Autonomous Agents and Multi-Agent Systems (2023) 37:21
https://doi.org/10.1007/s10458-023-09597-7

Fine-grained view on bribery for group identification

Niclas Boehmer[1] · Robert Bredereck[1,2,3] · Dušan Knop[4] · Junjie Luo[1,5,6]

Accepted: 4 January 2023 / Published online: 24 March 2023

Abstract

Given a set of agents qualifying or disqualifying each other, group identification is the task of identifying a *socially qualified* subgroup of agents. Social qualification depends on the specific rule used to aggregate individual qualifications . The classical bribery problem in this context asks how many agents need to change their qualifications in order to change the outcome in a certain way. Complementing previous results showing polynomial-time solvability or NP-hardness of bribery for various social rules in the constructive (aiming at making specific agents socially qualified) or destructive (aiming at making specific agents socially disqualified) setting, we provide a comprehensive picture of the parameterized computational complexity landscape. Conceptually, we also consider a more fine-grained concept of bribery cost, where we ask how many single qualifications need to be changed, nonunit prices for different bribery actions, and a more general bribery goal that combines the constructive and destructive setting.

1 Introduction

The University of Actual Truth (UAT) was paralyzed for months due to a heavy dispute of the scientists about who belongs to the group of *true scientists*. After some literature research on group identification, they asked every scientist to report who they believe is

✉ Robert Bredereck
robert.bredereck@tu-clausthal.de

Niclas Boehmer
niclas.boehmer@tu-berlin.de

Dušan Knop
dusan.knop@fit.cvut.cz

Junjie Luo
jjluo1@bjtu.edu.cn

1 Algorithmics and Computational Complexity, TU Berlin, Berlin, Germany

2 Humboldt-Universität zu Berlin, Berlin, Germany

3 Present Address: TU Clausthal, Clausthal-Zellerfeld, Germany

4 Czech Technical University in Prague, Prague, Czech Republic

5 Nanyang Technological University, Singapore, Singapore

6 School of Mathematics and Statistics, Beijing Jiaotong University, Beijing, China

qualified for being a true scientist (i.e., each scientist hands in a list of all scientists he or she believes to be a true scientist). Based on these individual qualifications, they applied several group identification rules to find the group of true scientists. Unfortunately, each rule either decides that nobody is a true scientist or that all are true scientists. It is, however, obvious to everyone that the group of true scientists must be a proper nonempty subset of scientists.

To get a feeling for how the group of true scientists could look like, presidents of friendly universities were invited to submit a proposal specifying who they believe the true scientists within the university are. To evaluate and compare every group being proposed by some president, they computed two different quality measures. For the rules that identified nobody as a true scientist, they defined the "truth distance" as the minimum number of scientists whose qualifications would need to be changed to make the group proposed by some president part of the true scientists. For the rules that initially identified everyone as a true scientist, they defined the "margin of truth" as the minimum number of scientists whose qualifications would need to be changed to make no one from the group proposed by some president a true scientist.

The above example describes a problem that appears in many situations where one needs to identify a socially qualified group of agents based only on the agents' pairwise binary qualifications. To solve this task, group identification rules have been developed [28, 37]: Among the most important rules studied in the literature, which are studied in this paper, are the *consent rule* [37] and the two iterative rules: *consensus-start-respecting rule* (CSR by Kasher [27]) and *liberal-start-respecting rule* (LSR by Kasher and Rubinstein [28]). Using the consent rule, which is parameterized by integers s and t, each agent qualifying itself is socially qualified if and only if at least s agents qualify it, and each agent disqualifying itself is socially disqualified if and only if at least t agents disqualify it. In the iterative rules, some criterion is used to determine an initial set of socially qualified agents, which is then iteratively extended by adding all agents who are qualified by at least one agent from the set of already socially qualified agents until convergence is reached. Under CSR, an agent is initially socially qualified if it is qualified by all agents, while, under LSR, all agents qualifying themselves are initially socially qualified.

Despite its simplicity, our example illustrates an important aspect of group identification rules: Group identification rules provide only a binary decision about membership to a specific group, while multiple degrees of certainty about membership may be desired. In extreme cases (as in the example), the identified group might contain indeed too many or too few agents. The distances that "solve" this issue in our example are concepts known in the literature, but usually motivated from a different viewpoint: "truth distance" corresponds to constructive bribery and "margin of truth" corresponds to destructive bribery. Herein, the classical bribery model assumes an external agent (with full knowledge over the individual qualifications) that aims to influence the outcome of the group identification process by convincing a limited number of agents to change their qualifications in a certain way to achieve some goal. While the assumptions behind the classical bribery motivation may be questionable in the context of group identification, we emphasize that computing bribery costs as a quality measure is very natural and useful in practice, as illustrated in our example.

Table 1 Overview of all our complexity results except for our parameterized complexity results for consent rules (see Figure 1)

	$f^{\mathrm{CSR}}/f^{\mathrm{LSR}}$		$f^{(s,t)}$	
	Agent	Link	Agent	Link
Const	P (†)	NP-c. (Th. 3) FPT wrt. $\lvert A^+ \rvert$ (Th. 4) W[1]-h. wrt. ℓ (Th. 3)	NP-c. (†)	P (Ob. 2)
Dest	P (†)	*P* (Th. 2)	NP-c. (†)	P (Ob. 2)
Const+Dest	P (Th. 1)	NP-c. (Th. 3)	NP-c. (Ob. 3)	P (Ob. 2)
Exact	P (Co. 1)	P (Th. 2)	NP-c. (Ob. 3)	P (Ob. 2)

Results with a † were proven by Erdélyi et al. [17]. All polynomial-time and fixed-parameter tractability results also hold for the priced versions of the considered problems

1.1 Our contributions

In this paper, we provide a more fine-grained view on computing bribery costs for group identification rules in four ways. First, we allow to combine *constructive* and *destructive bribery*. In particular, we allow to specify two disjoint sets A^+ and A^- of agents that must be (resp., must not be) socially qualified after the bribery action. This includes as a special case *exact bribery*, where one can specify the final socially qualified subgroup of agents. Second, we consider a more fine-grained concept of bribery costs called *link bribery*, where one counts the number of individual qualifications that need to be changed. So far, in the classical model, which we call *agent bribery*, only the number of agents that alter their qualifications is counted. Third, we consider priced versions of our bribery problems, i.e., in link bribery, each qualification has a price of being changed and, in agent bribery, each agent has a price of being bribed. Notably, the priced and unpriced versions of all considered problems have the same complexity, i.e., all our hardness results hold for the unpriced problems, while all algorithmic results apply to the priced problems. Fourth, we complement the classical (P vs. NP) computational complexity landscape by providing a comprehensive analysis of the parameterized complexity, focusing on naturally and well-motivated parameters such as the bribery cost and the sizes of the sets A^+ and A^- as well as rule-specific parameters. We refer to Table 1 and Fig. 1 for an overview of our results and to Sect. 2 for formal definitions of the rules and parameters. Note that the results for constructive bribery depicted in Fig. 1 analogously hold for destructive bribery with switched roles of s and t (see the work of Erdélyi et al. [17] and Lemma 1).

1.2 Related work

Faliszewski et al. [19] introduced bribery problems to the analysis of elections by studying the problem of making a given candidate a winner by changing the preferences of at most a given number of voters (a problem being closely related to constructive agent bribery in our setting). Since then, multiple variants of bribery differing in the goal and the pricing of a bribery have been proposed (see a survey by Faliszewski and Rothe [18]) and bribery problems have also been studied in the context of other collective decision problems [1, 4]. For example, Faliszewski et al. [20] introduced *microbribery*, where the manipulator pays

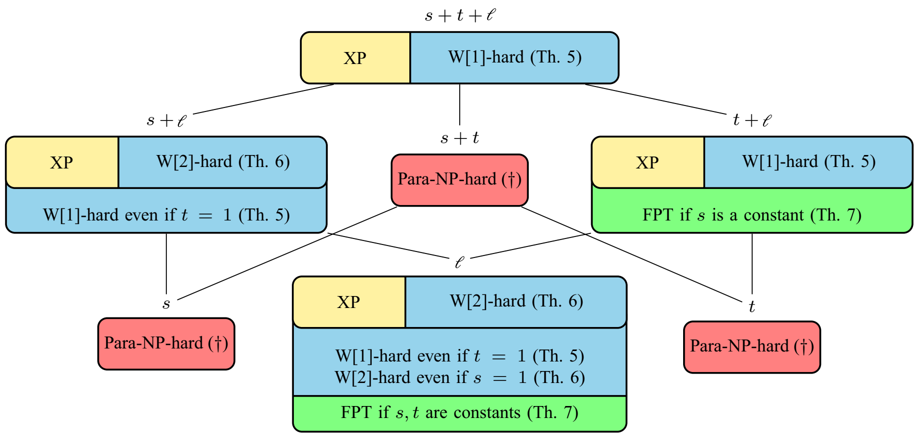

Fig. 1 Parameterized analysis of Const-$f^{(s,t)}$ Agent Bribery. Upper and lower bounds on the complexity of this problem with respect to each parameter are shown in the first line, followed by some special cases. The FPT result from Theorem 7 also holds for Const-$f^{(s,t)}$ \$Agent Bribery. The "XP" results for all parameter combinations containing ℓ are trivial and also hold for the priced version. Results with a † were proven by Erdélyi et al. [17]. Additionally, in Theorem 8, we prove that Const-$f^{(s,t)}$ \$Agent Bribery is FPT wrt. $|A^+|$

per flip in the preference profile of the given election. Microbribery is conceptually closely related to link bribery in the context of our problem. Furthermore, while Baumeister et al. [1] already considered a variant of exact bribery in the context of judgment aggregation, we are not aware of any applications of the combined setting of constructive and destructive bribery that we propose in this paper.

Despite different initial motivations, bribery in elections is closely related to the concept of *margin of victory*, where the goal is to measure the robustness of the outcome of an election or the "distance" of a candidate from winning the election [7, 31]. While both concepts have been mostly studied separately, some authors have developed a unified framework [3, 22, 38].

Initially, the group identification problem has been mainly studied from a social choice perspective by an axiomatic analysis of the problem and some social rules (see, e.g., the works of Dimitrov [10], Kasher and Rubinstein [28], and Samet and Schmeidler [37]). Possible applications of the group identification problem range from the identification of a collective identity [28] to the endowment of rights with social implications [37].

Recently, Yang and Dimitrov [39] and Erdélyi et al. [17] initiated the study of manipulation by an external agent both in the context of bribery and control in a group identification problem for the three mentioned social rules. Yang and Dimitrov [39] considered the complexity of agent deletion, insertion, and partition for constructive control, while Erdélyi et al. [17] extended their studies to destructive control. Moreover, Erdélyi et al. [17] analyzed the complexity of constructive agent bribery and destructive agent bribery. They proved that for both destructive and constructive bribery, for CSR and LSR, the related computational problems are polynomial-time solvable. Moreover, for constructive bribery, they proved that the computational problems for consent rules with $t = 1$ are also polynomial-time solvable. On the other hand, for $t \geq 2$ and $s \geq 1$, constructive bribery is already NP-complete. Finally, Erdélyi et al. [17] established a close relationship between constructive and destructive bribery for consent rules by proving that every constructive bribery problem can be converted into a destructive bribery problem by switching s and t and flipping all qualifications.

Independently to this paper (and in parallel to our conference version), Erdélyi and Yang [16] studied partly overlapping questions. In particular, they investigated the computational complexity of constructive and exact bribery for all three social rules considered in this paper assuming that the briber pays per modified qualification, which is equivalent to our link bribery cost model. In essence, the results of Erdélyi and Yang [16] overlap with ours in the unpriced version of the second part of Theorem 2, the NP-hardness from Theorem 3 and the unpriced version of Observation 2. In addition, Erdélyi and Yang [16] considered how the computational complexity of these problems changes if one requires that every agent qualifies exactly r agents before and after the bribery.

The group identification problem is formally related to multiwinner voting [21]. However, multiwinner voting is of a different flavor both in terms of intended applications and studied rules. Nevertheless, formally, group identification is equivalent to approval-based multiwinner voting with a variable number of winners where the set of voters and candidates coincide. While (approval-based) multiwinner voting with a variable number of winners has been studied by, for example, Duddy et al. [13], Kilgour [29], Lackner and Maly [30] and Faliszewski et al. [23], this specific setting has never been studied from a voting perspective. Moreover, so far, the work on bribery for (approval-based) multiwinner elections is limited to the setting where the number of winners is fixed [6, 22]. Notably, similar to our motivation, Faliszewski et al. [22] also studied how to measure the margin of victory in approval-based multiwinner elections through the lens of bribery.

The reason why we study the consent, consensus-start-respecting, and liberal-start-respecting rules is threefold. First, the two previous works on bribery and control in group identification [17, 39] also dealt with exactly these three rules. Second, these three rules are often studied in the group identification literature [10, 11, 26, 28, 33, 37]. One reason for their popularity might be that all three can be characterized via some natural axioms.[1] Third, at least for the two iterative rules it can easily happen that either all or none of the agents is socially qualified, a scenario in which the "truth distance" or "margin of truth" are particularly valuable: If no agent is qualified by every agent, no agent is socially qualified under the consensus-start-respecting rule. Under the liberal-start-respecting rule, it might happen that no one qualifies itself if social qualification comes with some duty (leading to no one being socially qualified) or everyone qualifies itself if social qualification comes with some right (leading to everyone being socially qualified). Moreover, for both rules, as soon as there is a single socially qualified agent, it can easily happen that all agents become socially qualified (i.e., as soon as the directed graph G, where agents are vertices and there is an arc from one agent a to another agent a' if a qualifies a', is strongly connected).

