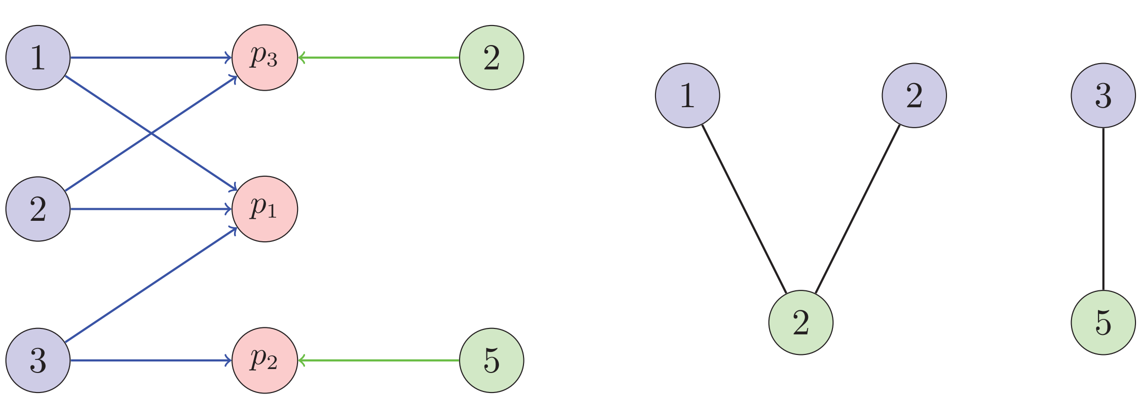
Example 2. For an example, see Figure 3. In the left figure, we see the StackMaxClosure problem with two followers. Follower 1 corresponds to the blue–red graph, whereas follower 2 corresponds to the green–red graph. In the right figure, we see the corresponding max independent set problem. The benefit project of value one, two, respectively, of follower 1 is linked to the benefit project of value two of follower 2 because all depend on priceable project 3. The benefit project of value three of follower 1 is linked to the benefit project of value five of follower 2 because both depend on priceable project 2. The optimal revenue is, therefore, eight, equal to the max independent set in the right figure because priceable project p_1 can always be given to follower 1, whereas it is optimal to give priceable project p_2 to follower 2 and priceable project p_3 to follower 1, implying that a revenue of three is derived from follower 1 and a revenue of five is derived from follower 2.

2.3.3. The General Case. We solve the leader's problem in the general two-follower case by a reduction to a problem of the previously mentioned easy case. That is, given any instance consisting of two digraphs $D_i = (R_i \cup B_i, A_i)$, $i \in \{1,2\}$ and weights $\boldsymbol{w}: B_1 \cup B_2 \to \mathbb{R}$, where B_1^- and B_2^- are not necessarily empty, we construct an auxiliary instance consisting of two digraphs $D_i' = (R_i' \cup B_i', A_i')$ and weights $\boldsymbol{w}': B_1' \cup B_2' \to \mathbb{R}$ in which there are no fixed-price cost projects and that satisfies the property that an optimal assignment $\boldsymbol{x}'$ corresponds to an optimal assignment $\boldsymbol{x}$ of the original instance. Because an optimal assignment $\boldsymbol{x}'$ of the easy instance given by D_1', D_2', and $\boldsymbol{w}'$ can be found efficiently (Lemma 2), we obtain the following main result of our paper.

Theorem 2. *StackMaxClosure with two followers in the setting in which only one copy of each project is available can be solved in strongly polynomial time.*

We prove Theorem 2 by showing that, for any instance $(D, \boldsymbol{w})$ given by $D_i = (R_i \cup B_i, A_i)$, $i \in \{1,2\}$ and weights $\boldsymbol{w}: B_1 \cup B_2 \to \mathbb{R}$, the problem of finding an assignment $\boldsymbol{x}: R_1 \cap R_2 \to \{1,2\}$ maximizing the leader's revenue $\pi^*(D, \boldsymbol{x}) = \pi_1^*(D_1, \boldsymbol{x}) + \pi_2^*(D_2, \boldsymbol{x})$ reduces to the problem of finding an optimal assignment $\boldsymbol{x}'$ in the auxiliary "easy" instance $(D', \boldsymbol{w}')$ constructed as follows.

Figure 3. (Color online) Left: StackMaxClosure with limited supply and two followers. Right: The corresponding max independent set problem.



2.3.3.1. Construction of Auxiliary "Easy" Instance. The auxiliary instance $(D', \boldsymbol{w}')$ associated with $(D, \boldsymbol{w})$ consists of two digraphs $D'_i = (R'_i \cup B'_i, A'_i)$ with priceable (red) vertices R'_i and fixed price (blue) vertices B'_i, and vertex weights $\boldsymbol{w}' : B'_1 \cup B'_2 \to \mathbb{R}$ constructed according to the following rules. For $i \in \{1,2\}$ and $j \neq i$ define

- $R'_i = R_i \cup B_i^- \cup B_j^-$.
- $B'_i = B_i^+ \cup \tilde{B}_j^-$, where $\tilde{B}_j^- = \{\tilde{v} \mid v \in B_j^-\}$ is a copy of the fixed cost projects of follower j.
- $A'_i = A_i \cup \{(\tilde{v}, v) \mid v \in B_j^-\}$.
- $w'(v) = \begin{cases} w(v) & \text{if } v \in B_i^+ \\ -w(v) & \text{if } \tilde{v} \in \tilde{B}_j^- \end{cases}$ for all $v \in B'_i$.

That is, we transform every fixed-price cost project into a priceable project, and additionally, we make a copy $\tilde{v}$ of each originally fixed-price cost and now priceable project $v \in B_i^-$, which now becomes a benefit project for the other follower $j \neq i$ of new weight $w'(\tilde{v}) = -w(v)$, and we add an arc $(\tilde{v}, v)$ to A'_j. See Figure 4 for an illustration.

The intuition behind the equivalence of an optimal assignment in the original instance and an optimal assignment in the auxiliary instance is that the difference in optimal revenue is equal to the sum of all weights of the cost projects. A cost project that is now priceable is either given to the same follower, implying that the costs are now extracted by the leader, or given to the other follower, implying an increase in revenue of the leader via the copy. However, this equivalence is only valid after some preprocessing as, otherwise, the leader can also extract the value of an optimal closure that is not linked to priceable vertices and, thus, is always selected by the follower. We now prove this formally.

