The Complexity of Bribery in Network-Based Rating Systems

Umberto Grandi
University of Toulouse
France
umberto.grandi@irit.fr

James Stewart
Imperial College London
United Kingdom
james.stewart13@imperial.ac.uk

Paolo Turrini
University of Warwick
United Kingdom
p.turrini@warwick.ac.uk

Abstract

We study the complexity of bribery in a network-based rating system, where individuals are connected in a social network and an attacker, typically a service provider, can influence their rating and increase the overall profit. We derive a number of algorithmic properties of this framework, in particular we show that establishing the existence of an optimal manipulation strategy for the attacker is **NP**-complete, even with full knowledge of the underlying network structure.

1 Introduction

The widespread use of online rating systems has given rise to the problem of how we might guarantee their reliability. The emergence of recommender systems (Bobadilla et al. 2013), as platforms that construct (often learning-based) protocols to match users and provide accurate suggestions, is one such effort to address this issue.

Meanwhile, research in artificial intelligence, in particular the fields of mechanism design, algorithmic game theory and computational social choice, has been paying increased attention to the strategic actions of decision-makers, studying notions such as manipulation and truthfulness for collective decision making, and coming up with formal requirements for those properties to be realised. Surprisingly though, as also noted in Tennenholtz (2008) and Alon et al. (2015), there is a lack of formal study of what guarantees are needed for recommendation systems to be reliable, an observation which can be extended to rating systems in general. Users' evaluations can be carefully screened, and obvious biases (e.g., ethinicity-based discrimination) can be detected, but no theoretical guarantee is provided on whether a rating-system effectively discourages manipulation.

In a recent paper, Grandi and Turrini (2016) proposed a network-based rating system and assessed the effect of *bribery*, a well known and studied form of manipulation (for a recent survey see, e.g., Faliszewski and Rothe, 2016), in comparison with classical rating systems such as the one used by the popular Tripadvisor® website. In their model, customers' decision-making is formed by aggregating the opinions of their peers — their *personalised* rating — and an external service provider can give incentives to customers to modify their rating, potentially increasing the overall expected revenue. They demonstrate the fundamental effect of uncertainty in preventing manipulation, in particular showing that the personalised rating system is (optimally) manipulable if the attacker has full knowledge of the underlying network and even, as is common in real-world rating systems, when a number of users do not express any opinion.

However, nothing is said about the computational difficulty of carrying out such an operation. Even when a system is manipulable in theory, this might for instance require the solution of a complex combinatorial problem. Understanding the practical barriers to manipulation is therefore an important challenge and it can lead to a major validation of a system only studied in theory.

In this paper we follow the standard approach of computational voting theory, which has shown that even if strategy-proof voting rule cannot be designed, as a consequence of the Gibbard-Satterthwaite Theorem (Gibbard 1973; Satterthwaite 1975), some voting rules exists that are safe against manipulation *in practice*, as computing manipulation strategies is **NP**-hard (see, e.g., Conitzer and Walsh, 2016).

Contribution. We show that even if a personalised rating system is manipulable in theory, the problem of manipulating it is intractable in practice. In particular, we establish that even when the attacker has full knowledge of the network the problem of determining the existence of a manipulation strategy guaranteeing at least a given reward — and, notably, an optimal one — is **NP**-complete. We do so by giving a polynomial-time reduction from the problem of finding an independent set of a given size k in a 3-regular graph.

Related Literature Although our focus is personalised rating, first introduced in Grandi and Turrini (2016), there are a number of relevant approaches that have close connections to our work. First and foremost, Conitzer et al. (2010), Bu (2013), Todo and Conitzer (2013) and Brill et al. (2016), who study the effect of adding fake profiles to a social network, a closely related problem to that of bribery. As already pointed at previously, an extremely relevant line of research is the work of Alon et al. (2015) and Lev and Tennenholtz (2017), who looks at theoretical guarantees for group recommendations, as well as papers that have looked at social network-based recommendations, such as (Andersen et al. 2008). See also the recent survey by Grandi (2017) for relevant literature on the interplay between mechanisms for collective choice and social networks. Finally, trust and reputation have been central topics in the multi-agent systems community (Conte and Paolucci 2002; Sabater and Sierra 2005; Garcin, Faltings, and Jurca 2009), and here we study the computational aspects of their manipulation.