1.3 Organization

In Sect. 2, we formally define the different computational questions we examine and provide background on parameterized complexity theory and some graph algorithms that we later use in our algorithms. Subsequently, in Sect. 3, we present our results for iterative rules dealing with agent bribery first (Subsect. 3.1) and afterwards turning to link bribery (Subsect. 3.2). In Sect. 4, we consider the consent rule. We start by observing that there

[1] A social rule is a consent rule if and only if it satisfies monotonicity, independence, and anonymity [37]. The consensus-start respecting rule is the only social rule satisfying consensus, minimal robustness, and irrelevance of outsider's view 1 [11]. The liberal-start respecting rule is the only social rule satisfying consensus, monotonicity, equal treatment of insiders' views, and irrelevance of outsider's view 2 [11].

exists a simple algorithm for (priced) link bribery for all our goals and subsequently conduct a detailed analysis of constructive link bribery in Subsect. 4.1 before explaining how the results of this analysis extend to the other goals in Subsect. 4.2. We conclude in Sect. 5 with a summary of our results and multiple pointers for possibilities for future work.

2 Preliminaries

In this section, we start by formally introducing the group identification problem and all considered social rules (Subsect. 2.1). Subsequently, we formally define all bribery goals and bribery cost models we analyze in this paper (Subsect. 2.2). Finally, in Subsect. 2.3, we introduce some graph theoretical notation, describe some common graph problems and the respective algorithms we use in our paper, and provide a brief introduction to parameterized complexity analysis.

2.1 Group identification

Given a set of agents $A = \{a_1, \dots, a_n\}$ and a so-called *qualification profile* $\varphi : A \times A \to \{-1, 1\}$, the group identification problem asks to return a subset of *socially qualified* agents using some *social rule f*. We write $f(A, \varphi)$ to denote the set of agents that are socially qualified in the group identification problem (A, φ) according to *f*. All agents which are not socially qualified are called *socially disqualified*. For two agents $a, a' \in A$, we say that *a qualifies a′* if $\varphi(a, a') = 1$; otherwise, we say that *a disqualifies a′*. For each agent $a \in A$, let $Q^+_\varphi(a) = \{a' \in A \mid \varphi(a', a) = 1\}$ denote the set of agents qualifying a and $Q^-_\varphi(a) = \{a' \in A \mid \varphi(a', a) = -1\}$ the set of agents disqualifying a in φ. Let $A^*_\varphi = \{a \in A \mid \forall a' \in A : \varphi(a', a) = 1\}$ be the set of agents who are qualified by everyone including themselves. We omit the subscript φ if it is clear from context. For every group identification problem, A and φ induce a so-called (directed) *qualification graph* $G_{A,\varphi} = (A, E)$ with $(a, a') \in E$ if and only if $\varphi(a, a') = 1$. For two agents a and a', we say that there exists a path from a to a' in (A, φ) if there exists a path from a to a' in $G_{A,\varphi}$.

Now, we define the social rules considered in this paper: For the *liberal-start-respecting rule* (f^{LSR}), we start with the set $K_1 = \{a \in A \mid \varphi(a, a) = 1\}$ and compute the set of socially qualified agents iteratively for $i = 2, \dots$ using

$$K_i = \left\{a \in A \mid \exists a' \in K_{i-1} : \varphi(a', a) = 1\right\}. \tag{1}$$

Notice that we always have $K_{i-1} \subseteq K_i$. We stop the process when $K_{i-1} = K_i$ and output K_i.

For the *consensus-start-respecting rule* (f^{CSR}), we start with the set $K_1 = A^*$ and for $i = 2, \dots$ we use Eq. 1) to compute iteratively the set of socially qualified agents. Note that both $f^{\mathrm{LSR}}$and f^{CSR}can be viewed as multiple-source breadth-first searches (with different sets of starting vertices). Accordingly, it is also possible to compute the set of socially qualified agents under f^{LSR} and f^{CSR} as the set of agents that correspond to vertices in the qualification graph that are reachable from vertices with a self-loop and vertices with incoming arcs from all vertices, respectively.

The *consent rule* ($f^{(s,t)}$) with parameters s and t with $s + t \leq n + 2$ determines the set of socially qualified agents as follows: If $\varphi(a, a) = 1$ for an agent $a \in A$, then a is socially qualified if and only if $|Q^+(a)| \geq s$. If $\varphi(a, a) = -1$ for an agent $a \in A$, then a is socially disqualified if and only if $|Q^-(a)| \geq t$. Note that the constraint $s + t \leq n + 2$ is chosen in a way that an agent needs less or as many qualifications ($s - 1 \leq n - (t - 1)$) from other agents to be socially qualified if it qualifies itself compared to if it disqualifies itself. Thus, an agent who qualifies itself but is socially disqualified cannot become socially qualified by simply disqualifying itself.

2.2 Bribery variants and costs

In the most general form of bribery, which we call Constructive+Destructive (Const.+Dest.) bribery, we are given a set A of agents, a qualification profile φ, and a social rule f together with two groups of agents A^+ and A^- and a budget ℓ. The task is then to alter the qualification profile φ such that in the altered profile φ' all agents in A^+ are socially qualified, i.e., $A^+ \subseteq f(A, \varphi')$, and all agents in A^- are socially disqualified, i.e., $A^- \subseteq A \setminus f(A, \varphi')$. The cost of the bribery (computed as specified below) is not allowed to exceed ℓ.

We also consider the following three special cases of Const+Dest-Bribery:

Constructive (Const.) Given a set $A^+ \subseteq A$ of agents, find a bribery such that $A^+ \subseteq f(A, \varphi')$.

Destructive (Dest.) Given a set $A^- \subseteq A$ of agents, find a bribery such that $A^- \subseteq A \setminus f(A, \varphi')$.

Exact Given a set $A^+ \subseteq A$ of agents, find a bribery such that $A^+ = f(A, \varphi')$.

We now specify the cost of a bribery. As the most important and natural special case, we consider *unit prices*. In Agent Bribery (with unit prices) the cost of a bribery is equal to the number of agents whose opinions are modified. Consequently, we ask whether it is possible to achieve the specified goal by altering the preferences of at most ℓ agents, where the briber is allowed to change the preferences of each agent in an arbitrary way. In Link Bribery (with unit prices), the cost of a bribery is equal to the number of single qualifications changed. Therefore, we ask whether it is possible to achieve the specified goal by altering at most ℓ qualifications, that is, flipping at most ℓ entries in φ.

More generally, we consider *priced variants* (denoted by a "$" as prefix in the name) of our problems where we are additionally given a price function ρ. For $Agent Bribery, ρ assigns to each agent a a positive integer price $\rho(a)$. For a subset of agents $A' \subseteq A$, we write $\rho(A')$ to denote $\sum_{a \in A'} \rho(a)$. For $Link Bribery, the price function ρ assigns to each ordered agent pair $(a, a') \in A \times A$ a positive integer price $\rho((a, a'))$. The costs of a bribery for $Agent Bribery is the sum of prices assigned to the bribed agents $A' \subseteq A$, i.e., $\sum_{a \in A'} \rho(a)$. The costs of a bribery for $Link Bribery is the sum of prices assigned to the modified qualifications $M' \subseteq A \times A$, i.e., $\sum_{(a,a') \in M'} \rho((a, a'))$. We always assume that the maximum cost of an action, that is, the maximum value of ρ is bounded by a polynomial in the number of agents. Note that Agent Bribery (resp., Link Bribery) is equivalent to $Agent Bribery (resp. $Link Bribery) with ρ assigning price 1 to every agent (resp., to every agent pair).

While we formulated all bribery problems as search problems, we sometimes also consider the naturally associated decision variant where the question is whether there is a successful bribery whose cost is at most some given integer.

Example. Let us consider an instance of the group identification problem consisting of five agents $A = \{a_1, a_2, a_3, a_4, a_5\}$ and the following qualification profile φ as an example.

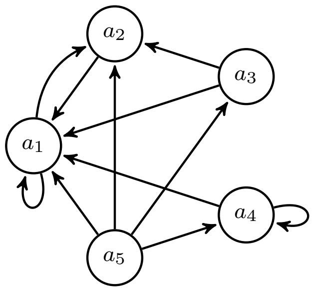

Fig. 2 Qualification graph of example instance from Subsect. 2.2

$$\varphi = \begin{array}{c} \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{array} \begin{array}{c} \begin{array}{ccccc} a_1 & a_2 & a_3 & a_4 & a_5 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

We also include the qualification graph of this instance in Fig. 2. For the liberal-start-respecting rule f^{LSR}, we start with the set of agents $K_1 = \{a_1, a_4\}$ that qualify themselves and have $K_2 = K_3 = \{a_1, a_2, a_4\}$. Thus, it holds that $f^{\mathrm{LSR}}(A, \varphi) = \{a_1, a_2, a_4\}$. For the consensus-start-respecting rule f^{CSR}, we start with the set of agents $K_1 = \{a_1\}$ that are qualified by everyone and have $K_2 = K_3 = \{a_1, a_2\}$. Thus, it holds that $f^{\mathrm{CSR}}(A, \varphi) = \{a_1, a_2\}$. For the consent rule $f^{(s,t)}$, the set of socially qualified agents depends on the chosen parameters s and t. Let $s = t = 3$, i.e., an agent needs two other agents agreeing on its opinion about itself to be classified according to its opinion. Thus, it holds that $f^{(s,t)}(A, \varphi) = \{a_1, a_2\}$.

To exemplify the bribery problems we study in this paper, we present some example questions and solutions for the above instance. Let us start with the liberal-start-respecting rule f^{LSR} and Link Bribery. Assuming that we are in the Constructive setting and the goal is to make all agents socially qualified ($A^+ = A$), there exist four optimal solutions, i.e., bribe one of a_1, a_2, a_4, and a_5 to qualify a_5. Assuming that we are in the Constructive+Destructive setting and the goal is to make a_5 socially qualified and a_3 socially disqualified ($A^+ = \{a_5\}$ and $A^- = \{a_3\}$), one optimal solution is to bribe a_5 to qualify itself and to disqualify a_3. We now turn to the consent rule $f^{(s,t)}$ with $s = t = 3$ and Agent Bribery. Assuming that we are in the Destructive setting and the goal is to make everyone disqualified ($A^- = A$), one optimal solution is to bribe a_1, a_2, and a_5 to disqualify everyone. Finally, assuming that we are in the Exact setting and the goal is to make a_1 and a_4 socially qualified and everyone else socially disqualified ($A^+ = \{a_1, a_4\}$), the optimal solution is to either bribe a_1 or a_3 to disqualify a_2 and qualify a_4.

2.3 Graph theory, graph algorithms, and parameterized complexity

A graph is a pair (V, E) of *vertices* and *edges*; if the graph is directed, then E is the set of *arcs*. A *walk* from s to t in a (directed) graph is a sequence $v_0 e_1 v_1 \dots e_p v_p$ with $s = v_0$ and $t = v_p$, where $e_i = (v_{i-1}, v_i)$ for all $i = 1, \dots, p$. A path is a walk where each vertex occurs at most once. Let $G = (V, E)$ be a graph, $F \subseteq E$ a set of edges, and $U \subseteq V$ a set of vertices. We write $G \setminus F$ to denote the graph $(V, E \setminus F)$ and $G - U$ to denote the graph with vertex set $V \setminus U$ whose edge set is the set of edges $e \in E$ with both endvertices in $V \setminus U$. We use standard graph notation; see, e.g., the monograph of [32]. We now define several graph problems that we use in the construction of different algorithms each accompanied by a statement concerning the running time for solving the respective problem.

Minimum Weighted Cut

Input:	A directed graph $G = (V, E)$ with arc weights $w\colon V \times V \to \mathbb{N}$ and two distinct vertices σ and τ.
Task:	Find a set of arcs $F \subseteq E$ minimizing $\sum_{e \in F} w(e)$ such that there is no path from σ to τ in $G \setminus F$.

Fact 1 ([36]) Minimum Weighted Cut can be solved in $O(|V| \cdot |E|)$ time.

Minimum Weighted Separator

Input:	A directed graph $G = (V, E)$ with vertex weights $w\colon V \to \mathbb{N}$ and two distinct vertices σ and τ.
Task:	Find a set of vertices $U \subseteq V \setminus \{\sigma, \tau\}$ minimizing $\sum_{u \in U} w(u)$ such that there is no path from σ to τ in $G - U$.

One can solve Minimum Weighted Separator using Minimum Weighted Cut due to, e.g., the following folklore construction: Given a graph $G = (V, E)$, to construct the corresponding flow network, every vertex $x \in V$ is replaced by two vertices x_{in} and x_{out} such that every ingoing arc to x is replaced by an ingoing arc to x_{in} and every outgoing arc from x is replaced by an outgoing arc from x_{out}. Furthermore, one adds an arc with weight w(x) from x_{in} to x_{out} and sets the weight of all other arcs to ∞. It is easy to verify that now a minimum weighted cut corresponds to an minimum weighted (σ, τ)-separator of the original graph.

Fact 2 ([36]) Minimum Weighted Separator can be solved in $O(|V| \cdot |E|)$ time.

Minimum Weighted Spanning Arborescence

Input:	A directed graph $G = (V, E)$ with arc weights $w\colon V \times V \to \mathbb{N}$ and a root r.
Task:	Find a set of arcs $F \subseteq E$ of the minimum weight such that in the graph (V, F) every vertex is reachable by a unique directed path from the root r.

Fact 3 ([8, 8, 14, 25, 25]) Minimum Weighted Spanning Arborescence can be solved in $O(|E| + |V| \log |V|)$ time.