Definition 1. Given an assignment $\boldsymbol{x}$, we call an assignment $\boldsymbol{x}'$ in D' an *extension of* $\boldsymbol{x}$ if $x'(v) = x(v)$ for all $v \in R_1 \cap R_2$.

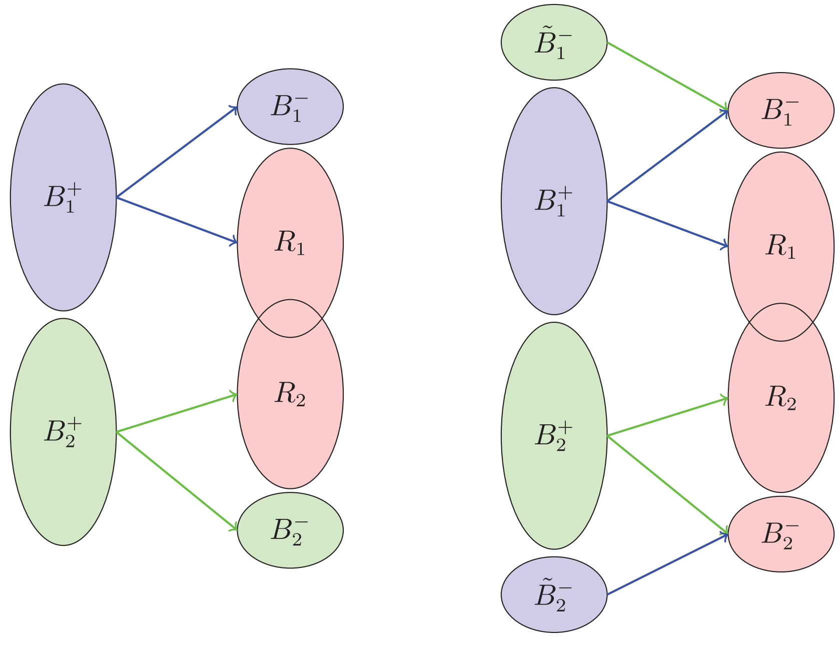
Similar to before, an extension $\boldsymbol{x}'$ partitions D' into $D'_1(\boldsymbol{x}')$ and $D'_2(\boldsymbol{x}')$, and we denote by $\pi^*(D', \boldsymbol{x}')$ the optimal revenue of the leader in D' with respect to $\boldsymbol{x}'$. Before constructing the auxiliary instance $(D', \boldsymbol{w}')$, we perform the following preprocessing step on $(D, \boldsymbol{w})$.

2.3.3.2. Preprocessing. Some projects are always selected by a follower $i \in \{1,2\}$. If $S \subseteq B_i^+$ satisfies

$$S \cap N_i(R_i) = \emptyset \quad \text{and} \quad w(S) > c(N_i(S)), \tag{4}$$

then $S \cup N(S)$ is guaranteed to be selected into a max closure of follower i, completely independent of the assignment $\boldsymbol{x}$ and the pricing $\boldsymbol{p} : R_1 \cup R_2 \to \mathbb{R}_+$ chosen by the leader. In a preprocessing step, we compute for both followers $i \in \{1,2\}$ a max closure C_i^* in the subgraph $D_i[(B_i \setminus N(R_i))]$ restricted to fixed-price vertices not dependent on any priceable project. In case $w(C_i^*) > 0$, we remove all vertices in C_i^* from D_i as they are selected by follower i

Figure 4. (Color online) Left: The original instance D. Right: The auxiliary instance D'.



anyway. This way, after preprocessing, we can assume without loss of generality that there is no set $S \subseteq B_i^+$ satisfying property Equation (4). Lemma 4 relies on this assumption.

Lemma 3. *For every assignment $\boldsymbol{x}$ in D, there exists an extension $\boldsymbol{x}'$ of $\boldsymbol{x}$ in D' such that $\pi^*(D',\boldsymbol{x}') \geq \pi^*(D,\boldsymbol{x}) + c(B_1^-) + c(B_2^-)$.*

Proof. Let $C_i^*(\boldsymbol{x})$ be the optimal project selection of follower $i = 1, 2$ in D with respect to assignment $\boldsymbol{x}$. Let $S_i^*(\boldsymbol{x}) = C_i^*(\boldsymbol{x}) \cap B_i^-$ denote the set of fixed-cost projects selected by follower i. Consider the extension $\boldsymbol{x}'$ of $\boldsymbol{x}$ that assigns a new priceable project $v \in B_i^-$ to follower i exactly if $v \in S_i^*(\boldsymbol{x})$. If $v \in B_i^- \setminus S_i^*(\boldsymbol{x})$, then v is assigned to the other follower $j \neq i$. Given such an assignment $\boldsymbol{x}'$, we may set prices equal to the original costs, that is, set $p(v) = -w(v)$ for all $v \in B_1^- \cup B_2^-$. This assignment $\boldsymbol{x}'$ together with such prices yields a revenue of $\pi^*(D,\boldsymbol{x}) + c(B_1^-) + c(B_2^-)$, and thus, the optimal revenue of $\boldsymbol{x}'$ is at least as high. □

Lemma 4. *After preprocessing, for every assignment $\boldsymbol{x}$ in D and any extension $\boldsymbol{x}'$ of $\boldsymbol{x}$ in D', we have $\pi^*(D',\boldsymbol{x}') \leq \pi^*(D,\boldsymbol{x}) + c(B_1^-) + c(B_2^-)$.*