Paper organisation. In Section 2 we introduce the basic definitions of personalised ratings and bribing strategies. In Section 3 we show basic results on bribing strategies, and in Section 4 we show our main result, which establishes the **NP**-completeness of computing an optimal bribing strategy and of deciding the possibility for successful manipulation. Section 5 concludes the paper. Due to space constraints, some of the proofs have been omitted.

2 Personalised Rating

In this section, we collect all the preliminary notions and definitions from Grandi and Turrini (2016) needed to establish our results.

Evaluations and Rating

The framework we will be working on consists of an abstract object r, called *restaurant*, and a finite set of individuals $C = \{c_1, ..., c_n\}$, called *customers*. Customers are connected in an undirected graph $E \subseteq C \times C$, intuitively their social network. For each $c \in C$, the *neighbourhood* of c is defined to be $N(c) = \{x \in C : (c, x) \in E\}$, with the requirement that $c \in N(c)$, $\forall c \in C$, i.e., every customer is connected to themself.

Customers concurrently submit an *evaluation* of the restaurant, which is modelled as an element from the set $Val \cup \{*\}$; where $Val \subseteq [0, 1]$ and the distinguished element $*$ represents the evaluation of a customer with no opinion. Note that a property of the chosen set Val is that it is closed under the operation $min\{1, x + y\}$ for all $x, y \in Val$ (where $min\{1, *\} = *$ and $x + * = * + x$ for all $x \in Val$). Most known rating methods, for example a discrete rating scale of one to five stars, can be mapped onto the interval $[0, 1]$ and analysed within this framework. The evaluations provided by customers thus take the form of an evaluation function $eval : C \to Val \cup \{*\}$.

Some or possibly all of the customers can express an opinion on the restaurant, and those who do are called *voters*, forming the set $V \subseteq C$ where $V = \{c \in C : eval(c) \neq *\}$. The set of voters is always assumed to be non-empty.

In contrast with what is typically proposed in this setting, i.e., defining the rating of the restaurant as the average rating expressed by *all* the customers C, Grandi and Turrini (2016) proposed a personalised version in which each customer would be shown the average rating expressed by her own neighbours. Formally, fixing a network E, and given $eval$ and $c \in C$, the expression $\mathbb{P}\text{-}\mathtt{rating}(c, eval)$, i.e., the personalised rating of customer c under evaluation $eval$, is defined as follows:

$$\mathbb{P}\text{-}\mathtt{rating}(eval, c) = \underset{k \in N(c) \cap V}{avg} (eval(k)),$$

with the additional assumption that every customer of the network is connected to at least one voter.

Utility and Bribing Strategies

Intuitively, individuals' personalised ratings indicate their propensity to use the service, in our case their propensity to go to the restaurant. It is assumed, therefore, that the actual utility a restaurant receives is proportional to the rating that it is given by the customers, formally $u^0_{\mathbb{P}} = \sum_{c \in C} \mathbb{P}\text{-}\mathtt{rating}(eval, c)$. This is a simplified setup, which can be generalised by assuming a linear correlation between the observed rating and the probability to use the service, without affecting the conclusions of this paper.

At the initial stage of the game, the restaurant owner receives $u^0_{\mathbb{P}}$ and can decide to invest part of it to influence a subset of customers and improve upon the initial situation. Such an investment is referred to as a *bribing strategy*. Formally, it is a function $\sigma : C \to Val$ such that $\sum_{c \in C} \sigma(c) \leq u^0_{\mathbb{P}}$. The latter constraint imposes that the strategy is budget-balanced, i.e., the service provider cannot reinvest more than the profit guaranteed by $u^0_{\mathbb{P}}$. The set of all strategies is referred to as Σ and σ^0 is defined to be the strategy that assigns 0 to all customers. A *bribing* strategy is any strategy different from σ^0.