Weighted Directer Steiner Tree (WDST)

Input:	A directed graph $G = (V, E)$ with arc weights $w\colon V \times V \to \mathbb{N}$, a set of terminals $T \subseteq V$, and a root vertex s.
Task:	Find a subset of arcs of G minimizing the total weight such that there is a directed path from the root s to every terminal $t \in T$.

Fact 4 ([34]) There is an algorithm solving Weighted Directer Steiner Tree whose running-time depends exponentially on $|T|$ (in fact, by $2^{|T|}$) and polynomially on the size of the input and the maximum arc weight. *Parameterized complexity.* To provide a fine-grained computational complexity analysis of our problems, we use tools from parameterized complexity [9, 12, 24, 35]. Herein, one identifies a *parameter k* (a positive integer) and takes a closer look at the computational complexity of the problem with respect to the parameter k and input size. A problem parameterized by k is called *fixed-parameter tractable* (in FPT) if it can be solved in $f(k) \cdot |I|^{O(1)}$ time, where $|I|$ is the size of a given instance, k is the parameter value, and f is a computable (typically super-polynomial) function. In order to disprove fixed-parameter tractability, we use a well-known complexity hierarchy of classes of parameterized problems:

$$\text{FPT} \subseteq \text{W}[1] \subseteq \text{W}[2] \subseteq \cdots \subseteq \text{XP}.$$

All these inclusions are widely believed to be proper. Hardness for parameterized classes are defined through parameterized reductions which are similar to classical polynomial-time many-one reductions. In this paper, it suffices to use polynomial-time many-one reductions which additionally ensure that the value of the parameter in the problem we reduce to depends only on the value of the parameter of the problem we reduce from. To look for tractable cases beyond single parameters, we also consider *parameter combinations*. For example, when referring to the combined parameter $k' + k'' + k'''$, we implicitly define a parameter $k = k' + k'' + k'''$. Naturally, hardness for a combined parameter implies hardness for each single parameter, while fixed-parameter tractability for a single parameter implies fixed-parameter tractability for each parameter combination that involves the respective parameter.

3 Iterative rules

We start by considering agent bribery in Subsect. 3.1, before we examine link bribery in Subsect. 3.2.

3.1 Agent bribery

For agent bribery, the already known positive results for constructive bribery and destructive bribery [17] extend to constructive+destructive bribery and even to its priced version. The algorithm for the general problem, however, is significantly more involved than the known ones, as both "positive" and "negative" constraints need to be simultaneously taken into account. The general idea of the algorithm is to bribe the agents that form a minimum (weighted) separator between the agents in A^+ and the agents in A^- in the (slightly altered) qualification graph such that they qualify themselves and all agents in A^+.

Before we start with formally proving this result, we make an observation about the start of the social qualification process of our iterative rules that allows to simplify our algorithms significantly. Note that for both iterative rules there are two reasons for an agent a to be socially qualified: The first reason is that the agent a becomes socially qualified in the first phase, i.e., $a \in K_1$, because a is either qualified by all other agents or qualifies itself.

The second reason is that the agent a is qualified by one other agent a' that was already qualified in a previous iteration. For agent bribery, it is clear that one may have to use the first option for some instances (if $K_1 = \emptyset$ and $A^+ \neq \emptyset$). The second option, however, is intuitively much cheaper so that using the first option should be necessary at most once. That this is indeed the case is not immediately obvious, so we show it formally in the following observation.

Observation 1 For both f^{CSR} and f^{LSR}, if there is a successful agent bribery that makes two agents a_1^* and a_2^* being initially socially qualified ($\{a_1^*, a_2^*\} \subseteq K_1$ after the bribery but $\{a_1^*, a_2^*\} \cap K_1 = \emptyset$ before the bribery), then there is also a successful bribery of at most the same cost that makes just one agent from $\{a_1^*, a_2^*\}$ initially socially qualified ($|\{a_1^*, a_2^*\} \cap K_1| = 1$ after the bribery) yet both agents socially qualified at some point ($\{a_1^*, a_2^*\} \subseteq K_t$ for some $t \geq 1$ after the bribery).

Proof To obtain the bribery that makes just one agent from $\{a_1^*, a_2^*\}$ initially socially qualified, we do the following. For f^{LSR}, since $\{a_1^*, a_2^*\} \subseteq K_1$ after the bribery but $\{a_1^*, a_2^*\} \cap K_1 = \emptyset$ before the bribery, we must bribe both a_1^* and a_2^* in the original bribery. To make just a_1^* initially socially qualified, we can bribe the same as before with the following adjustments: We make a_1^* to qualify a_2^* and we make a_2^* to disqualify itself. It is easy to see that the new bribery bribes the same set of agents and after the new bribery $a_2^* \in K_2 \setminus K_1$, as a_2^* does not qualify itself but it is qualified by a_1^*.

For f^{CSR}, if one of the two agents, say without loss of generality a_2^*, is bribed in the original bribery, then bribe exactly in the same way as before but a_2^* is now not qualifying itself anymore. Otherwise, select an arbitrary bribed agent (which is by assumption different from a_1^* and a_2^*) that does not qualify a_2^* before the bribery and keep it disqualifying a_2^*. In both cases, we bribe the same set of agents as in the original bribery. While with the new bribery agent a_2^* does not become initially socially qualified ($a_2^* \notin K_1$), a_2^* is still qualified by agent a_1^* and thus becomes socially qualified in the second iteration ($a_2^* \in K_2$). It is easy to verify that the final outcome with respect to the agent's social qualification also remains the same for all other agents (though, the qualification process might be different). □

By the above observation, it is never necessary to make more than one agent initially socially qualified, which we will use in our algorithms.

Theorem 1 *Const+Dest-f^{LSR} \$Agent Bribery is solvable in time $O(n^3)$, while Const+Dest-f^{CSR} \$Agent Bribery is solvable in time $O(n^4)$.*

Proof We first provide the algorithms for the two rules and then, since both are based on finding a minimum weighted separator, we analyze their running time.

LSR Let us focus on f^{LSR} first. Let $L = \{a \in A^- \mid \varphi(a, a) = 1\}$ be the set of all agents in A^- who qualify themselves. We first bribe all agents $a \in L$ such that they disqualify everyone. To determine which further agents we want to bribe, we try to find a minimum weighted separator between the vertices A^+ and A^- in an auxiliary graph based on the qualification graph. More precisely, we add a source vertex σ to the qualification graph and connect σ to all vertices in A^+ and vertices with a self-loop. Moreover, we merge all vertices in A^- into one sink vertex τ (see Figure 3 for a visualization). We assign the weight $\rho(a)$ to each

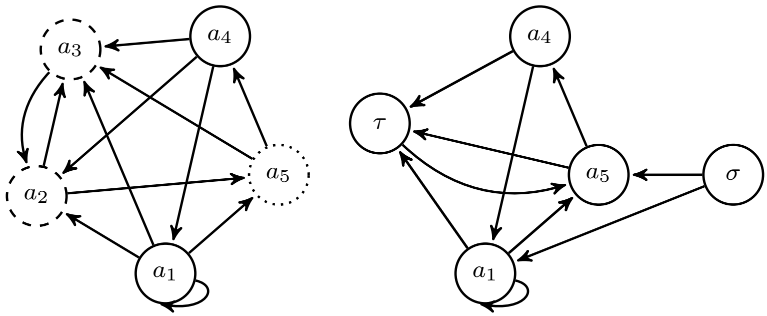

Fig. 3 Visualization of the transformation of a qualification graph where agents from A^+ are drawn as dotted circles and agents from A^- as dashed circles from Theorem 1 for f^{LSR}

vertex $a \in A \setminus A^-$. We call the resulting graph G. Subsequently, we calculate a minimum weighted (σ, τ)-separator A' in G:

If $A' = \emptyset$ and all agents $a \in A^+$ are already socially qualified, we are done. If $A' = \emptyset$ and this is not the case, we bribe an agent $a \in A \setminus A^-$ of the minimum weight and make it qualify itself and all agents in A^+ and disqualify all other agents. Now, all agents in A^+ are socially qualified, and since $A' = \emptyset$, all agents in A^- are socially disqualified.

If $A' \neq \emptyset$, as the existence of a (σ, τ)-path in the constructed graph implies that some agent in A^- will be socially qualified as soon as we make all agents in A^+ socially qualified, we need to bribe agents of total weight at least $\sum_{a \in A'} \rho(a)$ to destroy all (σ, τ)-paths.

In fact, we bribe all agents in A' such that they qualify themselves and all agents in A^+ and disqualify all other agents. Let G' be the graph which is obtained from G by deleting all arcs corresponding to the qualifications that got deleted and adding arcs corresponding to the qualifications that got inserted. By construction, there is no (σ, τ)-path in G'. Moreover, in the modified qualification profile, all agents in A^+ are socially qualified, as they are qualified by at least one agent $a \in A'$ qualifying itself. We claim that no agent in A^- can be socially qualified. Assume that there exists an agent $a^- \in A^-$ that is socially qualified. Then, there exists a path from some agent a^* which qualifies itself to a^- in the altered instance. If $a^* \notin A'$, then such a path implies that there exists an (σ, τ)-path in G' which cannot be the case. If $a^* \in A'$, since a^* only qualifies agents from A^+ and itself, there is also a path from some agent in A^+ to a^- in the altered instance. This also implies that there needs to exist an (σ, τ)-path in G' which again cannot be the case.

CSR For f^{CSR}, first assume that $A^+ = \emptyset$. If no agent from A^- is socially qualified, then we are done. Otherwise, we pick an arbitrary minimum-weight agent and make it disqualify everyone.

In the following, we assume that $A^+ \neq \emptyset$. If $A^* = \emptyset$, since $A^+ \neq \emptyset$, there has to be at least one agent who is qualified by everyone after the bribery. We guess an agent a^* that should be qualified by everyone in the end (we do this by iterating over all agents from A). Note that by Observation 1 one such agent will be enough. For each a^*, we bribe all agents that do not already qualify a^* such that they qualify all agents in $A^+ \cup \{a^*\}$ and disqualify all other agents. We still denote the set of agents who are qualified by everyone after the above bribery as A^*.

Then similar to the algorithm for f^{LSR}, we construct an auxiliary graph G. For this, we start with the underlying qualification graph. Subsequently, we insert a source vertex σ and connect it to all vertices in $A^+ \cup A^*$. Moreover, we contract all vertices from A^- into a sink vertex τ. As for f^{LSR}, we assign the weight $\rho(a)$ to each vertex $a \in A \setminus A^-$. Lastly, we remove all self-loops from G (notably we only remove the edges from G and do not bribe any agent in this step). Subsequently, we calculate a minimum weighted (σ, τ)-separator A' in G and bribe all agents in A' to qualify all agents in $A^+ \cup A^*$ and to disqualify all other agents. Subsequently, if not all agents in A^+ are socially qualified, we additionally bribe a cheapest agent $\widetilde{a}$ that is socially qualified after the already described bribery and let it qualify all agents in $A^+ \cup A^*$ and disqualify all other agents (note that such an agent needs to exist, as a^* is socially qualified).

First of all, note that the described bribery is always successful. In the case where no agent $\widetilde{a}$ in addition to the agents from A' is bribed, the presented bribery is also obviously optimal, as it is always necessary to bribe agents at a cost at least $\rho(A')$ to socially disqualify all agents from A^- (assuming that a^* was guessed correctly). Therefore it remains to consider the case where an agent $\widetilde{a}$ in addition to the agents A' was bribed. Then, no agent from the separator A' is socially qualified before the bribery of $\widetilde{a}$ (otherwise, all agents from A^+ would be socially qualified already after bribing the agents from A' resulting in no need to bribe an additional agent $\widetilde{a}$). This in turn implies that the (σ, τ)-separator A' touches no path between a vertex from A^* and τ, because there never was such a path. From the absence of a path between A^* and τ it follows that no agent that is part of a minimal (and also not minimum weighted) (σ, τ)-separator can be socially qualified before the bribery of $\widetilde{a}$. Thus, independent of which minimal separator we choose and how we bribe the agents in the separator, the set of agents being socially qualified after the bribery of these agents remains the same. Thus, we need at least cost $\rho(A')$ to separate the agents from A^+ and A^- in this case and at least cost $\rho(\widetilde{a})$ to make all agents from A^+ socially qualified. This minimum cost is exactly the cost of the described bribery.

Finally, to bound the running time, observe that each auxiliary graph G has $O(|A|)$ nodes and $O(|A|^2)$ arcs. Finding a minimum weighted (σ, τ)-separator can thus be done in $O(|A|^3)$ time by Fact 2. For f^{CSR}, we get an additional factor $|A|$, since we may have to iterate over all possible agents as the initially qualified agent a^*. □

From this it immediately follows that (priced) agent bribery is also polynomial-time solvable for all other considered variants including the previously completely unstudied case of exact bribery.

Corollary 1 *Exact-/Const-/Dest-f^{CSR}/f^{LSR} \$Agent Bribery is solvable in polynomial time.*

3.2 Link bribery

We now turn to the setting of link bribery and settle the complexity of the related decision problem for all bribery goals and both iterative rules. We start by proving that the problem is polynomial-time solvable for (priced) destructive bribery and (priced) exact bribery. For destructive bribery, similar to Theorem 1, we separate the agents that are initially socially qualified from the agents in A^- in the qualification graph. However, here, as we pay per changed qualification, we need to calculate a minimum weighted cut. For exact bribery, we need to separate the agents from A^- and A^+ while making sure that all agents from A^+ are socially qualified in the end.

Theorem 2 *Dest-f^{LSR} \$Link Bribery can be solved in $O(n^3)$ time, Dest-f^{CSR} \$Link Bribery can be solved in $O(n^4)$ time, Exact- f^{LSR} \$Link Bribery can be solved in $O(n^2)$ time, and Exact- f^{CSR} \$Link Bribery can be solved in $O(n^3)$ time.*

Proof We first prove the theorem for Dest-f^{LSR} \$Link Bribery.