Proof. Let $\boldsymbol{x}$ be an assignment in D and $\boldsymbol{x}'$ be an extension of $\boldsymbol{x}$ in D'. Let $C_i^*(\boldsymbol{x})$ be the optimal project selection in D with respect to assignment $\boldsymbol{x}$, and let $C_i^*(\boldsymbol{x}')$ be the optimal project selection in D' with respect to assignment $\boldsymbol{x}'$ of follower $i \in \{1,2\}$. Recall that $\pi^*(D',\boldsymbol{x}') = \pi_1^*(D',\boldsymbol{x}') + \pi_2^*(D',\boldsymbol{x}')$. Observe that each priceable project $v \in B_i^-$ in D' for $i \neq j$ that is assigned to follower j yields a revenue of $-w(v)$. Because all priceable vertices in $D_i'(\boldsymbol{x}')$ belong to $C_i^*(\boldsymbol{x}')$ (cf. Theorem 1), it follows that these vertices are exactly the vertices in $B_i^- \setminus C_i^*(\boldsymbol{x}')$. That is, we have $B_1^- \setminus C_1^*(\boldsymbol{x}') = \{v \in B_1^- \mid \boldsymbol{x}'(v) = 2\}$ and $B_2^- \setminus C_2^*(\boldsymbol{x}') = \{v \in B_2^- \mid \boldsymbol{x}'(v) = 1\}$. Let $\bar{D}_1(\boldsymbol{x}')$ denote the subgraph obtained from $D_1'(\boldsymbol{x}')$ by removing all vertices in $B_2^- \setminus C_2^*(\boldsymbol{x}')$ and their copies $\{\tilde{v} \mid v \in B_2 \setminus C_2^*(\boldsymbol{x}')\}$. Define $\bar{D}_2(\boldsymbol{x}')$ similarly. It follows that $\pi^*(D',\boldsymbol{x}') = \pi_1^*(\bar{D}_1,\boldsymbol{x}') + c(B_2^- \setminus C_2^*(\boldsymbol{x}')) + \pi_2^*(\bar{D}_2,\boldsymbol{x}') + c(B_1^- \setminus C_1^*(\boldsymbol{x}'))$. Thus, to prove Lemma 4, it suffices to show that

$$\pi_i^*(\bar{D}_i,\boldsymbol{x}') \leq \pi_i^*(D,\boldsymbol{x}) + c(C_i^*(\boldsymbol{x}') \cap B_i^-) \tag{5}$$

for $i \in \{1,2\}$.

To shorten notation, let $C^* := C_i^*(\boldsymbol{x})$ and $C' := C_i^*(\boldsymbol{x}')$. Moreover, let $C_+^* = C^* \cap B_i^+$, $C_+' = C' \cap B_i^+$, $C_-^* = C^* \cap B_i^-$, $C_-' = C' \cap B_i^-$ and $R = R_i(\boldsymbol{x})$.

Recall from Section 2.2 that, for a subset S of fix price projects, $f(S)$ denotes the max flow value in the auxiliary flow network restricted to the subgraph induced by $S \cup \{s,t\}$. By Lemma 1, we know that

$$\pi_i^*(\bar{D}_i,x) = w(C_+^* \cap N(R)) + f((C_+^* \setminus N(R)) \cup C_-^*) - f(C_+^* \cup C_-^*). \tag{6}$$

Recall from Theorem 1 that, in order to compute $\pi_i^*(\bar{D}_i,\boldsymbol{x}')$, we need to consider the auxiliary flow network w.r.t. the single-follower instance $\bar{D}_i(\boldsymbol{x}')$ and compare the max flow value for the case in which $c(v,t) = \infty$ for all $v \in R_i(\boldsymbol{x}')$ with the max flow value for the case in which $c(v,t) = 0$ for all $v \in R_i(\boldsymbol{x}')$. Observe that

$$\pi_i^*(\bar{D}_i,\boldsymbol{x}') \leq w(C_+'), \tag{7}$$

and from Equations (6) and (7), we observe that, in order to prove Equation (5), it suffices to show

$$w(C_+') + f(C_+^* \cup C_-^*) \leq w(C_+^* \cap N(R)) + f((C_+^* \setminus N(R)) \cup C_-^*) + c(C_-'). \tag{8}$$

Let f^* be a max flow in the auxiliary flow network induced by the projects in $C_+^* \cup C_-^*$. We partition the value $f(C_+^* \cup C_-^*)$ of flow f^* into

$$f(C_+^* \cup C_-^*) = v_1 + v_2 + v_3,$$

where v_1 denotes the value of flow f^* sent through the benefit projects in $(C_+^* \setminus C_+') \setminus N(R)$, v_2 denotes the value of flow f^* sent through the benefit projects in $(C_+^* \setminus C_+') \cap N(R)$, and v_3 denotes the value of flow f^* sent through the benefit projects in $C_+^* \cap C_+'$.

Note that, by our preprocessing, we know that $w(C_+^* \setminus N(R)) \leq c(C_-^*)$. Thus,

$$f((C_+^* \setminus N(R)) \cup C_-^*) = w(C_+^* \setminus N(R)) \geq v_1 + w((C_+^* \cap C_+') \setminus N(R)), \tag{9}$$

where the inequality follows by definition of v_1.

By Equation (9), in order to show Equation (8), it suffices to prove that

$$w(C_+' \setminus C_+^*) + w(C_+^* \cap C_+' \cap N(R)) + v_2 + v_3 \leq w(C_+^* \cap N(R)) + c(C_-'). \tag{10}$$

Note that $v_2 \leq w((C_+^* \setminus C_+') \cap N(R))$. Hence, it suffices to prove

$w(C'_+ \setminus C^*_+) + w(C^*_+ \cap C'_+ \cap N(R)) + v_3 \le w(C^*_+ \cap C'_+ \cap N(R)) + c(C'_-)$, which is equivalent to

$$w(C'_+ \setminus C^*_+) + v_3 \le c(C'_-). \tag{11}$$

Finally, we observe that

$$v_3 \le c(C^*_- \cap C'_-), \tag{12}$$

for the following reason. Because C^* and C' are closures, there are no arcs from $C^*_+ \cap C'_+$ to $C'_- \setminus C^*_-$ and no arcs from $C^*_+ \cap C'_+$ to $C^*_- \setminus C'_-$, implying that our flow f^* needs to send all flow that goes through benefit projects in $C^*_+ \cap C'_+$ through cost projects in $C^*_- \cap C'_-$. Hence, the value of that flow, namely v_3, is upper bounded by the capacity $c(C^*_- \cap C'_-)$. Summarizing, we derived that Equation (5) and, thus, the statement of the lemma follows if we can show

$$w(C'_+ \setminus C^*_+) \le c(C'_- \setminus C^*_-). \tag{13}$$

However, Equation (13) must be true as, otherwise, $C' \cup C^*$ is a closure in D of larger value than C^* in contradiction to the optimality of C^*. □