The evaluation $eval^\sigma(c)$, i.e., the customers' evaluation after the execution of a strategy σ, is defined as $eval^\sigma(c) = min\{1, eval(c) + \sigma(c)\}$, where $* + \sigma(c) = \sigma(c)$, if $\sigma(c) \neq 0$, and $* + \sigma(c) = *$, if $\sigma(c) = 0$. We can again relax this assumption by assuming that the effect of a bribe is linearly correlated to the new evaluation given by a customer, without affecting our results. A strategy is said to be *efficient* if $\sigma(c) + eval(c) \leq 1$ for all $c \in C$.

The change in utility due to the execution of a strategy σ is defined as

$$u^\sigma_{\mathbb{P}} = \sum_{c \in C} \mathbb{P}\text{-}\mathtt{rating}(eval^\sigma, c) - \sum_{c \in C} \sigma(c),$$

where $\mathbb{P}\text{-}\mathtt{rating}(eval^\sigma, c)$ is the $\mathbb{P}\text{-}\mathtt{rating}$ of customer c calculated with the new evaluation $eval^\sigma$.

Strategies can be more or less rewarding for the attacker. Let σ be a strategy. The *revenue* of σ is defined as $r_{\mathbb{P}}(\sigma) = u^\sigma_{\mathbb{P}} - u^0_{\mathbb{P}}$. σ is *profitable* if $r_{\mathbb{P}}(\sigma) > 0$. Moreover, we say that a strategy σ is *optimal* if $u^\sigma_{\mathbb{P}} \geq u^{\sigma'}_{\mathbb{P}}$ for all $\sigma' \in \Sigma$. Finally, a network-based rating system is said to be *bribery-proof* if σ^0 is optimal.

Example 1. *Consider three customers* c_1, c_2, c_3, *with* $eval(c_1) = eval(c_2) = 0.5$ *and* $eval(c_3) = *$, *and such that* $E = \{(c_1, c_2), (c_2, c_3)\}$. *We have that* $\mathbb{P}\text{-}\mathtt{rating}(eval, c_1) = \mathbb{P}\text{-}\mathtt{rating}(eval, c_2) = \mathbb{P}\text{-}\mathtt{rating}(eval, c_3) = 0.5$, $u^0_{\mathbb{P}} = 1.5$. *Let* σ *be such that* $\sigma(c_1)$=0.5 *and* $\sigma(c_2)$=$\sigma(c_3)$=0. *Such strategy is budget balanced, but is such that* $\mathbb{P}\text{-}\mathtt{rating}(eval^\sigma, c_1) = \mathbb{P}\text{-}\mathtt{rating}(eval^\sigma, c_2) = 0.75$ *and* $\mathbb{P}\text{-}\mathtt{rating}(eval^\sigma, c_3) = 0.5$, *therefore* $u^\sigma_{\mathbb{P}} = 2$, *so, subtracting the expenses,* $r_{\mathbb{P}}(\sigma) = 0$. *This is true for every strategy that bribes only* A *by less than* 0.5, *while the revenue is negative if the bribe is strictly higher, as it would end up wasting utility. Consider now* σ' *such that* $\sigma'(c_3)$=0.5 *and* $\sigma'(c_2)$=$\sigma'(c_1)$=0. *In this case* $\mathbb{P}\text{-}\mathtt{rating}(eval^{\sigma'}, c_1) = \mathbb{P}\text{-}\mathtt{rating}(eval^{\sigma'}, c_2) = \mathbb{P}\text{-}\mathtt{rating}(eval^{\sigma'}, c_3)$=0.5 *and therefore* $u^{\sigma'}$=1.5. *However* $r_{\mathbb{P}}(\sigma) = -\ 0.5$, *as we spent* 0.5 *to influence* C.

However, each strategy σ^* bribing only c_2 up to 0.5 yields a strictly positive revenue. In particular $\sigma^*(c_2)$=0.5 and $\sigma^*(c_3)$=$\sigma^*(c_1)$=0 is the only optimal strategy, with $\mathbb{P}\text{-}\mathtt{rating}(eval^{\sigma^*}, c_1) = 0.75 = \mathbb{P}\text{-}\mathtt{rating}(eval^{\sigma^*}, c_2)$ *and* $\mathbb{P}\text{-}\mathtt{rating}(eval^{\sigma^*}, c_3) = 1$*, which means* $u^{\sigma^*} = 2.5$ *and* $r_{\mathbb{P}}(\sigma^*) = 0.5$.