Destructive LSR We start by bribing all agents from A^- who qualify themselves such that they all disqualify themselves and subtract the corresponding costs from the budget ℓ and update the profile φ. Let G be the qualification graph of the altered instance. We now consider a slightly modified version of G and calculate a minimum weighted cut to solve the problem. We start with G and assign to every arc e the weight $\rho(e)$. Subsequently, we add a source vertex σ to G and connect σ to each vertex a with a self-loop by an arc of weight $\rho((a,a))$; afterwards, self-loops can be removed. Additionally, we introduce a sink vertex τ and connect every vertex from A^- to the sink τ using arcs with weight $\ell + 1$. Now, we compute a minimum weighted (σ,τ)-cut $E' \subseteq E$ (using Fact 1). If the weight of a minimum weighted cut is at least $\ell + 1$, then we reject the instance, as no solution exists. If the weight of a minimum weighted (σ,τ)-cut is at most ℓ, then it cannot contain any arc that ends in τ and we remove the qualifications corresponding to the cut from the qualification profile: If $(a,a') \in E'$ for some $a, a' \in A$, we make a disqualify a'. If $(\sigma,a) \in E'$ for some $a \in A$, we make a disqualify itself.

On the one hand, after this bribery, no agent $a^- \in A^-$ is socially qualified, as the absence of a (σ,τ)-path implies that there does not exist a path from an agent qualifying itself to a^- in the altered qualification profile. On the other hand, the described bribery is optimal, as the existence of a (σ,τ)-path always implies that at least one agent in A^- is socially qualified in the end. *Destructive CSR*: For Dest-f^{CSR} \$Link Bribery, we need to follow a slightly more involved approach. For some $a \in A$, we denote by $\rho^{\zeta}(a)$ the cheapest possible cost of bribing some agent a' currently qualifying a to disqualify a, i.e. $\rho^{\zeta}(a) = \min_{a'' \in Q^+(a)} \rho(a'', a)$, and fix an agent $\zeta(a) \in \arg\min_{a'' \in Q^+(a)} \rho(a'', a)$. Let $d = \sum_{a \in A^*} \rho^{\zeta}(a)$ be the cheapest costs of bribing, for each agent a from the set A^* of agents that are qualified by everyone, at least one agent a' to disqualify a. (Note that, since we pay per modified qualification, costs for bribing the same agent twice, i.e. when $\zeta(a) = \zeta(a')$ for some $a \neq a'$, are indeed to be summed up. That is, the agents are handled independently from each other.) If $d \leq \ell$, we accept as we can make all agents from A^* socially disqualified (and thus no agent and, in particular, no agent from A^- is socially qualified) by bribing $\{(\zeta(a), a) \mid a \in A^*\}$. In the following, we assume $\ell < d$.

As $\ell < d$, after the bribery there will be at least one agent from A^* that remains to be qualified by all agents. We guess this agent (by iterating over all agents) and denote it as a^* in the following. Observe that if we have correctly guessed a^*, then an optimal bribery will never bribe agents from $A^* \setminus \{a^*\}$ to disqualify themselves. The only effect bribing an agent $\hat{a}$ from $A^* \setminus \{a^*\}$ to disqualify itself can have is that $\hat{a}$ may not be initially socially qualified anymore. However, agent a^* (assumed to be guessed correctly as an agent that remains initially socially qualified) anyway qualifies all agents from A^* including $\hat{a}$, except if we bribe a^* to disqualify $\hat{a}$. If this, however, is the case, then bribing $\hat{a}$ to disqualify itself was useless.

We can now compute a minimum weighted cut after modifying the qualification graph G as follows. Again, we start with G and assign to every arc e the weight $\rho(e)$. We add a source vertex σ and a sink vertex τ. We add an arc with weight $\ell + 1$ from σ to a^*. We connect every vertex from A^- by an arc of weight $\ell + 1$ to τ. Now, we compute a minimum weighted (σ, τ)-cut in the modified qualification graph. If the weight of a minimum weighted cut is at least $\ell + 1$, then we reject the current guess of a^* and continue with the next iteration, because from this it follows that any bribery that keeps a^* initially socially qualified is too expensive. If the weight of a minimum weighted (σ, τ)-cut is at most ℓ, then it cannot contain any arc that touches σ or τ. Hence, it is possible to delete all qualifications corresponding to arcs in the minimum cut to obtain a minimal bribery. Similar to above, the described bribery is successful and minimal assuming a^* was correctly guessed, as there exists an (σ, τ)-path in the constructed graph if and only if an agent from A^- is socially qualified.

Our algorithm keeps track and returns a cheapest bribery over all guesses of a^* (if any) and returns "no" if all guesses are rejected. It is easy to verify that in the latter case indeed no bribery at cost at most ℓ can exist.

We now turn to Exact- $f^{\text{LSR}}/f^{\text{CSR}}$ \$Link Bribery.

Exact LSR: For f^{LSR}, we start by making all agents A^- disqualify themselves if this is not already the case. Subsequently, we remove all qualifications from an agent in A^+ to an agent in A^-.

To ensure that all agents in A^+ are socially qualified, we reduce the problem to finding a spanning arborescence of minimum weight. For this, we build an auxiliary graph G and an edge weight function ω as follows. We initialize G as the underlying qualification graph restricted to the vertices from A^+. For a visualization of the following construction see Fig. 4. Let R be the set of vertices which have a self-loop or which are reachable from a vertex with a self-loop in the qualification graph. We merge all vertices from R into a single new vertex r; when $R = \emptyset$ this means to simply add a new vertex r. Let $a, a' \notin R$ be two vertices corresponding to not yet socially qualified agents. If there is an arc from a to a' in the qualification graph, then we set its weight under ω to 0. If there was no arc from a to a', then we create a new arc from a to a' in G and set its weight under ω to $\rho((a, a'))$ (the cost of bribing a to qualify a'). Furthermore, we create an arc from r to each vertex $a' \notin R$ and set its weight under ω to $\min\left(\rho((a', a')), \min_{a \in R} \rho((a, a'))\right)$ (the cheapest cost of either bribing a' to qualify itself or of bribing an agent from R to qualify a'). What remains is to find a spanning arborescence X with root r in G of minimum weight with respect to weight function ω using Fact 3.

We claim that a spanning arborescence of cost at most ℓ exists if and only if there is a bribery of cost at most ℓ that makes all agents from A^+ socially qualified. Assume that X is an arborescence of cost ℓ. From this, we construct a bribery as follows. For each arc $(a, a') \in X$ with $a, a' \notin R$, we bribe a to qualify a' if a does not already do so. This contributes $\rho((a, a')) = \omega((a, a'))$ to the cost of the bribery and the weight of the arborescence. For each arc (r, a') with $a' \notin R$, if $\rho((a', a')) \leq \min_{a \in R} \rho((a, a'))$, then we make a' qualify itself. This contributes $\rho((a', a')) = \omega((r, a'))$ to the cost of the bribery and the weight of the arborescence. Otherwise, let $a'' := \arg\min_{a \in R} \rho((a, a'))$. We bribe a'' to qualify a, which contributes $\rho((a, a'')) = \omega((r, a'))$ to the cost of the bribery and the weight of the arborescence. Thus, the cost of the constructed bribery is clearly the same as the weight of the computed spanning arborescence. Moreover, recall that all agents from R are already socially qualified before the bribery and as X is a spanning arborescence, there exists a path from r to every vertex from G in $G[X]$. Further observe that every agent a adjacent to r in $G[X]$ is socially qualified after the bribery, as $(r, a) \in X$ implies that we either bribe a to qualify itself or we bribe some agent who is already socially qualified before the bribery to qualify a. Combining these two observations it follows that all agents are socially qualified after the bribery.

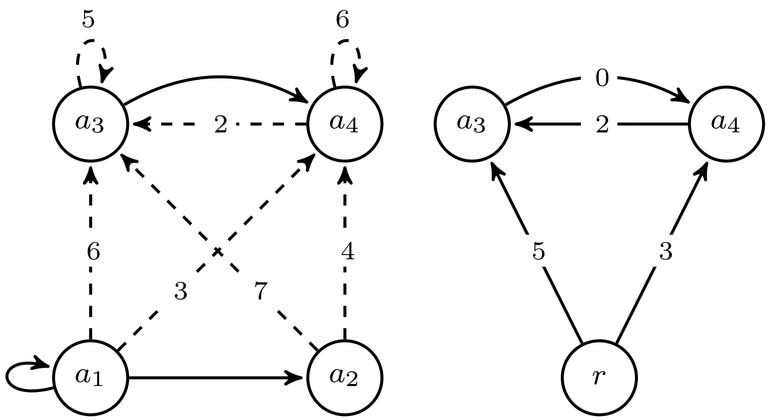

Fig. 4 Visualization of the transformation of an instance with $A = \{a_1, a_2, a_3, a_4\}$ and $A^+ = A$ of Exact-f^{LSR} \$Link Bribery from Theorem 2. On the left side, the solid arcs display the qualification graph, while the dashed arcs indicate the cost of adding the corresponding qualification. On the right side, the corresponding weighted graph constructed as described in Theorem 2 is displayed. The instance admits two solutions, that is, bribing a_3 to qualify itself or bribing a_4 to qualify a_3 and a_1 to qualify a_4

For the other direction, assume that there is a bribery of cost ℓ. From this, we construct a spanning arborescence X as follows. For each $a, a' \notin R$ such that a qualifies a' after the bribery we add (a, a') to X. If a already qualified a' before the bribery, this contributes 0 to the cost of the bribery and $\omega((a, a')) = 0$ to the weight of the arborescence; otherwise this contributes $\omega((a, a')) = \rho((a, a'))$ to the cost of the bribery and the weight of the arborescence. For each $a \notin R$ that qualifies itself after the bribery, we add (r, a) to X (note that by the definition of R, a cannot qualify itself before the bribery). The edge (r, a) contributes $\omega((r, a)) \leq \rho((a, a))$ to the weight of X and $\rho((a, a))$ to the cost of the bribery. For each pair of agents $a \in R$ and $a' \notin R$ such that a qualifies a' after the bribery, we add (r, a) to X (note that by the definition of R, a cannot qualify a' before the bribery). The edge (r, a) contributes $\omega((r, a)) \leq \rho((a, a'))$ to the weight of X and $\rho((a, a'))$ to the cost of the bribery. The weight of X is clearly at most the cost of the bribery. Moreover, for the sake of contradiction assume that X is not a spanning arborescence. Then, there is some agent $a \notin R$ who is not connected to r in $G[X]$. However, this implies that a cannot be socially qualified after the bribery. *Exact CSR*: For f^{CSR}, if A^+ is empty, we are either done or there exist socially qualified agents. In the latter case, we denote by $\rho^{\zeta}(a)$ the cheapest possible cost of bribing some agent a' to disqualify a, i.e., $\rho^{\zeta}(a) = \min_{a'' \in Q^+(a)} \rho(a'', a)$, and fix an agent $\zeta(a) \in \arg\min_{a'' \in Q^+(a)} \rho(a'', a)$. By this definition, removing all qualifications from $\{(\zeta(a), a) \mid a \in A^*\}$ gives an optimal bribery.

Otherwise, if A^+ is nonempty, we start by deleting all qualifications from agents in A^+ to agents in A^-. Subsequently, similar to the above, we try to make all agents in A^+ socially qualified by computing a spanning arborescence of the minimum weight of a graph based on the qualification graph. Handling the set of initially socially qualified agents, however, will be again more complicated (similar as for Dest-f^{CSR} \$Link Bribery). Observe that if A^+ is nonempty, then after the bribery the set of initially socially qualified agents (which are qualified by all agents) cannot be empty. If A^* is nonempty without any bribery, then we continue very similarly as for f^{LSR}: Again, to ensure that all agents in A^+ are socially qualified, we consider the underlying qualification graph restricted to the vertices from A^+

and slightly modify it to solve the problem by finding a spanning arborescence of the minimum weight. Let R be the set of vertices which are initially qualified by all agents or which are reachable from a vertex initially qualified by all agents. We merge all vertices from R into a single new vertex r. Let $a, a' \notin R$ be two vertices corresponding to not yet socially qualified agents. If there is an arc from a to a' in the qualification graph, then we set its weight to 0. If there was no arc from a to a', we create a new arc from a to a' and set its weight to $\rho((a, a'))$ (the cost of bribing a to qualify a'). Furthermore, we create an arc from r to each vertex $a' \notin R$ and set the weight to $\min_{a \in R} \rho((a, a'))$ (bribing an agent from R to qualify a').[2] What remains is to find a spanning arborescence of the minimum weight with root r (again by Fact 3). It is easy to verify that a spanning arborescence of the minimum weight corresponds to a cheapest possible bribery that makes all agents from A^+ socially qualified and vice versa.

Finally, let A^* be empty (and A^+ be still nonempty). We continue by guessing an agent a^* in A^+ that is qualified by everyone (by iterating over all agents from A^+). We make a^* qualified by all agents and proceed as in the case where A^* was nonempty.

For the running time bounds, recall that the constructed graph will have $O(n^2)$ edges and $O(n)$ nodes. Moreover, for f^{CSR}, we get an additional factor n for iterating over all possible choices of the initially qualified agent a^*. Applying the known running time bounds for computing a minimum weighted cut or a spanning arborescence of the minimum weight from Fact 1 and Fact 3 gives the bounds claimed in the theorem. □

In contrast to this, the corresponding problem for constructive bribery is NP-complete and even W[2]-hard parameterized by the budget ℓ. This difference in the complexity of the problem for constructive bribery and destructive bribery is somewhat surprising, as their complexity is the same in the case of agent bribery. We show the hardness of Const-f^{CSR}/f^{LSR} Link Bribery by a reduction from Set Cover, which is NP-complete and W[2]-hard with respect to the requested size of the cover [9]. The general idea of the reduction is to introduce one agent for each element (these form the set A^+) and for each set, where each set-agent qualifies the agents corresponding to the elements in the set. Notably, as all our hardness results, the hardness holds already for unit costs.