Proof of Theorem 2. It follows from our considerations that an optimal assignment $\boldsymbol{x}$ for $(D, \boldsymbol{w})$ can be found as follows: after preprocessing, we construct the auxiliary easy instance $(D', \boldsymbol{w}')$ as described and compute an optimal assignment $\boldsymbol{x}'$ for $(D', \boldsymbol{w}')$ via reduction to a bipartite max stable set problem. By Lemmas 3 and 4, we know that there exists an extension $\boldsymbol{x}'$ such that

$$\pi^*(D', \boldsymbol{x}') = \pi^*(D, \boldsymbol{x}) + c(B_1^-) + c(B_2^-).$$

Because $c(B_1^-) + c(B_2^-)$ is a constant, that is, independent of assignment $\boldsymbol{x} : R_1 \cap R_2 \to \{1, 2\}$, we obtain an optimal assignment $\boldsymbol{x}$ from $\boldsymbol{x}'$ by setting $x(v) = x'(v)$ for all $v \in R_1 \cap R_2$. □

Remark 1. The result of Theorem 2 can be extended to digraphs that also intersect in benefit projects. In that case, we extend the construction of the max independent set problem by adding edges between two benefit projects that are in both digraphs, and then, the same arguments apply.

3. Complexity Results

Recall that Briest et al. [7] show that StackBipartiteVC with a single follower with priceable vertices on one side of the partition is solvable in time polynomial in $|V|$ and $|E|$. Baïou and Barahona [2] strengthen this result by describing a faster algorithm to compute the optimal prices. In the following section, we show that StackBipartiteVC with a single follower in general, that is, for which priceable vertices can be on both sides of the partition, is APX-hard. Afterward, in Section 3.2, we show that StackMaxClosure with two followers and unlimited supply is APX-hard. Subsequently, in Section 3.3, we show that StackMaxClosure with three followers in the setting in which only one copy per project is available is NP-hard.

3.1. StackBipartiteVC with a Single Follower

The main result of this section shows that StackBipartiteVC with a single follower is NP-complete and even APX-complete. This answers the open problem proposed by Baïou and Barahona [2]. We use a reduction from the NP-hard problem maximum independent set (Karp [14]). Recall that Maximum Independent Set takes as input an undirected graph $G = (V, E)$ and asks for a subset $S \subseteq V$ of the vertices of maximum cardinality such that no two vertices in S are linked by an edge.

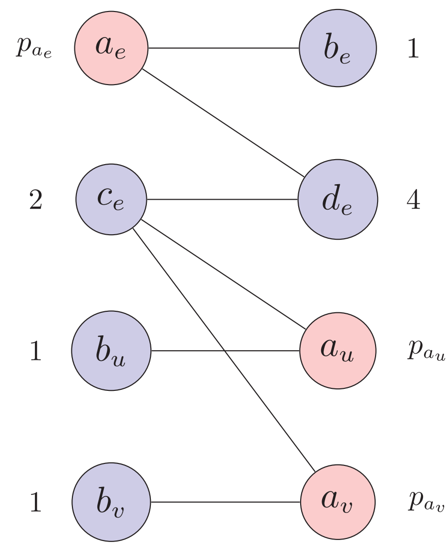
Theorem 3. *StackBipartiteVC with a single follower is NP-hard.*

Proof. We show that the decision variant of StackBipartiteVC with a single follower is NP-complete. Clearly, the problem is in NP because, for any given target revenue r, vertex set $C \subseteq V$, and price vector $\boldsymbol{p} : R \to \mathbb{R}_+$, we can check in polynomial time whether C is indeed a vertex cover with $\sum_{C \cap R} p(v) \ge r$. To see that the problem is NP-complete, consider any instance $G = (V, E)$ of Maximum Independent Set and construct a bipartite graph $G' = (U \cup W, E')$ as follows:

a. For each vertex $v \in V$, create a priceable vertex $a_v \in W$, a fixed-price vertex $b_v \in U$ with weight $w(b_v) = 1$, and an edge $\{b_v, a_v\} \in E'$.

b. For each edge $e = \{u, v\} \in E$, create a priceable vertex $a_e \in U$, a fixed-price vertex $b_e \in W$ with weight $w(b_e) = 1$, a fixed-price vertex $c_e \in U$ with weight $w(c_e) = 2$, a fixed-price vertex $d_e \in W$ with weight $w(d_e) = 4$, an edge $\{a_e, b_e\} \in E'$, an edge $\{a_e, d_e\} \in E'$, an edge $\{c_e, d_e\} \in E'$, an edge $\{c_e, a_u\} \in E'$, and an edge $\{c_e, a_v\} \in E'$. See Figure 5.

Figure 5. (Color online) Gadget corresponding to edge $e = \{u, v\} \in E$.



We show that, for every optimal pricing strategy $\boldsymbol{p}^*$ of the leader with associated min vertex cover $C^* = C^*(\boldsymbol{p}^*)$ in G', the vertices in $S^* := \{v \in V \mid a_v \in C^*\}$ form a maximum independent set in G and vice versa.

We start by showing that, for any independent set $S \subseteq V$ in G, there exists a price strategy p^S for the leader such that the follower selects among the vertices in $\{a_v \mid v \in V\}$ exactly those corresponding to S and for which the leader receives a revenue of $5 \cdot |E| + |S|$. Consider an arbitrary independent set $S \subseteq V$ in G, and set prices as follows: prices on the vertices in $\{a_v \mid v \in S\}$ are set to one, and prices on all vertices in $\{a_v \mid v \in V \setminus S\}$ are set to infinity. This way, the leader forces the follower to select all vertices corresponding to the independent set S into the vertex cover (assuming that the follower breaks ties in favor of the leader). Moreover, because for every edge $e \in E$, at least one endpoint belongs to $V \setminus S$, the follower necessarily selects all vertices $\{c_e \mid e \in E\}$ into the optimal cover, independent of the prices on the vertices in $\{a_e \mid e \in E\}$. Note that the follower selects a_e into the optimal vertex cover if and only if the price on a_e does not exceed five. Otherwise, if the price on a_e is larger than five, the follower selects d_e and b_e for a price of five into the cover and leaves out a_e. Thus, by setting the prices on all vertices in $\{a_e \mid e \in E\}$ to five, the leader receives a revenue of $5\,|E| + |S|$ using this pricing strategy.