Bribery-Proofness

As shown by Grandi and Turrini (2016), when the positions of the individuals on the network are known, and in the absence of non-voters, $\mathbb{P}\text{-}\mathtt{rating}$ is not bribery-proof, and an algorithm can be devised to compute an optimal bribing strategy. This is however only shown for the rather unrealistic case in which all customers give an opinion of the restaurant, by providing an example of a profitable bribing strategy. Within this paper, we study the more general and realistic case where a subset of customers might choose not to provide any evaluation.

3 Bribes under the $\mathbb{P}\text{-}\mathtt{rating}$

In this section, and in the rest of the paper, we consider the case in which the service provider has complete knowledge of the customers' network but where some of the customers do not vote. The service provider is allowed to bribe a subset of all of the customers. We denote this case *non-voters and known locations (NVKL)*. More formally, the service provider receives a network (C, E) and an evaluation *eval* as input, which also determines the subset $V \subset C$ of customers who have voted.

The effect of bribing a voter is intuitively simpler, as $E \cap V \times V$ is not affected by evaluation updates. When bribing a non-voter though, the set of voters itself might change. So one might think that the order in which customers' are bribed plays a significant role in the rating's manipulation. However, we show next that this is not the case, as sequences of bribes can be decomposed into atomic ones, independent of their order. Proofs are omitted in the interest of space.

Single Bribes

We begin by considering the revenue gained by (efficiently) bribing a solo voter $x \in V$:[1]

Proposition 1. *Let $V \subseteq C$, let $x \in V$, and let σ be an efficient bribing strategy such that $\sigma(x) = b > 0$ and $\sigma(y) = 0$, for all $y \in C \setminus \{x\}$. For each $y \in N(x)$, set $\nu_y = |N(x) \cap V|$. The revenue gained is*

$$b\left[\left(\sum_{y \in N(x)} \frac{1}{\nu_y}\right) - 1\right].$$

We then move to computing the revenue gained by (efficiently) bribing a solo non-voter $x \in C \setminus V$:

Proposition 2. *Let $V \subseteq C$ give rise to an evaluation eval, let $x \in C \setminus V$, and let σ be an efficient bribing strategy such that $\sigma(x) = b > 0$ and $\sigma(y) = 0$, for all $y \in C \setminus \{x\}$. For each $y \in N(x)$, set $\nu_y = |N(y) \cap V|$. The revenue gained is*

$$b \sum_{y \in N(x)} \left(\frac{1}{\nu_y + 1} - \frac{1}{|N(x)|}\right) - \sum_{y \in N(x)} \left(\frac{1}{\nu_y(\nu_y + 1)} \sum_{k \in N(y) \cap V} eval(k)\right).$$

Proposition 1 shows us that for some fixed amount, the extent to which a voter is profitable to bribe can be expressed as *a function of only the network structure* (by this we refer to the topology of the network and the positions of non-voters on the network). Contrary to this, we see by Proposition 2 that in order to express the extent to which a non-voter is profitable to bribe, we require the evaluation of the network as well as its structure.

Independence of Bribing Order

We now explore how the order of bribing customers impacts the resulting revenue.

Proposition 3. *Let $V \subseteq C$ give rise to an evaluation eval, let $x, x' \in C \setminus V$ be distinct non-voters, and let σ be an efficient strategy such that $\sigma(x) = b > 0$, $\sigma(x') = b' > 0$, and $\sigma(y) = 0$, for all $y \in C \setminus \{x, x'\}$. No matter whether we bribe x before x' or x' before x, the resulting cumulative revenue will be the same.*

Now we consider the case where we compare bribing a non-voter x and then a voter x' with bribing the voter x' and then the non-voter x.