Theorem 3 *Const-f^{CSR}/f^{LSR} Link Bribery is NP-complete and W[2]-hard with respect to ℓ.*

Proof We reduce from the Set Cover problem, where given a number k, a universe of elements U, and a set of subsets of $\mathcal{S} \subseteq 2^U$ with $\bigcup \mathcal{S} = U$, one has to decide whether there exists a subset $\mathcal{S}' \subseteq \mathcal{S}$ of size at most k which covers U.

We prove the theorem for f^{CSR} and f^{LSR} by the same reduction from Set Cover. Given an instance $(\mathcal{S}, U, k)$ of Set Cover, we construct an instance of Const-f^{CSR}/f^{CSR} Link Bribery with budget $\ell = k$, agents $A = \mathcal{S} \cup U \cup \{a^*\}$ and $A^+ = U$. The qualification profile is created as follows: a^* only qualifies itself. Each $S \in \mathcal{S}$ qualifies all $a \in S$ and a^* and disqualifies all other agents. Each $a \in U$ qualifies a^* and disqualifies all other agents. Consequently, without any bribery a^* and no one else is socially qualified for both f^{CSR} and f^{LSR}. Note that the parameter ℓ in the constructed instance is linearly bounded in the parameter

[2] Note that here, opposed to the case for f^{LSR}, it cannot be beneficial to bribe an agent a to qualify itself and make it qualified by all agents so that it would be part of the set of initially socially qualified agents, because either the agent is already qualified by some agent from R (which implies that a is already socially qualified) or this is at most as expensive as bribing the cheapest agent from R to qualify a.

k of the given instance (in fact, $k = \ell$). We now prove that there exists a successful bribery of cost at most ℓ in the constructed instance if and only if there exists a set cover of size at most k in the given Set Cover instance.

($\Rightarrow$) Assume that $\mathcal{S}'$ is a cover of U of size at most k. Then, the briber bribes a^* such that a^* qualifies all agents in $\mathcal{S}'$ which requires at most ℓ insertions. As $\mathcal{S}'$ is a cover of U, for each $a \in U$ there is some $S \in \mathcal{S}'$ with $a \in S$. Before the bribery a^* is qualified by everyone and S qualifies a. After the bribery a^* qualifies S. Thus, a and, in fact, all agents from A^+ are socially qualified after the bribery.

($\Leftarrow$) Assuming that there exists a successful bribery resulting in the qualification profile φ', there also exists a successful bribery where only a^* is bribed and only links from a^* to some agent from $\mathcal{S}$ are inserted, as it is possible to replace every inserted qualification (a, a') for two $a, a' \in A$ with (a^*, a') and every arc pointing to some vertex $u \in U$ by an arc from a^* to some vertex from $\mathcal{S}$ which contains u. We claim that $\mathcal{S}' := \{S \in \mathcal{S} \mid \varphi'(a^*, S) = 1\}$ is a solution to the given Set Cover instance: First, as the budget for the bribery is ℓ, it needs to hold that $|\mathcal{S}'| = \ell = k$. For the sake of contradiction, assume that there is some $a \in U$ such that a is not covered by $\mathcal{S}'$. Then, a is also not socially qualified after the bribery, as it is only qualified by agents from $\mathcal{S} \setminus \mathcal{S}'$, which are in turn not qualified by anyone. We reached a contradiction. □

Apart from the parameter budget ℓ, which might be small in most applications as only agents which are close to the boundary of being socially (dis)qualified might be interested in their precise margin, another natural parameter is the set of agents we want to make socially qualified, i.e., $|A^+|$. This parameter may also be not too large in most applications, as one may only be interested in the classification of a limited number of agents. In contrast to the negative parameterized result for ℓ, by reducing the problem to the Weighted Directed Steiner Tree (WDST) problem, it is possible to prove that Const-f^{CSR} /f^{LSR} \$Link Bribery is FPT with respect to $|A^+|$.

Theorem 4 *Const-f^{CSR}/f^{LSR} \$Link Bribery is FPT with respect to* $|A^+|$.

Proof We start by considering f^{LSR} before we turn to f^{CSR}.

LSR We examine a close connection between Const-f^{LSR} \$Link Bribery and WDST, which is FPT with respect to the size of the terminal set from Fact 4. For a visualization of the following construction see Fig. 5. Let I be an instance of Const-f^{LSR} \$Link Bribery and consider the qualification graph where we add one additional root vertex r. Moreover, for each existing arc, set its weight to 0 and replace every self-loop (a, a) for some $a \in A$ by an arc from r to a (again of weight 0). For every pair of vertices a and a' where a disqualifies a', we create an additional arc from a to a' of weight $\rho((a, a'))$. For every agent a that does not initially qualify itself, we add an arc from r to a with weight $\rho((a, a))$. Finally, we mark all vertices in A^+ as terminals. We call the resulting graph G'.

Now, it is easy to verify that there exists a subset of arcs E^* in G' of total weight at most ℓ ensuring that there is a directed path from the root r to every terminal $t \in T$ in G' if and only if there is a successful bribery of cost ℓ for I.

For the "if direction", consider a successful bribery for I. In particular, consider the graph G' constructed above for the WDST instance and set the weight of every arc corresponding to a bribed agent pair to zero. Now, it is easy to verify that the modified WDST instance must have a solution of total weight 0 (just consider an arbitrary spanning

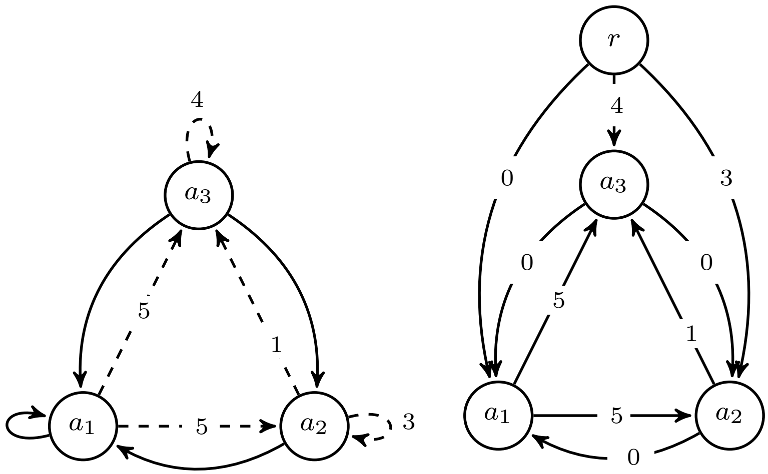

Fig. 5 Visualization of the transformation of an instance with $A = \{a_1, a_2, a_3\}$ and $A^+ = A$ of CONST-f^{LSR} \$LINK BRIBERY from Theorem 4. On the left side, the solid arcs display the qualification graph, while the dashed arcs indicate the cost of adding the corresponding qualification. On the right side, the corresponding weighted graph constructed as described in Theorem 4 is displayed. The instance admits two solutions, that is, bribing a_3 to qualify itself or bribing a_2 to qualify itself and a_2 to qualify a_3

arborescence of minimum weight with root r). Clearly, the set of arcs with modified weights that may be part of the selected arborescence has weight at most ℓ in G'.

For the "only if direction", consider a subset of arcs E^* of total weight at most ℓ ensuring that there is a directed path from the root r to every terminal $t \in T$ in G'. Now, observe that bribing for every arc $(a, a') \in E^*$ of nonzero weight the agent a to qualify agent a' (and a' to qualify itself if $a = r$) is indeed of total cost ℓ and yields a successful bribery, as a path from r to some terminal t implies that the vertex t will be socially qualified.

CSR For f^{CSR}, we start by guessing an agent a^* that is qualified by everyone (by iterating over all agents from A, where if there already exists such an agent, we pick it), bribe all agents that currently disqualify a^* to qualify it, and subtract the costs from the budget. Then, we apply an algorithm very similar to the above case, where the corresponding auxiliary graph is even a little bit simpler: In particular, consider the qualification graph where we add one additional root vertex r. Moreover, for each existing arc, set its weight to 0 and remove every self-loop. For every pair of vertices a and a' where a disqualifies a', we create an additional arc from a to a' of weight $\rho((a, a'))$. For every agent a that is qualified by everyone (including at least a^*), we add an arc from r to a with weight 0. Finally, we mark all vertices from A^+ as terminals and set p to ℓ. We call the resulting graph G'. The correctness proof works analogous to the case of f^{LSR}. □

The hardness results for constructive bribery imply that constructive+destructive bribery is also NP-hard and W[2]-hard with respect to ℓ. Utilizing a slightly more involved reduction from EXACT COVER BY 3 SETS, it is even possible to show that the NP-hardness of constructive+destructive bribery extends to the case where the briber is only allowed to delete qualifications. This may be surprising, as destructive bribery alone is polynomial-time solvable.

Proposition 1 *Const+Dest-f^{CSR}/f^{LSR} Link Bribery remains NP-complete even if the briber is only allowed to delete qualifications.*

Proof We prove the proposition for both rules at the same time by a reduction from Exact Cover By 3 Sets, a version of set cover where every set contains exactly three elements and every element appears in exactly three sets. Given an instance $(\mathcal{S},\mathcal{U})$ of X3C, where $|\mathcal{U}| = |\mathcal{S}| = 3m$ for some $m \in \mathbb{N}$, let $A = \mathcal{S} \cup \mathcal{U} \cup \{a^*\} \cup \{a'\} \cup \{a''\}$, $A^+ = \mathcal{U} \cup \{a^*\}$ and $A^- = \{a'\} \cup \{a''\}$. We construct the qualification profile as follows. Everyone qualifies a^* (including a^* itself). Agent a^* qualifies all agents from $\mathcal{S}$. All agents from $\mathcal{S}$ qualify a' and a'' and the three agents corresponding to their elements. We set the budget to $\ell = 4m$ and allow the briber only to delete qualifications. We now prove that there exists a successful bribery of cost at most ℓ in the constructed instance if and only if there exists an exact cover in the given X3C instance. See Fig. 6 for a visualization of the forward direction.

($\Rightarrow$) Assume that $\mathcal{S}' \subseteq \mathcal{S}$ is a cover of $\mathcal{U}$ with $|\mathcal{S}'| = m$. Then, we bribe a^* to disqualify all agents from $\mathcal{S} \setminus \mathcal{S}'$ (there are $2m$ of them). Moreover, we make all agents from $\mathcal{S}'$ disqualify a' and a'' (there are $2m$ of them). In the resulting qualification profile, all agents from $\mathcal{S}'$ are still socially qualified, while all agents from $\mathcal{S} \setminus \mathcal{S}'$ are not socially qualified. Thereby, all agents in $\mathcal{U}$ are socially qualified, as they are qualified by one agent from $\mathcal{S}'$, while none of a' and a'' is socially qualified, as they are only qualified by members of $\mathcal{S} \setminus \mathcal{S}'$.

($\Leftarrow$) Let us assume that we found a successful bribery of cost at most ℓ. Then, in the resulting qualification profile, the subset $\mathcal{S}' \subseteq \mathcal{S}$ of agents which are still socially qualified needs to cover all agents from $\mathcal{U}$. We claim that $|\mathcal{S}'| \leq m$ and, in fact $|\mathcal{S}'| = m$, and thereby that $\mathcal{S}'$ induces a solution of the X3C-instance. For the sake of contradiction, assume that $|\mathcal{S}'| > m$. In this case, to ensure that neither a' nor a'' are socially qualified, a budget of at least $|\mathcal{S} \setminus \mathcal{S}'|$ was needed to prevent the existence of a path from a^* to a' or a'' visiting an agent from $\mathcal{S} \setminus \mathcal{S}'$. Moreover, a budget of $2|\mathcal{S}'|$ was needed to make all agents in $\mathcal{S}'$ disqualify a' and a''. This results in an overall cost of $2|\mathcal{S}'| + |\mathcal{S} \setminus \mathcal{S}'|$, which is larger than $4m$ as $|\mathcal{S}'| > m$. □

The remaining question is to pinpoint the parameterized complexity of constructive+destructive bribery with respect to $|A^+| + |A^-|$. Despite the fact that the FPT result for constructive bribery suggests that it may be possible to prove fixed-parameter tractability for this case, we were not even able to prove that this problem lies in XP, leaving this as an intriguing open problem for future work.

4 Consent rule

For the consent rule, all computational problems in the link bribery setting are easy to solve. The main reasoning here is that every possible link bribery action can influence the social qualification of only one agent, namely, the sink of the corresponding arc.