Conversely, let $\boldsymbol{p}^*$ denote an arbitrary optimal price vector of the leader and C^* be the corresponding min vertex cover chosen by the follower with ties broken in favor of the leader. Suppose, for the sake of contradiction, $S^* := \{v \in V \mid a_v \in C^*\}$ is not an independent set. Let $E' \subseteq E$ be the subset of edges for which both endpoints are in S^* and $V[E'] \subseteq S^*$ be the set of endpoints of the edges in E'. The set E' is not empty because of our assumption of S^* being not an independent set. We partition the edges in E' into two disjoint sets $E' = E'_1 \cup E'_2$ so that E'_1 contains all edges $e \in E'$ with $c_e \notin C^*$, and $E'_2 = E' \setminus E'_1$ contains those edges in E' with $c_e \in C^*$. Note that $\sum_{v \in V[E']} p^*(a_v) \leq |V[E']| + 2\,|E'_1|$. Otherwise, if $\sum_{v \in V[E']} p^*(a_v) > |V[E']| + 2\,|E'_1|$, the follower could improve by switching to cover $C^* \setminus \{a_v \mid v \in V[E']\} \cup \{b_v \mid v \in V[E']\} \cup \{c_e \mid e \in E'_1\}$.

For each $e \in E'_2$, vertex c_e belongs to C^*. Thus, a_e belongs to C^* if and only if the price on a_e does not exceed three because, for $p^*(a_e) > 3$, the follower could improve by switching to vertex cover $C^* \setminus \{c_e, a_e\} \cup \{d_e, b_e\}$. Therefore, we have $p^*(a_e) = 3$, implying that the leader receives a revenue of three from each vertex a_e with $e \in E'_2$.

For each edge $e \in E'_1$, vertex c_e does not belong to C^*. Thus, a_e is not selected by the follower for the following reason: the follower selects a_e only if $p^*(a_e) \leq 1$. But, if $p^*(a_e) \leq 1$, the follower could improve by switching to $C^* \setminus \{d_e, b_e\} \cup \{c_e, a_e\}$. Therefore, the leader receives no profit from a_e for every edge $e \in E'_1$.

Let $U \subseteq V[E'] \subseteq S^*$ be a subset of vertices of size at most $|E'|$ such that each edge of E' contains at least one endvertex in U. Note that such a cover can easily be constructed by selecting one endvertex of each edge in E' arbitrarily. We derive the desired contradiction by observing that the leader can improve the revenue as follows: the leader raises the prices on the vertices in $\{a_v \mid v \in U\}$ to infinity, sets the prices on all vertices in $\{a_v \mid v \in V[E'] \setminus U\}$ to one, and raises the prices on all vertices in $\{a_e \mid e \in E'\}$ to five. Under the new price vector, an optimal vertex cover for the follower is $C^* \setminus \{\{a_v \mid v \in U\} \cup \{d_e, b_e \mid e \in E'_1\}\} \cup \{b_v \mid v \in U\} \cup \{c_e, a_e \mid e \in E'_1\}$.

Let's have a closer look at the change in the leader's revenue under this new pricing strategy: Because $\sum_{v \in V[E']} p^*(a_v) \leq |V[E']| + 2\,|E'_1|$, the leader's revenue obtained from vertices in $\{a_v \mid v \in V\}$ might decrease but at most by $|V[E']| + 2\,|E'_1| - |V[E'] \setminus U| = |U| + 2\,|E'_1|$. However, the leader's revenue obtained from vertices in $\{a_e \mid e \in E\}$ increases by the amount $2\,|E'_2| + 5\,|E'_1|$. Because $|U| \leq |E'| = |E'_1| + |E_2|$, it follows that the leader's revenue increases by the positive amount of at least $2 \cdot |E'_2| + 5\,|E'_1| - |U| - 2\,|E'_1| \geq |E'| > 0$: a contradiction to the optimality of p^*.

Summarizing, for each maximum independent set S^* in G exists a price vector $\boldsymbol{p}^*$ of revenue $5\,|E| + |S^*|$. Conversely, for each optimal pricing strategy $\boldsymbol{p}^*$ with corresponding vertex cover C^* in G', the set $S^* := \{a_v \mid v \in C^*\}$ forms an independent set in G, and the revenue of the leader is $5\,|E| + |S^*|$. Hence, the price vector maximizes the leader's revenue if and only if S^* is a maximum independent set. □

APX is the class of optimization problems in NP that admit a constant-factor polynomial-time approximation algorithm. A polynomial-time approximation scheme (PTAS) is an approximation algorithm that, with a given parameter $\epsilon > 0$, produces in polynomial time a solution whose objective value is at most a multiplicative factor of $1 - \epsilon$ away from the objective value of an optimal solution. A problem is said to be APX-hard if there is a PTAS reduction from all problems in APX to that problem. APX-hard problems that belong to class APX are called APX-complete. We strengthen the result of Theorem 3 by showing that Stackelberg vertex cover in a bipartite graph with a single follower is APX-complete. The following definition is due to Papadimitriou and Yannakakis [17].

Definition 2. Let Π_1 and Π_2 be two optimization problems. We say Π_1 L-reduces to Π_2 if there exists polynomial-time computable functions f, g, and constants $\alpha, \beta > 0$ such that, for each instance $I \in \Pi_1$, the following holds:
1. $f(I) \in \Pi_2$ such that $OPT(f(I)) \leq \alpha \cdot OPT(I)$.
2. Given any solution s to $f(I)$, $g(s)$ is a feasible solution to I such that $|cost(g(s)) - OPT(I)| \leq \beta \cdot |cost(s) - OPT(f(I))|$.

Theorem 4. *StackBipartiteVC with a single follower is APX-complete.*

Proof. Notice that Stackelberg vertex cover in bipartite graphs is in APX. We can compute the optimal prices for each side of the bipartition independently while pricing the priceable nodes on the other side at ∞ by using the algorithm of Briest et al. [7], and by selecting the better of these two solutions, we obtain a 1/2-approximation algorithm.

We use an L-reduction from maximum independent set with degree at most three. Let Π_1 denote the maximum independent set problem with degree at most three, Π_2 denote the Stackelberg vertex cover problem in bipartite graphs, and $I \in \Pi_1$. The maximum independent set problem with degree at most three is shown to be APX-hard by Alimonti and Kann [1]. The polynomial-time transformation f from Π_1 to Π_2 is the same function as we used in the proof of Theorem 3. We show that this transformation satisfies the first property of an L-reduction.