Proposition 4. *Let $V \subseteq C$ give rise to an evaluation eval, let $x \in C \setminus V$, let $x' \in V$, and let σ be an efficient strategy such that $\sigma(x) = b > 0$, $\sigma(x') = b' > 0$, and $\sigma(y) = 0$, for all $y \in C \setminus \{x, x'\}$. No matter whether we bribe x before x' or x' before x, the resulting revenue will be the same.*

Propositions 3 and 4 show that, despite the fact that bribing non-voters transforms the set of voters, we can ignore the order of bribes when evaluating the effect of a strategy.

A (Non-Optimal) Greedy Algorithm

As strategies involving multiple customers can be decomposed by looking at strategies bribing a single one, which can then be executed without attention to the order, a greedy algorithm could be proposed to construct an optimal strategy as follows. First, we select the customer who will yield the highest revenue when bribed the maximal amount allowed by the initial budget and their own evaluation. Note that this could be either a voter or a non-voter. Then, repeat the process until the initial budget is exhausted, or until all individuals on the network who do not have maximal evaluation yield a negative revenue when bribed. This simple idea, which was shown to work when everyone votes (Grandi and Turrini 2016), does not yield an optimal strategy, as the following example shows.

Example 2. *Consider a 6-clique $\mathcal{X}$ of non-voters, each connected to an associated voter with evaluation $\frac{1}{2}$ as is depicted in Figure 1. The initial utility of the network is as follows:*

$$u^0_{\mathbb{P}} = \sum_{c \in C} \mathbb{P}\text{-}rating(c, eval) = \sum_{c \in \mathcal{X}} \frac{1}{2} + \sum_{c \in C \setminus \mathcal{X}} \frac{1}{2} = 6.$$

[1] This result is a reformulation of Proposition 12 from previous work by (Grandi and Turrini 2016).

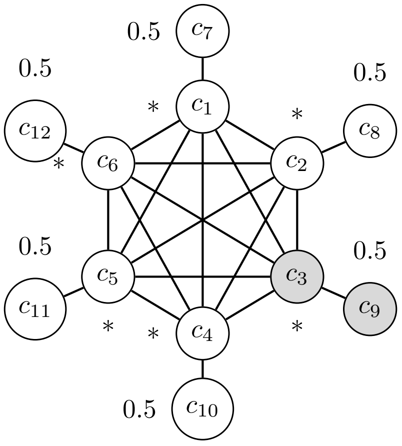

Figure 1: A network for which a greedy algorithm does not yield an optimal bribing strategy. The numbers or $*$, above or left of each node, indicate the initial evaluation of the corresponding customer.

Suppose that we bribe some clique customer $x \in \mathcal{X}$ its maximal amount. By Proposition 2, the revenue gain is:

$$= \sum_{y\in N(x)} \left(\frac{1}{2} - \frac{1}{7}\right) - \sum_{y\in N(x)} \left(\frac{1}{2} \sum_{k\in N(y)\cap V} \frac{1}{2}\right) = \frac{3}{4}.$$

Alternatively, suppose we bribe some non-clique customer $x \in C \setminus \mathcal{X}$ its maximal amount $\frac{1}{2}$. The revenue gain from doing so, by Proposition 1, is

$$\frac{1}{2}\left[\left(\sum_{y\in N(x)} \frac{1}{\nu_y}\right) - 1\right] = \frac{1}{2}\left[\left(\sum_{y\in N(x)} 1\right) - 1\right] = \frac{1}{2}.$$

Since $\frac{3}{4} > \frac{1}{2}$, a greedy algorithm would first bribe a clique-customer the amount 1.

Consider now any currently unbribed non-voter and the pair it makes with the unique voter it is adjacent to, as is depicted in Figure 1 by the vertices coloured in gray. As long as the non-voter remains unbribed, we can bribe the voter by $\frac{1}{2}$ and gain an increase in revenue. Consequently, any greedy algorithm must bribe at least one customer of the pair. This is true for all five remaining such pairs. Therefore, the least amount that a greedy algorithm bribes, from here on, is $\frac{1}{2}$ per pair. The revenue produced by the bribing strategy computed by a greedy algorithm is therefore:

$$r_{\mathbb{P}}(\sigma) \leq 12 - 6 - 1 - \frac{5}{2} = \frac{5}{2}.$$

This is due to the fact that the maximum utility of the network after executing any strategy σ is $12 = |C|$, the initial utility of the network is 6, the amount 1 is spent on the first bribe, and at least $\frac{1}{2}$ is spent on bribing the remaining five voter/non-voter pairs (we established that at least one customer of each of these pairs must be bribed).