For the consent rule, we distinguish four types of outcomes for some agent a: (1a) agent a qualifies itself and at least $s-1$ other agents also qualify a, (1b) agent a qualifies itself and less than $s-1$ other agents qualify a, (2a) agent a disqualifies itself and at least $t-1$ other agents disqualify a, and (2b) agent a disqualifies itself and less than $t-1$ other agents disqualify a. Moreover, an agent that shall be socially qualified must end up

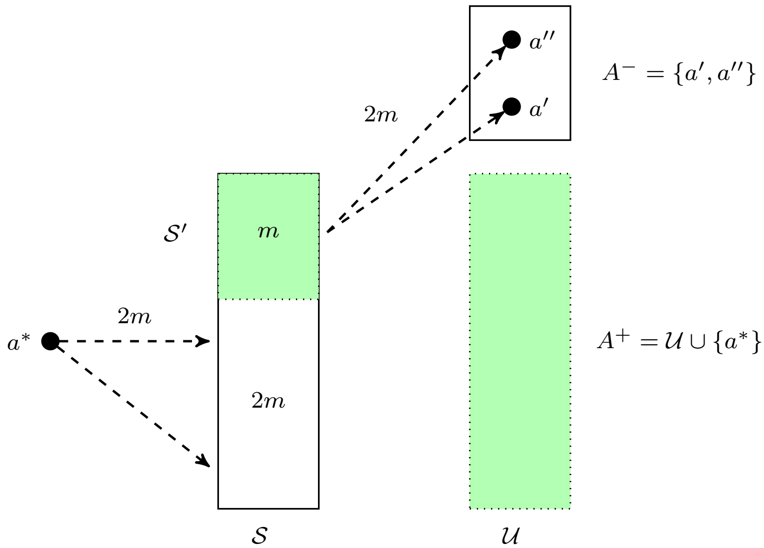

Fig. 6 Visualization of the reduction from Proposition 1 for a successful bribery with budget $\ell = 4m$. The $4m$ dashed arrows will be deleted in the bribery. The remaining arrows are omitted. The socially qualified agents after the bribery are agents in the green dotted area together with a^*

in case (1a) or (2b) and an agent that shall be socially disqualified must end up in case (1b) or (2a).

For each of these cases, computing the cheapest bribery such that the case applies for an agent a in the modified profile is easy: If this is not already the case, first bribe a to (dis)qualify itself. Then, sort the missing (dis)qualifications of all other agents by their costs and iteratively pick the cheapest one and make it (dis)qualify a until the condition stated in the specific case is met and a becomes socially (dis)qualified. This way, for constructive bribery, we can compute for each agent from A^+ the costs for the two cases (1a) or (2b) and choose the cheaper one, while for destructive bribery we can compute for each agent from A^- the costs for the two cases (1b) or (2a) and choose the cheaper one. As mentioned above, each agent can be handled independently. Thus, we need to sort at most $2(|A^+| + |A^-|)$ times at most $|A|$ arc weights, ending up with the following.

Observation 2 Const+Dest-$f^{(s,t)}$ \$Link Bribery can be solved in $O((|A^+| + |A^-|) \cdot |A| \log |A|)$ time.

Now, turning to agent bribery, we start by focusing on Const-$f^{(s,t)}$ (\$)Agent Bribery in Subsect. 4.1 and explain how the results for constructive bribery translate to our other three bribery goals in Subsect. 4.2.

4.1 Constructive agent bribery

Erdélyi et al. [17, Theorem 2] proved that Const-$f^{(s,t)}$ Agent Bribery is NP-complete for all $s \geq 1$ and $t \geq 2$, which is why we focus on the parameterized complexity of this NP-complete problem. Studying bribery problems for the consent rule, s and t are natural parameters to consider, as at least one of these parameters may be small in most applications: In problems where socially qualified agents acquire a privilege, t should be small, while for problems where social qualification implies some obligation or duty, s should be small. However, the hardness result of Erdélyi et al. [17] directly implies that CONST-$f^{(s,t)}$ AGENT BRIBERY is para-NP-hard with respect to $s + t$. However, the reduction has no implications on the parameterized complexity of the problem with respect to ℓ and $|A^+|$. In the following, we conduct a parameterized analysis of CONST-$f^{(s,t)}$ (\$)AGENT BRIBERY with respect to s, t, ℓ and $|A^+|$, and then, in Subsect. 4.2, we explain how to adapt our results to the other three bribery goals considered.

Interestingly, constructive agent bribery for consent rules can be seen as a variant of a set cover problem: Every agent from A^+ that is not initially socially qualified needs to gain a certain number of qualifications, that is, the agent needs to be covered a certain number of times, and bribing an agent $a \in A$ corresponds to covering all agents from A^+ that agent a initially disqualified.[3] In the following, we start by analyzing the parameterized complexity of CONST-$f^{(s,t)}$ (\$)AGENT BRIBERY with respect to the parameters s, t, and ℓ. Subsequently, we analyze the influence of the number of agents that we need to make socially qualified, i.e., $|A^+|$, on the complexity of the problem.

4.1.1 Parameterized complexity with respect to *s*, *t*, and ℓ

Erdélyi et al. [17] proved that CONST-$f^{(s,t)}$ AGENT BRIBERY is in XP with respect to s if $t = 1$. However, they left open whether this problem is FPT or W[1]-hard. Moreover, there exists a trivial brute force XP-algorithm for ℓ, while it is again open whether the problem is FPT or W[1]-hard with respect to ℓ. We answer both questions negatively in the following theorem:

Theorem 5 *CONST-$f^{(s,t)}$ AGENT BRIBERY is W[1]-hard with respect to $s + \ell$ even if $t = 1$.*

Proof We reduce from INDEPENDENT SET $(G = (V, E), k)$, where the question is to decide whether there exists a set of pairwise non-adjacent vertices of size k in the given graph G. INDEPENDENT SET is W[1]-hard with respect to k [12]. Given an instance of the problem, in the group identification problem, we insert for each vertex $v \in V$, one *vertex-agent* a_v and, for each edge $\{u, v\} \in E$, one *edge-agent* $a_{u,v}$ and one designated *dummy agent* $\tilde{a}_{u,v}$. We set $A^+ = \{a_{u,v}, \tilde{a}_{u,v} \mid \{u, v\} \in E\}$. All dummy agents qualify only themselves. For each $\{u, v\} \in E$, $a_{u,v}$ qualifies $\tilde{a}_{u,v}$ and itself, while it is only qualified by the two agents a_u and a_v. See Fig. 7 for a visualization. We set $s = k + 2$, $t = 1$ and $\ell = k$. Note that the combined parameter $s + \ell$ of the constructed instance is linearly bounded in the parameter k of the given instance (in fact, $s + \ell = 2k + 2$). We now prove that there exists a successful bribery of cost at most ℓ in the constructed instance if and only if there exists an independent set of size k in G. Note that all dummy agents need at least k additional qualifications to be socially qualified, while all edge-agents need at least $k - 1$ additional qualifications.

[3] The number of qualifications an agent $a \in A^+$ that initially disqualifies itself needs to gain obviously depends on whether a is bribed (to qualify itself). However, this can be easily modeled when viewing the problem as a variant of set cover with demands: Set the initial demand, i.e., the number of sets which need to cover a, of a to $|Q^-(a)| - (t - 1)$ and insert $n - (t - 1) - s$ copies of a in the set that corresponds to bribing a. Formally, this is only possible if we have multisets that allow for multiple copies of an agent in one set. So we, in fact, need to consider a multiset cover variant.

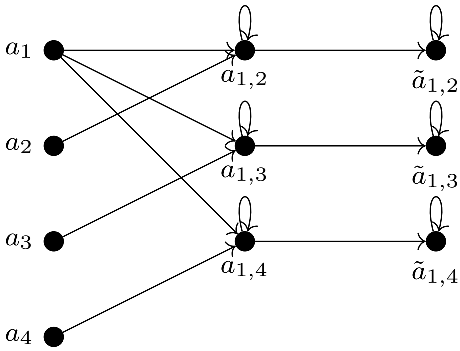

Fig. 7 Visualization of the reduction from Theorem 5 for a given graph $G = (\{1,2,3,4\}, \{\{1,2\}, \{1,3\}, \{1,4\}\})$

($\Rightarrow$) Assume that $V' \subseteq V$ is an independent set of size k in G. Then, we bribe all vertex-agents that correspond to the agents in V' and let them qualify everyone. Thereby, every edge-agent gains at least $k-1$ additional qualifications, as every edge is only touched by at most one vertex in V' and thus each edge-agent was qualified by at most one agent a_v with $v \in V'$ before the bribery. Moreover, as no vertex-agent qualifies a dummy agent, all dummy agents gain k qualifications.

($\Leftarrow$) Assume we are given a successful bribery $A' \subseteq A$ consisting of k agents. We claim that only vertex-agents can be part of A'. Assuming that for some $\{u,v\} \in E$ either $a_{u,v}$ or $\tilde{a}_{u,v}$ are part of A', then $\tilde{a}_{u,v}$ cannot be socially qualified after the bribery, as it gained at most $k-1$ qualifications over the bribery. It additionally holds that, for each $\{u,v\} \in E$, at most one of a_v and a_u can be part of A', as $a_{u,v}$ gained at least $k-1$ additional qualifications by the bribery. Thereby, A' consists of k vertex-agents which initially do not both qualify the same edge-agent. From this it follows that $V' = \{v \in V \mid a_v \in A'\}$ is an independent set of size k in G. ☐

Remark 1 In the conference version of this paper [2], we observed that it is vital for the correctness of the reduction that no non-vertex-agent can be bribed. This is ensured by the existence of dummy agents who are all missing exactly ℓ qualifications to become socially qualified. Motivated by this, we wrongly claimed that Const-$f^{(s,t)}$ Agent Bribery with $t = 1$ can be solved in $g(s)\mathcal{O}(n^2)$ for some function g if the number of agents that are missing exactly ℓ qualifications to become socially qualified can be bounded in a function of s. In fact, this is not the case and the reduction from Theorem 5 can be easily adapted to show W[1]-hardness even if only one agent is missing exactly ℓ qualifications: We modify the construction by inserting an additional agent a^* who qualifies everyone except itself and add a^* to A^+. Further, we set $s = k+3$, $t = 1$, and $\ell = k+1$. Note that we need to bribe a^* to qualify itself and that afterwards all dummy agents are missing k qualifications to become socially qualified and that the remaining budget is k.

From Theorem 5, it follows that combining ℓ with parameters s and t is not enough to achieve fixed-parameter tractability. Even fixing t as constant and considering ℓ and s as parameter remains hard. The only hope left when considering these three parameters is to treat s as a constant. Doing this, it turns out that parameterizing the problem by ℓ alone is not enough to achieve fixed-parameter tractability even in the restricted case where $s = 1$. This can be shown by a reduction from Dominating Set where given a graph $G = (V,E)$ and an integer k, the task is to decide whether there exists a subset of vertices $V' \subseteq V$ of size at most k such that each vertex $v \in V$ is either part of V' or adjacent to at least one vertex from V'. Dominating Set parameterized by k is W[2]-hard [12].

Theorem 6 *Const-$f^{(s,t)}$ Agent Bribery is W[2]-hard with respect to ℓ even if $s = 1$.*

Proof We prove this theorem by a reduction from Dominating Set. Given an undirected graph $G = (V, E)$, for $v \in V$, let $d(v)$ denote the degree of v in G and let $d^* = \max_{v \in V} d(v)$ denote the maximum degree of a vertex in G. We construct a corresponding group identification problem as follows.

For each $v \in V$, we introduce a so-called *vertex-agent* a_v who qualifies everyone except itself and the agents corresponding to its neighbors in G. Moreover, for each $v \in V$, we introduce $d^* - d(v)$ dummy agents qualifying everyone except v. We set A^+ to be the set of all vertex-agents, $s = 1$, $t = d^* + 1$ and $\ell = k$. Note that the parameter ℓ of the constructed instance is linearly bounded in the parameter k of the given instance (in fact, $\ell = k$).

Initially, no vertex-agent is socially qualified, as each agent a_v disqualifies itself and is disqualified by its $d(v)$ neighbors and $d^* - d(v)$ designated dummy agents. During the bribing, each agent needs to gain one additional qualification either from itself or from another agent to be included in the set of socially qualified agents. We now prove that there exists a successful bribery of cost at most ℓ in the constructed instance if and only if there exists a dominating set of size at most k in G.

($\Rightarrow$) Assume that $V' \subseteq V$ is a dominating set of G. Then, we bribe all vertex-agents corresponding to vertices in V' such that they qualify everyone. Then, all agents in V' are socially qualified as they qualify themselves. Moreover, for each agent a_v with $v \in V \setminus V'$, at least one of the vertex-agents corresponding to one of v's neighbors no longer disqualify a_v. Thus, a_v is disqualified by itself and by at most $d^* - 1$ other agents and is, thereby, socially qualified.

($\Leftarrow$) Assume that we have found a successful bribery bribing agents $A' \subseteq A$. Then, without loss of generality, we can assume that no dummy agent is part of A', as instead of bribing a dummy agent it is always possible to bribe the corresponding vertex-agent. Let $V' = \{v \in V \mid a_v \in A'\}$. As every vertex-agent a_v needs to either qualify itself or gain an additional qualification by at least one vertex agent corresponding to one of v's neighbors to be socially qualified in the end, V' is a dominating set of G. □

Parameterized by $\ell + t$, however, while keeping s as a constant, the problem becomes, in fact, fixed-parameter tractable:

Theorem 7 *Const-$f^{(s,t)}$ \$Agent Bribery is FPT with respect to $\ell + t$ when s is a constant.*

Proof We first prove the theorem for Const-$f^{(s,t)}$ Agent Bribery and afterwards explain how to extend the algorithm for the priced version of this problem. Let $A' \subseteq A$ denote the subset of agents that we want to bribe. Note that it is always optimal to bribe all agents from A' to qualify everyone including themselves (note that it is, in particular, optimal to make an agent from A^+ qualify itself because we assume that $s + t \leq n + 2$; thus, an agent who qualifies itself but is socially disqualified cannot become socially qualified by disqualifying itself). We distinguish the case where $\ell < s$ and $\ell \geq s$. In the former case, we simply iterate over all ℓ

-subsets of A, bribe them to qualify everyone and check whether the bribery was successful. This leads to an overall running time of $\mathcal{O}(n^{s+2})$, as there are $n^\ell \leq n^s$ many ℓ-subsets of A and for each subset checking whether it is sufficient to bribe the agents from this subset takes $\mathcal{O}(n^2)$. It remains to consider the latter case where $\ell \geq s$. Here, every agent qualifying itself will be socially qualified in the end because we bribe all agents from A' to qualify everyone and, thereby, each agent will be qualified by at least s agents after the bribery. Thus, this problem reduces to the case of $s = 1$ for which we describe the algorithm in the following.