From the proof of Theorem 3, it follows that $OPT(f(I)) = OPT_{SVC} = 5\,|E| + OPT_{MIS} = 5\,|E| + OPT(I)$. This implies that

$$OPT(f(I)) = 5\,|E| + OPT(I) \leq 15\,|V| + OPT(I) \leq 61 \cdot OPT(I),$$

where the first inequality follows from $|E| \leq 3\,|V|$ as the degree of each vertex is at most three and the second inequality from $OPT_{MIS} \geq |V|/4$ as the greedy algorithm already yields an independent set of size $|V|/4$ (for every vertex that is greedily selected, at most three other vertices are deleted). Hence, $\alpha = 61$.

Let s be a feasible solution of $f(I)$ with corresponding chosen vertex cover C^* and profit $\pi(s)$. We define the independent set $g(s)$ of Π_1 as follows. For all $v \in s$, $v \in g(s)$ if $a_v \in C^*$, $p^*_{a_v} = 1$ and $a_u \notin C^*$ for all $\{u, v\} \in E$. By the proof of Theorem 3, $OPT(f(I)) = 5\,|E| + OPT(I)$ and $\pi(s) \leq 5\,|E| + |g(s)|$. This implies that

$$OPT(I) - |g(s)| = OPT(f(I)) - 5\,|E| - |g(s)| \leq OPT(f(I)) - \pi(s),$$

and hence, $\beta = 1$. □

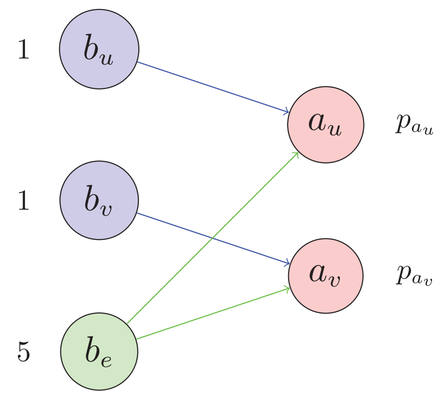
3.2. StackMaxClosure with Two Followers and Unlimited Supply

We now study the Stackelberg max closure problem with two followers and unlimited supply. That is, for each project, there are sufficiently many copies available so that the leader, in principle, could sell a copy of each project to each follower who is interested in that particular project. However, any two copies of the same project need to be sold for the same price.

Briest et al. [7] show that StackBipartiteVC with priceable vertices on one side, three followers, and unlimited supply is weakly NP-complete implying hardness for StackMaxClosure with three followers. We strengthen this result by showing that StackMaxClosure with two followers and unlimited supply is NP-hard as well and even APX-hard.

Theorem 5. *StackMaxClosure with two followers and unlimited supply is NP-hard.*

Figure 6. (Color online) Gadget created for edge $e = \{u, v\} \in E$.



Proof. We show that the decision variant of StackMaxClosure with two followers and unlimited supply is NP-complete. Clearly, the problem is in NP because, for any given target revenue r, closures $C_1^*, C_2^* \subseteq V$, and price vector $\boldsymbol{p}: R \to \mathbb{R}_+$, we can check in polynomial time whether $\sum_{C_1^* \cap R} p(v) + \sum_{C_2^* \cap R} p(v) \geq r$. To see that the problem is NP-complete, consider any instance $G = (V, E)$ of Maximum Independent Set. Define an instance of StackMaxClosure with two followers and unlimited supply consisting of two digraphs $D_i = (B_i \cup R_i, A_i)$ with $B_1 \cap B_2 = \emptyset$ and weights $\boldsymbol{w}: B_1 \cup B_2 \to \mathbb{R}$ as follows (see Figure 6 in which follower D_1 is the blue–red subgraph and D_2 is the green–red subgraph).

a. For each vertex $v \in V$, create a priceable vertex $a_v \in R_1 \cap R_2$, a fixed-weight vertex $b_v \in B_1$ with $w(b_v) = 1$, and add an arc (b_v, a_v) to A_1.

b. For each edge $e = \{u, v\} \in E$, create a fixed-weight vertex $b_e \in B_2$ with $w(b_e) = 5$ and add two arcs (b_e, a_u) and (b_e, a_v) to A_2.

We show that, for every optimal pricing strategy $\boldsymbol{p}^*$ of the leader with associated max closure $C_1^* = C_1^*(\boldsymbol{p}^*)$ in D_1, the vertices in $S^* := \{v \in V \mid a_v \in C_1^*\}$ form a maximum independent set in G and vice versa.

We start by showing that, for any independent set $S \subseteq V$ in G, there exists a price strategy $\boldsymbol{p}^S$ for the leader such that the follower selects among the vertices in $\{a_v \mid v \in V\}$ exactly those corresponding to S, and such that the leader receives a revenue of $5 \cdot |E| + |S|$. Consider an arbitrary independent set $S \subseteq V$ in G and set prices as follows: prices on the vertices in $\{a_v \mid v \in S\}$ are set to one, and prices on all vertices in $\{a_v \mid v \in V \setminus S\}$ are set to $\sum_{u \in N(v) \setminus S} 2.5 + \sum_{u \in N(v) \cap S} (5\,|N(u)| - 1)/\,|N(u)|$, where $N(v)$ denotes the set of neighbors of v in G. This way, the leader forces follower 1 to select all vertices corresponding to the independent set S into the closure (assuming that the follower breaks ties in favor of the leader) and follower 2 to select all priceable vertices into the closure. To see this latter observation, notice that, for all $E' \subseteq E$, we have that

$$\begin{aligned}\sum_{v \in V[E']} \boldsymbol{p}_{(v)} &= |V[E'] \cap S| + \sum_{v \in V[E'] \setminus S} \left(\sum_{u \in N(v) \setminus S} 2.5 + \sum_{u \in N(v) \cap S} \frac{5\,|N(u)| - 1}{|N(u)|} \right) \\ &\geq |V[E'] \cap S| + \sum_{v \in V[E'] \setminus S} \left(\sum_{u \in (N(v) \cap V[E']) \setminus S} 2.5 + \sum_{u \in N(v) \cap V[E'] \cap S} \frac{5\,|N(u) \cap V[E']| - 1}{|N(u) \cap V[E']|} \right) \\ &= 5\,|E'|\,,\end{aligned}$$

where the inequality follows because $(5x - 1)/x$ is increasing in x. In particular, the closure that contains all fixed-price and priceable vertices yields a value of zero. This implies $\boldsymbol{p}^S$ yields a revenue of $|S| + 5\,|E|$.