Consider the strategy σ' of bribing all non-clique customers fully. That is,

$$\sigma'(6){=}\sigma'(7){=}\sigma'(8){=}\sigma'(9){=}\sigma'(10){=}\sigma'(11){=}\sigma'(12){=}\frac{1}{2}$$

and $\sigma'(x) = 0$ for all other customers x. The revenue gained by playing this strategy is

$$r_{\mathbb{P}}(\sigma') = \sum_{c\in C} \mathbb{P}\text{-}rating(c, eval) - u^0_{\mathbb{P}} - 3 = 12 - 6 - 3 = 3$$

We therefore conclude that the greedy algorithm does not compute an optimal bribing strategy.

While previous work has used a greedy approach to show whether a system is manipulable by finding the optimal strategy, this does not work for the general case. The question now is whether we can find an optimal bribing strategy in polynomial-time, or at least compute whether there exists a successful manipulation strategy in polynomial-time, or, instead, whether there might be a complexity-theoretic barrier to doing so.

4 The Complexity of Bribery under the $\mathbb{P}$-rating

We now investigate, from a complexity theoretic standpoint, the problem of computing a bribing strategy yielding at least some given revenue, under our assumptions – when not every customer votes and the restaurant has full knowledge of each customer's position. This, notice, will allow us to determine the existence of both a successful manipulation strategy, and an optimal strategy. Firstly we re-formulate the above optimisation problem as a decision problem.

BRIBE-NVKL
Instance: Network (C, E), evaluation $eval_0$, $\rho \in \mathbb{Q}$
Yes-Instance: An instance of BRIBE-NVKL s.t. there exists a strategy σ with $\mathbf{r}(\sigma) \geq \rho$

Any instance of the above problem should adhere to the usual restrictions of the framework. These are, most importantly, that the initial evaluation is such that every customer $c \in C$ is adjacent to at least one customer $c' \in C$ such that $eval(c') \neq *$ (recall that every customer is adjacent to itself). Also, any strategy σ is such that $\sum_{c\in C} \sigma(c)$ is at most the initial utility resulting from $eval_0$.

The following proposition is straightforward:

Proposition 5. BRIBE-NVKL *is in* **NP**.

Proof. Given a customer network (C, E), an evaluation $eval$, and $\rho \in \mathbb{Q}$, we can clearly decide whether a given strategy σ yields a revenue of at least ρ in polynomial-time (we simply evaluate the strategy). It therefore follows that BRIBE-NVKL is in **NP**. □

NP-hardness

In what follows, we show that BRIBE-NVKL is **NP**-hard, by giving a reduction from the known **NP**-complete problem of finding an independent set on 3-regular graphs, aka ISREG(3) (Garey and Johnson 1990).

Recall that a graph G is 3-regular if the degree of every vertex is 3, and an independent set of G is a subset X of its vertices such that there is no edge of G joining any pair of vertices in X. We can now give the following definition:

ISREG(3)

Instance:	A 3-regular graph G, $k \in \mathbb{N}$
Yes-Instance:	An instance of ISREG(3) such that G has an independent set of size at least k

We can prove the following:

Proposition 6. BRIBE-NVKL *is* **NP**-*hard.*

Proof. We start by giving a reduction from an arbitrary instance of ISREG(3) to an instance of BRIBE-NVKL. That is, given a 3-regular graph G and $k \in \mathbb{N}$, we construct a network (C, E), an initial evaluation $eval_0$, and $\rho \in \mathbb{Q}$ such that G has an independent set of size at least $k \iff$ there exists a strategy on $((C, E), eval_0)$ that yields a revenue of at least ρ. Given a 3-regular graph G, we define a network of customers as follows:

Customers The set C of customers is composed of *old*, *pendant*, and *edge* cutomers. For all vertices $v \in G$, we create an *old customer* $v \in C$, as well as a set of *pendant customers* $v_1, .., v_n \in C$, where n is the number of vertices of G. For each edge (u, v) of G, we introduce an *edge customer* $w_{u,v} \in C$.