For each $a \in A$, let $y_a = \max(0, |Q^-(a)| - (t-1))$ denote the number of additional qualifications a needs to get to become socially qualified without qualifying itself. For every $a \in A^+$ with $y_a > \ell$, it needs to hold that $a \in A'$. More generally, for all $a \in A^+$, either $a \in A'$ or a y_a-subset of $Q^-(a)$ needs to be part of A'. These ideas give rise to Algorithm 1, which we call as CalcB(A, φ, A^+, $\emptyset$, ℓ, y_{a_1}, ..., y_{a_n}). This algorithm identifies at each step all agents that need to qualify themselves because the remaining budget is not sufficient otherwise and all agents that are already socially qualified after bribing the agents from A'. Subsequently, an arbitrary agent from A^+ which is not already socially qualified after bribing the agents from A' is selected and we branch over bribing it or one of the agents disqualifying it.

If the algorithm returns a solution, this solution is clearly correct, as agents from A^+ are only deleted if they become socially qualified after bribing the agents from A' and we correctly keep track of the remaining budget. Thus, it remains to prove that if there exists at least one solution, the algorithm will always find one: Assume that $S \subseteq A$ is a subset of agents such that after making all agents from S qualify everyone, all agents from A^+ are socially qualified. We now argue that there exists at least one branch in the execution of the algorithm where it always holds that $A' \subseteq S$. We prove this by induction on the size of A' during the run of the algorithm. Before inserting any agents in A', the condition is trivially fulfilled. Let us assume that A' contains i agents and it holds that $A' \subseteq S$. There exist two possibilities where the $i+1$th agent $\tilde{a}$ is added to A' in the algorithm. Assuming that $\tilde{a}$ is added to A' in line 2, then it needs to hold that $\tilde{a}$ is also part of S because assuming that all agents from A' are bribed, the remaining budget is not high enough to bribe enough agents such that $\tilde{a}$ gets socially qualified in the end without qualifying itself. Otherwise, $\tilde{a}$ gets inserted into A' in line 9. In line 9, we branch over bribing an agent a^*, who is still missing at least one qualification to get socially qualified after bribing the agents from A', and bribing one of the agents disqualifying a^*. As we have assumed that $A' \subseteq S$, it needs to hold that either a^* or an agent disqualifying a^* is part of $S \setminus A'$, as otherwise a^* cannot be socially qualified after bribing the agents from S. Thus, for at least one branch, it still holds that $A' \subseteq S$ after inserting the $i+1$th agent into A'.

The depth of the recursion is bounded by ℓ. Moreover, the branching factor is bounded by the maximum $|Q^-(a^*)| + 1$ taken over all agents a^* picked in Line 7. As for every a^* picked in Line 7 it needs to hold that $y_{a^*} \leq \ell$ (because otherwise a^* would have been deleted from A^+ in line 2), it follows that $|Q^-(a^*)| \leq \ell + (t-1)$. Thereby, the overall running time of the algorithm lies in $\mathcal{O}(n^2(\ell + t)^\ell)$, as our search tree contains at most $(\ell + (t-1))^\ell$ states and for each state the procedure specified in Algorithm 1 runs in $\mathcal{O}(n^2)$.

Algorithm 1 CalcB(A, φ, A^+, A', p, $y_{a_1}, \ldots, y_{a_n}$)

Input: Agents A, qualification profile φ, subset of agents A^+ and A', remaining budget p, and, for each $i \in [n]$, number y_{a_i} of additional qualifications a_i needs to get to become socially qualified without qualifying itself
Output: Set of agents A' to bribe
1: **for** $a \in A^+$ with $y_a - |A' \cap Q^-_\varphi(a)| > p$ **do**
2: $A' = A' \cup \{a\}$; $A^+ = A^+ \setminus \{a\}$; $p = p - 1$;
3: **for** $a \in A^+$ with $a \in A'$ or $|A' \cap Q^-_\varphi(a)| \geq y_a$ **do**
4: $A^+ = A^+ \setminus \{a\}$;
5: **if** $p < 0$ **then return** Reject
6: **if** $A^+ = \emptyset$ **then return** A'
7: Pick an arbitrary $a^* \in A^+$
8: **for** $a \in \{a^*\} \cup (Q^-_\varphi(a^*) \setminus A')$ **do**
9: **return** CalcB(A, φ, A^+, $A' \cup \{a\}$, $p - 1$, $y_{a_1}, \ldots, y_{a_n}$)

Priced Variant We now show the theorem for CONST-$f^{(s,t)}$ \$AGENT BRIBERY. We still distinguish the case where $\ell < s$ and $\ell \geq s$. In the former case, we simply iterate over all subsets of A that cost at most ℓ, bribe them to qualify everyone and check whether the bribery was successful. Since each agent has a positive integer price, the size of subsets of A that cost at most ℓ is upper bounded by $\ell < s$, and hence, the running time is $\mathcal{O}(n^{s+2})$.

In the latter case where $\ell \geq s$, compared to the unpriced version, it does not hold anymore that every agent qualifying itself will be socially qualified in the end because with budget ℓ it is not guaranteed that we can bribe at least s agents. So the main problem for CONST-$f^{(s,t)}$ \$AGENT BRIBERY is how to guarantee that if an agent $a \in A^+$ is bribed, then a will be socially qualified at the end. Let $A^+_{\leq \ell} = \{a \in A^+ \mid y_a \leq \ell\}$ be the set of agents from A^+ that need at most ℓ additional qualifications to become socially qualified and $A^+_{>\ell} = \{a \in A^+ \mid y_a > \ell\}$ the set of agents from A^+ that need more than ℓ additional qualifications to become socially qualified.

We distinguish the case where $n \leq s + \ell + t$ and where $n > s + \ell + t$. In the former case, since s is a constant, we have $n = \mathcal{O}(\ell + t)$ and we can brute force through all subsets of A that cost at most ℓ, bribe them to qualify everyone and check whether the bribery was successful. This leads to an overall running time of $\mathcal{O}((\ell + t)^2 2^{\ell+t})$, as there are at most $2^{\ell+t}$ subsets of A.

In the latter case, a more involved approach is needed. We claim that for every $a \in A^+_{\leq \ell}$, if a qualifies itself, then a will be socially qualified. Notice that for $a \in A^+_{\leq \ell}$, $n = |Q^+(a)| + |Q^-(a)| \leq |Q^+(a)| + y_a + (t - 1) \leq |Q^+(a)| + \ell + t$. Thus, as we are in the case with $n > s + \ell + t$, we can assume that for every $a \in A^+_{\leq \ell}$, $|Q^+(a)| \geq s$, which implies that a will be socially qualified if a qualifies itself. Next, for agents in $A^+_{>\ell}$, we have to bribe all of them because the budget is not enough to make them socially qualified if any one of them is not bribed. However, after doing so, it is not necessarily the case that all agents from $A^+_{>\ell}$ are already socially qualified. To ensure that all agents from $A^+_{>\ell}$ are socially qualified, we can iterate over all subsets of A of size at most s, bribe them and then apply Algorithm 1 with appropriate adjustments. To summarize, this leads to the following algorithm: First, we bribe all agents in $A^+_{>\ell}$ that do not qualify themselves, set A' to be the set of these agents and modify the budget $\ell = \ell - \rho(A')$. If all agents from $A^+_{>\ell}$ are already socially qualified after this bribery, we call Algorithm 1 as CalcB(A, φ, $A^+_{\leq \ell}$, A', ℓ, y_{a_1}, ..., y_{a_n}) with the following minor modifications: Delete lines 1 and 2 (as we already know in advance that the branching factor in line 8 is bounded), and in line 9 return CalcB(A, φ, A^+, $A' \cup \{a\}$, $p - \rho(a)$, y_{a_1}, ..., y_{a_n}). If not, we iterate over all subsets of A with size at most s and with cost at most ℓ, bribe them and check whether all agents in $A^+_{>\ell}$ are socially qualified afterwards. For each subset such that after bribing agents from this subset all agents from $A^+_{>\ell}$ are socially qualified, we do the following: We add the agents from this subset to A' and modify the budget ℓ accordingly, and apply Algorithm 1 as CalcB(A, φ, $A^+_{\leq\ell}$, A', ℓ, y_{a_1}, ..., y_{a_n}) with the same modifications described above.

If the above algorithm returns a solution, this solution is clearly correct. Conversely, assume that $S \subseteq A$ is a subset of agents with cost at most ℓ such that after making all agents from S qualify everyone, all agents from A^+ are socially qualified. First of all, it obviously needs to hold that $A^+_{>\ell} \subseteq S$. Next, let $s' = \min\{|S \setminus A^+_{>\ell}|, s\}$ and let S' be a set of agents such that $A^+_{>\ell} \subseteq S' \subseteq S$ and $|S' \setminus A^+_{>\ell}| = s'$. It is easy to verify that S' is a subset of A with cost at most ℓ (as $S' \subseteq S$) and size at most s, and all agents in $A^+_{>\ell}$ will be socially qualified if we bribe all agents in S'. Therefore, the above algorithm will apply Algorithm 1 after bribing all agents in S' and Algorithm 1 will find S (or another successful bribery containing S').

Iterating over all subsets of A with size at most s and checking whether all agents from $A^+_{>\ell}$ are socially qualified costs $\mathcal{O}(n^{s+2})$. Afterwards, for each such subset, we query Algorithm 1 once leading to an overall running time in $\mathcal{O}(n^{s+4}(\ell + t)^{\ell})$. □

4.1.2 Parameterized complexity with respect to $|A^+|$

Finally, analyzing the influence of the number of agents that should be made socially qualified on the complexity of the problem, it turns out that restricting this parameter is more powerful than restricting ℓ, as the problem is FPT with respect to $|A^+|$ for arbitrary s and t. We solve the problem by constructing an Integer Linear Program (ILP).

Theorem 8 *Const-$f^{(s,t)}$ \$Agent Bribery is FPT with respect to $|A^+|$.*

Proof We first guess the subset $\widetilde{A} \subseteq A^+$ of agents from A^+ which we want to bribe (by iterating over all subsets of A^+). We bribe all agents from $\widetilde{A}$ to qualify everyone (including themselves) and adjust ℓ and φ, accordingly. Note that in agent bribery it is never rational to bribe an agent from A^+ to disqualify itself, as we assume that $s + t \leq n + 2$. Now the problem is how to bribe the agents in $A \setminus A^+$ to make all agents in A^+ socially qualified. We reduce this to an ILP with $|A^+| + 1$ constraints. For every agent $a \in A \setminus A^+$, we introduce a variable $x_a \in \{0, 1\}$ (with $x_a = 1$ if and only if we are going to bribe a). We have two types of conditions for agents in A^+ based on whether they qualify themselves:

$$\Bigg(\sum_{a' \in A \setminus A^+ : \varphi(a',a)=-1} x_{a'} \Bigg) \geq s - |Q^+_{\varphi}(a)| \qquad \forall a \in A^+ : \varphi(a,a) = 1$$

$$\Bigg(\sum_{a' \in A \setminus A^+ : \varphi(a',a)=-1} x_{a'} \Bigg) \geq |Q^-_{\varphi}(a)| - (t-1) \quad \forall a \in A^+ : \varphi(a,a) = -1$$

Finally, we set the objective function

$$\min \sum_{a \in A \setminus A^+} \rho(a) \cdot x_a \,.$$

If the above ILP is feasible and the value of the objective function is at most ℓ, then there exists a successful bribery (given by $x_a = 1$): We bribe all $a \in A \setminus A^+$ with $x_a = 1$ to qualify everyone. In the bribed instance, for every agent $a \in A^+$ with $\varphi(a,a) = 1$, we have that at least s agents qualify a making a socially qualified, and for every agent $a \in A^+$ with $\varphi(a,a) = -1$, at most $t-1$ agents disqualify a making a socially qualified. If the above ILP is not feasible, then there is clearly no successful bribery which bribes exactly the agents $\widetilde{A}$ from A^+. We can iterate over all $2^{|A^+|}$ subsets of A^+, and for each of them use the algorithm of Eisenbrand and Weismantel [15] to solve the corresponding ILP in time $2^{\mathcal{O}(|A^+|^2)}$, since the constraint matrix is indeed binary (and contains both lower and upper-bounds on variables). □

4.2 Agent bribery for other bribery goals

To apply results obtained for CONST-$f^{(s,t)}$ (\$)AGENT BRIBERY to the other considered bribery goals, we use the following result by Erdélyi et al. [17, Lemma 1]:

Lemma 1 ([17]) $f^{(s,t)}(\varphi, A) = A \setminus f^{(t,s)}(-\varphi, A)$, *where* $-\varphi$ *is obtained from* φ *by flipping all values of* φ.

By Lemma 1, Theorems 5 to 8 naturally extend to DEST-$f^{(s,t)}$ AGENT BRIBERY where s and t switch roles:

Corollary 2 *DEST-*$f^{(s,t)}$ *AGENT BRIBERY is* $W[1]$*-hard with respect to* $t + \ell$ *even if* $s = 1$ *and* $W[2]$*-hard with respect to* ℓ *even if* $t = 1$. *DEST-*$f^{(s,t)}$ *\$AGENT BRIBERY is FPT with respect to* $\ell + s$ (*treating t as a constant*) *and FPT with respect to* $|A^-|$.