Conversely, let $\boldsymbol{p}^*$ denote an arbitrary optimal price vector of the leader and C_1^*, C_2^* be the corresponding max closure chosen by follower 1, 2, respectively, with ties broken in favor of the leader. Suppose, for the sake of contradiction, that $S^* := \{v \in V \mid a_v \in C_1^*\}$ is not an independent set. Let $E' \subseteq E$ be the subset of edges for which both endpoints are in S^*. The set E' is not empty because of our assumption of S^* being not an independent set. For each $e = \{u, v\} \in E'$, the optimal price on a_u, a_v is $p^*(a_u), p^*(a_v) \leq 1$. This implies that the optimal profit is at most $|S^*| + 2\,|E'| + 5\,|E \setminus E'|$.

Let $U \subseteq V[E'] \subseteq S^*$ be a subset of vertices of size at most $|E'|$ such that each edge of E' contains at least one end vertex in U. Note that such a closure can easily be constructed by selecting one end vertex of each edge in E' arbitrarily. Then, $S^* \setminus U$ is an independent set, and by setting prices according to price strategy $\boldsymbol{p}^{S^* \setminus U}$ as described, the leader obtains a profit of $|S^* \setminus U| + 5|E| \geq |S^*| + 4|E'| + 5|E \setminus E'| > |S^*| + 2|E'| + 5|E \setminus E'|$, where the first inequality follows because $|U| \leq |E'|$. This is a contradiction to the assumption that $\boldsymbol{p}^*$ is optimal.

Summarizing, for each maximum independent set S^* in G exists a price vector $\boldsymbol{p}^*$ of revenue $5|E| + |S^*|$. Conversely, for each optimal pricing strategy $\boldsymbol{p}^*$ with corresponding vertex cover C^* in G', the set $S^* := \{a_v \mid v \in C^*\}$ forms an independent set in G, and the revenue of the leader is $5|E| + |S^*|$. Hence, the price vector maximizes the leader's revenue if and only if S^* is a maximum independent set. □

Theorem 6. *StackMaxClosure with two followers and unlimited supply is APX-hard.*

Proof. The proof follows the same lines as the proof of Theorem 4. □

3.3. StackMaxClosure with Three Followers and Limited Supply

Whereas StackMaxClosure with two followers is hard in the case of unlimited supply but efficiently solvable in the case only one copy per project is available, the question remains whether StackMaxClosure for more than two followers is efficiently solvable in the setting in which only one copy per project is available. We answer this question negatively. Note that, for three followers, even in the easy case without any fixed-cost project, the problem boils down to finding a maximum independent set in a tripartite graph, which is NP-hard. We can, therefore, conclude that StackMaxClosure with at least three followers and limited supply is NP-hard as well.

Theorem 7. *StackMaxClosure with three followers in the setting in which only one copy of each project is available is NP-hard.*

Proof. We use a reduction from the Maximum Independent Set problem in a tripartite graph. Phillips and Warnow [18] show that this problem is NP-hard. Given the tripartite graph $G = (U_1 \cup U_2 \cup U_3, E)$ for the maximum independent set, we create a StackMaxClosure problem with three followers as follows. We treat the original vertices as benefit projects of the followers with a value of one. For every edge $\{u, v\} \in E$, we create a priceable project $r_{(u,v)}$. We define three bipartite digraphs $D_i = (V_i, A_i)$ with $V_i = U_i \cup \{r_{(u,v)} \mid u \in U_i \text{ or } v \in U_i\}$ for every follower $i =$ 1, 2, 3. We create an arc for every $u \in U_i$ to all $r_{(u,v)}$ and $r_{(v,u)}$ in D_i for $i = 1, 2, 3$. This can be done in polynomial time.

Now, we need to show that the optimal solution of the StackMaxClosure problem gives us an optimal solution for the Maximum Independent Set. Observe that a benefit project can only be selected if all priceable projects connected to it are assigned to that follower. Given that priceable projects correspond to edges, a benefit project can only be selected if all edges incident to that vertex are assigned to that follower. So an assignment x in the StackMaxClosure problem induces an independent set. Because, in the StackMaxClosure problem, the number of selected benefit projects is maximized, the size of the independent set is maximized and, hence, the correspondence. □

Theorem 8. *There is a $2/k$-approximation algorithm for StackMaxClosure with $k \geq 2$ followers and limited supply.*

Proof. By Theorem 2, we can solve the problem with two followers. Observe that there are $\binom{k}{2} = k(k-1)/2$ pairs of players. We can compute the optimal prices for each pair of players by using the algorithm in the proof of Theorem 2 and setting all other prices equal to infinity. By selecting the best of these solutions, we obtain a $2/k$-approximation algorithm. □

4. Open Problems

We have solved the complexity status of StackMaxClosure with limited and unlimited supply, but several interesting classes of problems remain open. First, we assume that the digraphs of the followers are not allowed to intersect in cost projects. Can the algorithm for two followers with limited supply be extended in case we relax this assumption? Second, Briest et al. [7] show that StackBipartiteVC with priceable vertices on one side and a single follower can be solved in polynomial time. We prove that StackBipartiteVC is NP-hard. This raises the question whether the Stackelberg vertex cover can be solved in polynomial time in simpler classes of networks such as trees. Third, are there any better approximation algorithms for the NP-hard problems discussed in this paper?

Appendix. Proof of Theorem 1

Proof of Theorem 1. The proof basically goes along the same lines as the proof in Briest et al. [7] for StackBipartiteVC with a single follower and with priceable vertices on one side of the partition.

Claim 1. *In every iteration of Algorithm 1 with current maximum flow f^*, all priceable vertices $v \in R$ belong to the current cut* $C(f^*) \cup \{s\} = \{v \in \hat{V} \mid \exists \text{ no v-t-path in } \hat{D}_{f^*}\}$.