Network The network E relating customers is defined as follows. For each old customer v, there is an edge $(v, v_i) \in E$ for $i = 1, 2, ..., n$, connecting it to the related pendant customers. For every edge (u, v) of G, we add $(u, w_{u,v})$ and $(w_{u,v}, v)$ to E, relating the two old customers with the corresponding edge customer.

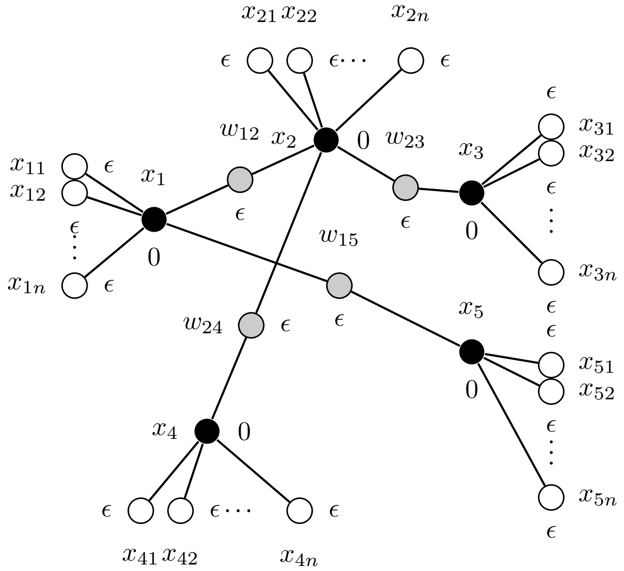

Figure 2: The figure above shows a portion of a 3-regular graphs, formed by the five black vertices (additional edges required by 3-regularity have been omitted). Black vertices are therefore old customers, white vertices are pendant customers, and grey vertices are edge customers. The associated bribing strategy is marked with 0, 1 and ϵ labels.

For any such network as constructed above, we can define an initial evaluation $eval_0$ as follows, where $0 < \epsilon < 1$ is some value that will be set later in the proof:

- If $c \in C$ is an old customer then $eval_0(c){=}*$ (non-voter).
- If $c \in C$ is an edge or pendant customer then $eval_0(c){=}\epsilon$.

An example of the construction of the customer network and evaluation from a graph G can be seen in Figure 2.

By the construction of the network, we have that for all $c \in C$, the $\mathbb{P}$-rating$(c, eval_0) = \epsilon$ (recall that we assumed $c \in N(c)$ for all customers). Every customer of the newly constructed customer network contributes ϵ to the initial utility of the network and therefore $u^0_{\mathbb{P}} = \epsilon(n + n^2 + \frac{3n}{2})$. We now choose ϵ so that $u^0_{\mathbb{P}} = k$; that is, so that

$$\epsilon = \frac{k}{n + n^2 + \frac{3n}{2}}.$$

By assumption the restaurant owner can only make bribes totalling at most k. Furthermore, note that the initial evaluation is a valid one in that every customer of the network is adjacent to at least one voter. Finally, let

$$\rho = k(1 - \epsilon)\left(\frac{1}{n + 4} + \frac{n + 3}{2}\right) - k.$$

($\Longrightarrow$) Suppose that our instance (G, k) of ISREG(3) is a yes-instance; that is, there is a set I of k vertices such that no two vertices of I are adjacent in G. Consider the bribing strategy for (C, E) (as constructed above) where $\sigma(c) = 1$, for every old customer corresponding to some vertex of I, and $\sigma(c') = 0$ for all other $c' \in C$.