Turning to constructive+destructive and exact bribery, recall that Erdélyi et al. [17, Theorem 2] proved that CONST-$f^{(s,t)}$ AGENT BRIBERY is NP-complete for all $s \geq 1$ and $t \geq 2$ by a reduction from VERTEX COVER in which they set $A^+ = A$. Applying Lemma 1, this implies that EXACT-/CONST+DEST-$f^{(s,t)}$ AGENT BRIBERY is NP-hard as soon as either $s > 1$ or $t > 1$. Observe that in the only remaining case, i.e., $s = t = 1$, the social qualification of an agent only depends on its opinion about itself. Thus, both EXACT-/CONST+DEST-$f^{(s,t)}$ AGENT BRIBERY are linear-time solvable in this case:

Observation 3 EXACT-/CONST+DEST-$f^{(s,t)}$ AGENT BRIBERY is linear-time solvable for $s = t = 1$. For all other values of s and t, this problem is NP-complete.

Moreover, it is also possible to extend some of our results to constructive+destructive bribery by a slight adaption of Theorem 5 and Theorem 8:

Corollary 3 *CONST+DEST-*$f^{(s,t)}$ *\$AGENT BRIBERY is FPT with respect to* $|A^+| + |A^-|$. *CONST+DEST-*$f^{(s,t)}$ *AGENT BRIBERY parameterized by* $\ell + t$ *is* $W[1]$*-hard even if* $s = 1$ *and also* $W[1]$*-hard with respect to* $\ell + s$ *even if* $t = 1$.

Proof To prove the first part, we can use a similar approach as in the proof of Theorem 8. We first guess the subset $\widetilde{A}^+ \subseteq A^+$ of agents from A^+ and the subset $\widetilde{A}^- \subseteq A^-$ of agents from A^- which we want to bribe (by iterating over all combinations of subsets from A^+ and

A^-). We bribe all agents in $\widetilde{A}^+$ to qualify all agents in A^+ (including themselves) and disqualify all agents in A^-. Similarly, we bribe all agents in $\widetilde{A}^-$ to qualify all agents in A^+ and disqualify all agents in A^- (including themselves). The remaining problem is how to bribe agents in $A \setminus (A^+ \cup A^-)$, which can be reduced to an ILP with $|A^+ \cup A^-| + 1$ constraints analogously to the ILP in the proof of Theorem 8.

The second part holds by using Theorem 5 and Corollary 2. □

Aiming for positive results, parameterizing constructive+destructive bribery by just one of $|A^+|$ and $|A^-|$ is not enough, as the problem is even NP-hard for $s = 2$ and $t = 1$ and $A^+ = \emptyset$ (and thus also for $s = 1$, $t = 2$ and $A^- = \emptyset$), which follows from the reduction from Erdélyi et al. [17] mentioned at the beginning of this section.

Finally, we consider the exact bribery. Here, the two W[1]-hardness results from Corollary 3, which follow from Theorem 5, still hold, as it is possible to precisely specify the desired outcome of the group identification problem in the reduction in Theorem 5. While parameterizing the problem by $|A^+| + |A^-|$ is not meaningful in this context, as discussed above, one of $|A^+|$ and $|A^-|$ combined with s and t is not enough to achieve any positive results:

Corollary 4 *Exact-* $f^{(s,t)}$ *Agent Bribery parameterized by* $\ell + t$ *is* W[1]*-hard even if* $s = 1$ *and also* W[1]*-hard with respect to* $\ell + s$ *even if* $t = 1$.

5 Conclusion

In this paper, we extended the research on bribery in group identification by considering priced bribery, a new model for bribery cost (which was independently introduced by Erdélyi et al. [17]), and two new bribery goals. Moreover, we described how it is possible to use bribery as a method to calculate the margin of victory or distance from winning of agents in a group identification problem. We showed that for all considered rules both quantities can be computed efficiently if the number of agents whose social qualification we are interested in is small. Moreover, we identified some further cases where the general problem is polynomial-time solvable. Namely, for both the liberal-start-respecting rule and the consensus-start-respecting rule, we observed that every bribery involving destructive constraints splits the qualification graph into two parts. As in agent bribery it is easy to make multiple agents socially qualified at the same time, this observation gives rise to a polynomial-time algorithm for the most general variant of priced constructive+destructive bribery for agent bribery. For link bribery, it is possible to use this observation to construct a polynomial-time algorithm for priced destructive bribery. However, the corresponding question for constructive bribery is NP-hard.

For the consent rule, link bribery turns out to be solvable in a straightforward way. In contrast to this, for agent bribery, finding an optimal bribery corresponds to finding a set of agents fulfilling certain covering constraints. This task makes the problem para-NP-hard with respect to $s + t$ [17] and W[1]-hard with respect to ℓ even for $t = 1$ or $s = 1$. Recall that ℓ is probably the most canonical parameter—the budget of the bribery which is either the number or the total costs of the bribery actions. We proved that in the constructive setting, the rule parameter t is slightly more powerful than the rule parameter s in the sense that the problem is still hard parameterized by $s + \ell$ even if $t = 1$, while becoming fixed-parameter tractable parameterized by $t + \ell$ for constant s. Thus, we can see that the complexity implications of the two rule-specific parameters s and t are asymmetric in this case.

Finally, we would like to stimulate further research in this area with some open problems arising from our work. First, we focused on the design of new FPT-algorithms for which we used techniques ranging from rather basic ones (such as branching tree analysis) to the use of ILPs. A natural open question striking from this is whether some combination of parameters for which we construct an FPT-algorithm can be improved to yield a polynomial-sized kernel. Another important direction is to identify new parameters for group identification (mainly for consent rules). A possible example of which could be the (edge/vertex) distance of a symmetrisation of the qualification graph to a cluster graph. Yet another natural example is the Kendall-tau distance to a master (or central) profile, which may arise in the cases when qualification is something objective that can be assessed, for instance, by a test. Moreover, it is also natural to consider the maximum number Δ of agents that some agent qualifies as a parameter. In fact, recently Boehmer et al. [5] showed that Const-$f^{(s,t)}$ Agent Bribery is fixed-parameter tractable with respect to $\Delta + s$ for $t = 1$ and fixed-parameter tractable with respect to the combined parameter $s + t + \ell + \Delta$, which complements our W[1]-hardness result for this problem for the combined parameter $s + t + \ell$ from Theorem 5.

Acknowledgements NB was supported by the DFG project MaMu (NI 369/19). DK is partly supported by the OP VVV MEYS funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics"; part of the work was done while DK was affiliated with TU Berlin and supported by project MaMu (NI 369/19). JL is supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 2 (MOE2019-T2-1-045); part of the work was done while JL was affiliated with TU Berlin and supported by the DFG project AFFA (BR 5207/1 and NI 369/15). This work was started at the research retreat of the TU Berlin Algorithms and Computational Complexity group held in September 2019.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

1. Baumeister, D., Erdélyi, G., & Rothe, J. (2011). How hard is it to bribe the judges? A study of the complexity of bribery in judgment aggregation. In *Proceedings of the 2nd iternational conference on algorithmic decision theory (ADT '11)* (pp. 1–15).
2. Boehmer, N., Bredereck, R., Knop, D., & Luo, J. (2020). Fine-grained view on bribery for group identification. In *Proceedings of the 29th international joint conference on artificial intelligence (IJCAI '20)* (pp. 67–73).
3. Boehmer, N., Bredereck, R., Faliszewski, P., & Niedermeier, R. (2021a). Winner robustness via swap- and shift-bribery: Parameterized counting complexity and experiments. In *Proceedings of the 30th Iiternational joint conference on artificial intelligence (IJCAI '21)* (pp. 52–58).
4. Boehmer, N., Bredereck, R., Heeger, K., & Niedermeier, R. (2021). Bribery and control in stable marriage. *Journal of Artificial Intelligence Research, 71*, 993–1048.
5. Boehmer, N., Bredereck, R., Knop, D., & Luo, J. (2021c). Finding small multi-demand set covers with ubiquitous elements and large sets is fixed-parameter tractable. CoRR arXiv:2104.0124

6. Bredereck, R., Faliszewski, P., Niedermeier, R., & Talmon, N. (2016). Complexity of shift bribery in committee elections. In *Proceedings of the 30th AAAI conference on artificial intelligence (AAAI '16)*, (pp. 2452–2458).
7. Cary, D. (2011). Estimating the margin of victory for instant-runoff voting. In *Proceedings of the 2011 electronic voting technology workshop on trustworthy elections (EVT/WOTE '11).*
8. Chu, Y. J., & Liu, T. H. (1965). On the shortest arborescence of a directed graph. *Scientia Sinica, 14*, 1396–1400.
9. Cygan, M., Fomin, F. V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., & Saurabh, S. (2015). *Parameterized Algorithms*. Springer.
10. Dimitrov, D. (2011). The social choice approach to group identification. In *Consensual processes* (pp. 123–134).
11. Dimitrov, D., Sung, S. C., & Xu, Y. (2007). Procedural group identification. *Mathematical Social Sciences, 54*(2), 137–146.
12. Downey, R. G., & Fellows, M. R. (2013). *Fundamentals of Parameterized Complexity*. Springer.
13. Duddy, C., Piggins, A., & Zwicker, W. S. (2016). Aggregation of binary evaluations: a Borda-like approach. *Social Choice and Welfare, 46*(2), 301–333.
14. Edmonds, J. (1967). Optimum branchings. *Journal of Research of the National Bureau of Standards B, 71*(4), 233–240.
15. Eisenbrand, F., & Weismantel, R. (2018). Proximity results and faster algorithms for integer programming using the Steinitz lemma. In *Proceedings of the 29th annual ACM-SIAM symposium on discrete algorithms (SODA '18)* (pp. 808–816).
16. Erdélyi, G., & Yang, Y. (2020). Microbribery in group identification. In *Proceedings of the 19th international conference on autonomous agents and multiagent systems (AAMAS '20)* (pp. 1840–1842).
17. Erdélyi, G., Reger, C., & Yang, Y. (2020). The complexity of bribery and control in group identification. *Autonomous Agents and Multi-Agent Systems, 34*(1), 8.
18. Faliszewski, P., & Rothe, J. (2016). Control and bribery in voting. In *Handbook of computational social choice* (pp. 146–168).
19. Faliszewski, P., Hemaspaandra, E., & Hemaspaandra, L. A. (2009). How hard is bribery in elections? *Journal of Artificial Intelligence Research, 35*, 485–532.
20. Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L. A., & Rothe, J. (2009). Llull and Copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research, 35*, 275–341.
21. Faliszewski, P., Skowron, P., Slinko, A., & Talmon, N. (2017a). Multiwinner voting: A new challenge for social choice theory. In *Trends in computational social choice* (pp. 27–47).
22. Faliszewski, P., Skowron, P., & Talmon, N. (2017b). Bribery as a measure of candidate success: Complexity results for approval-based multiwinner rules. In *Proceedings of the 16th conference on autonomous agents and multiagent systems (AAMAS '17)* (pp. 6–14).
23. Faliszewski, P., Slinko, A., & Talmon, N. (2020). Multiwinner rules with variable number of winners. In *Proceedings of the 24th European conference on artificial intelligence (ECAI '20)* (pp. 67–74).
24. Flum, J., & Grohe, M. (2006). *Parameterized Complexity Theory*. Springer.
25. Gabow, H. N., Galil, Z., Spencer, T. H., & Tarjan, R. E. (1986). Efficient algorithms for finding minimum spanning trees in undirected and directed graphs. *Combinatorica, 6*(2), 109–122.
26. Houy, N. (2007). "i want to be a J!'': Liberalism in group identification problems. *Mathematical Social Sciences, 54*(1), 59–70.
27. Kasher, A. (1993). Jewish collective identity. In *Jewish Identity* (pp. 56–78).
28. Kasher, A., & Rubinstein, A. (1997). On the question "who is a j?'' a social choice approach. *Logique et Analyse, 40*(160), 385–395.
29. Kilgour, D. M. (2016). Approval elections with a variable number of winners. *Theory and Decision, 81*(2), 199–211.
30. Lackner, M., & Maly, J. (2021). Approval-based shortlisting. In *Proceedings of the 20th international conference on autonomous agents and multiagent systems (AAMAS '21)* (pp. 737–745).
31. Magrino, T.R., Rivest, R.L., & Shen, E. (2011). Computing the margin of victory in IRV elections. In *Proceedings of the 2011 electronic voting technology workshop / workshop on trustworthy elections (EVT/WOTE '11).*
32. Matoušek, J., & Nešetřil, J. (2009). *Invitation to Discrete Mathematics* (2nd ed.). Oxford University Press.
33. Miller, A. D. (2008). Group identification. *Games and Economic Behavior, 63*(1), 188–202.
34. Nederlof J (2009) Fast polynomial-space algorithms using Möbius inversion: Improving on Steiner tree and related problems. In *Proceedings of the 36th international colloquium on automata, languages and programming (ICALP '09)* (pp. 713–725).
35. Niedermeier, R. (2006). *Invitation to Fixed-Parameter Algorithms*. Oxford University Press.

36. Orlin, J.B. (2013). Max flows in O(nm) time, or better. In *Proceedings of the 45th annual ACM symposium on theory of computing (STOC '15)* (pp. 765–774).
37. Samet, D., & Schmeidler, D. (2003). Between liberalism and democracy. *Journal of Economic Theory, 110*(2), 213–233.
38. Xia, L. (2012). Computing the margin of victory for various voting rules. In *Proceedings of the 13th ACM conference on electronic commerce (EC '12)* (pp. 982–999).
39. Yang, Y., & Dimitrov, D. (2018). How hard is it to control a group? *Autonomous Agents and Multi-Agent Systems, 32*(5), 672–692.