Proof of Claim 1. Clearly, at the beginning of the algorithm with initial max-flow f^*, all priceable vertices belong to $C(f^*)$ because $c(v,t) = 0$ for all $v \in R$. For the sake of contradiction, suppose there is a first iteration of the while loop, in which at the end of the while loop the statement of the claim is false. Let $v \in R$ be the vertex selected for augmentation, f^* be the optimal flow at the beginning of the while loop, and $\hat{f}$ be the optimal flow at the end of the while loop, that is, after the capacity on arc (v, t) has been increased by the amount $\hat{f}(v,t) - f^*(v,t)$ through augmentation along the path P chosen for augmentation. Note that arc (v, t) remains saturated and so does not enter the new residual network $\hat{D}_{\hat{f}}$. To prove the claim, it remains to show that $C(f^*) \setminus C(\hat{f}) = \emptyset$. Suppose there is some $w \in C(f^*) \setminus C(\hat{f})$. Thus, there exists a w–t path Q in $\hat{D}_{\hat{f}}$ that was not present in $\hat{D}_{f^*}$. It follows that Q contains at least one arc (i, j) that was created by augmentation along P. Consider the last such arc (i, j) in Q. Because the arc was created through augmentation along P, it follows that P contains the arc (j, i) in the opposite direction. Thus, there must have been an s–t path consisting of the subpath of P from s to j and the subpath of Q from j to t prior to the augmentation along P. This is the desired contradiction to the optimality of f^* and finishes the proof of the claim. □

Let c_∞ denote the value of the maximum flow in $\hat{D}$ if $c(v,t) = \infty$ for all $v \in R$, and let c_0 denote the value of the maximum flow in $\hat{D}$ if $c(v,t) = 0$ for all $v \in R$. Notice that the difference $c_\infty - c_0$ is an upper bound on the revenue of the leader because $c_\infty - c_0$ is exactly the maximal gain in the objective for the follower that can be achieved by using projects in R. To show that Algorithm 1 terminates with optimal prices p^*, we show that the induced revenue $\pi^*(D)$ is equal to this upper bound $c_\infty - c_0$.

By our claim, all priceable vertices are selected by the follower. This implies that the revenue of the leader is the difference between the value of the minimum cut after Algorithm 1 terminated and the value of the minimum cut with $p_v = 0$. Therefore, we only need to show that c_∞ is the value of a minimum cut after the termination of Algorithm 1.

Assume we take the solution of the algorithm and increase the price on all priceable vertices by some arbitrarily small $\epsilon > 0$. Because the algorithm has terminated, increasing the prices does not yield an augmenting path. If there exists an augmenting path, then this path uses only one (v, t) edge, and thus, this path also exists if only the price of v is increased, which cannot be. This implies that the maximum flow remains the same. Moreover, all priceable vertices leave the minimum cut. This proves that, after the algorithm terminates, the value of the minimum cut must be c_∞. □

References

[1] Alimonti P, Kann V (2000) Some APX-completeness results for cubic graphs. *Theoretical Comput. Sci.* 237(1–2):123–134.
[2] Baïou M, Barahona F (2016) Stackelberg bipartite vertex cover and the preflow algorithm. *Algorithmica* 74(3):1174–1183.
[3] Balcan MF, Blum A, Mansour Y (2008) Item pricing for revenue maximization. *Proc. Ninth ACM Conf. Electronic Commerce* (ACM, New York), 50–59.
[4] Bilò D, Gualà L, Proietti G, Widmayer P (2008) Computational aspects of a 2-player Stackelberg shortest paths tree game. *Internat. Workshop Internet Network Econom.* (Springer, Berlin, Heidelberg), 251–262.
[5] Böhnlein T, Kratsch S, Schaudt O (2021) Revenue maximization in Stackelberg pricing games: Beyond the combinatorial setting. *Math. Programming* 187(1):653–695.
[6] Böhnlein T, Schaudt O, Schauer J (2019) Stackelberg packing games. *Workshop Algorithms Data Structures* (Springer, Cham, Switzerland), 239–253.
[7] Briest P, Hoefer M, Krysta P (2012) Stackelberg network pricing games. *Algorithmica* 62(3):733–753.
[8] Briest P, Chalermsook P, Khanna S, Laekhanukit B, Nanongkai D (2010) Improved hardness of approximation for Stackelberg shortest-path pricing. *Internat. Workshop Internet Network Econom.* (Springer, Berlin, Heidelberg), 444–454.
[9] Cabello S (2012) Stackelberg shortest path tree game, revisited. Preprint, submitted July 10, https://arxiv.org/abs/1207.2317.
[10] Cardinal J, Demaine ED, Fiorini S, Joret G, Langerman S, Newman I, Weimann O (2011) The Stackelberg minimum spanning tree game. *Algorithmica* 59(2):129–144.
[11] Cristi A, Schröder M (2022) Negative prices in network pricing games. *Oper. Res. Lett.* 50(2):99–106.
[12] Hochbaum DS (2001) A new–old algorithm for minimum-cut and maximum-flow in closure graphs. *Networks* 37(4):171–193.
[13] Joret G (2011) Stackelberg network pricing is hard to approximate. *Networks* 57(2):117–120.
[14] Karp RM (1972) *Reducibility Among Combinatorial Problems*. Miller RE, Thatcher JW, Bohlinger JD, eds. Complexity of Computer Computations. *The IBM Res. Sympos. Ser.* (Springer, Boston), 85–103.
[15] Kőnig D (1931) Gráfok és mátrixok. *Matematikai és Fizikai Lapok* 38:116–119.
[16] Labbé M, Marcotte P, Savard G (1998) A bilevel model of taxation and its application to optimal highway pricing. *Management Sci.* 44(12):1608–1622.
[17] Papadimitriou CH, Yannakakis M (1991) Optimization, approximation, and complexity classes. *J. Comput. System Sci.* 43(3):425–440.
[18] Phillips C, Warnow TJ (1996) The asymmetric median tree—A new model for building consensus trees. *Discrete Appl. Math.* 71(1–3): 311–335.
[19] Picard JC (1976) Maximal closure of a graph and applications to combinatorial problems. *Management Sci.* 22(11):1268–1272.
[20] Roch S, Savard G, Marcotte P (2005) An approximation algorithm for Stackelberg network pricing. *Networks* 46(1):57–67.
[21] Van Hoesel S (2008) An overview of Stackelberg pricing in networks. *Eur. J. Oper. Res.* 189(3):1393–1402.
[22] Von Stackelberg H (1934) *Marktform und Gleichgewicht* (Springer, Vienna).