Let us now compute the revenue obtained by σ. Recall that the revenue is equal to the increase in $\mathbb{P}$-$\mathtt{rating}$ of the bribed customers and their neighbourhoods (old, pendant, and edge customers), minus the cost of the bribe. The cumulative increase in rating of bribed old customers is:

$$k\left(\frac{1 + (n + 3)\epsilon}{n + 4} - \epsilon\right) = k\frac{1 - \epsilon}{n + 4}.$$

The cumulative increase in rating of pendent customers is:

$$nk\left(\frac{1 + \epsilon}{2} - \epsilon\right) = nk\frac{1 - \epsilon}{2}.$$

Finally, the increase in rating due to edge customers is:

$$3k\left(\frac{1 + \epsilon}{2} - \epsilon\right) = 3k\frac{1 - \epsilon}{2}.$$

Recall that bribed old customers correspond to an independent set in G. Summing up, the revenue of strategy σ is:

$$k(1 - \epsilon)\left(\frac{1}{n + 4} + \frac{n + 3}{2}\right) - k = \rho.$$

Therefore $((C, E), eval_0, \rho)$ is a yes-instance of NVKL.

($\Longleftarrow$) We now suppose that $((C, E), eval_0, \rho)$ is a yes-instance of NVKL and that σ is a bribing strategy that yields a revenue of at least ρ. We will assume that σ is also optimal, i.e., that there is no other strategy σ' yielding a higher revenue. We will now show that σ can be transformed into a revenue-equivalent strategy such that (a) only old customers are bribed, (b) all bribed old customers are bribed fully, and (c) exactly k old customers are bribed.

We begin by showing the following technical lemma, whose proof is omitted in the interest of space:

Lemma 7. *Let σ be an optimal bribing strategy. Let X be the set of customers for which $eval(x) < eval^{\sigma}(x) < 1$. Let $v_y = |N(y) \cap V|$ for any $y \in C$. For $x, y \in X$*

$$\sum_{z \in N(x)} \frac{1}{v_z} = \sum_{z \in N(y)} \frac{1}{v_z}.$$

So, given an optimal bribing strategy, we can move bribes amongst non-fully bribed voters arbitrarily without affecting the revenue acquired so long as we do not totally remove all the bribe from a customer that was not originally a non-voter, and we do not turn a non-voter into a voting one.

Revenue equivalent strategy - new customers. Let us call *new customers*, the set of edge and pendant customers. We begin by showing that σ can be modified into an optimal strategy that does not bribe any new customer.

If a new customer is bribed then the bribes to new customers can be enumerated in descending order as $1-\epsilon, 1-\epsilon, ..., 1-\epsilon, \epsilon_1, \epsilon_2, ..., \epsilon_s$, for some $s \geq 0$ and where $0 < \epsilon_i < 1-\epsilon$ for each $i = 1, 2, ..., s$, with possibly no bribe of $1-\epsilon$. By Lemma 7, we can move bribes amongst the new customers so that we may assume that all but at most one new customer is not fully bribed; that is, that $s \leq 1$. The following result is needed:

Lemma 8. *There exists an old customer c' who has not been bribed and where at most one of its adjacent new pendant customers has been bribed.*

First, we suppose that *there exists a fully bribed new customer*, and derive a contradiction with the optimality of σ.

Consider the bribe of $1-\epsilon$ to c, and consider the increase in $\mathbb{P}$-rating generated by this single bribe. If c is a new pendant customer then this contribution is certainly less than 2 as $|N(c)| = 2$, and if c is a new edge customer then this contribution is less than 3 as $|N(c)| = 3$. Therefore, in all cases, the bribe of $1-\epsilon$ to c contributes less than 3 units to the overall utility accrued from σ.

By Lemma 8, let c' be an old customer that is not bribed and that is adjacent to at most one new pendant customer that has been bribed. Consider moving the $1-\epsilon$ bribe from c to c'; so, we obtain a new (efficient) strategy σ'. Let us examine the increase in $\mathbb{P}$-rating generated by this new $1-\epsilon$ bribe.

At least $n-1$ of the new pendant customers adjacent to c' have not been bribed and so the associated cumulative increase in rating is given by $(n-1)\frac{1}{2} - (n-1)\epsilon$ and given that $\epsilon \leq \frac{2}{2n+5}$ then the cumulative increase in utility is

$$(n-1)\left(\frac{1}{2} - \epsilon\right) > \frac{n-1}{2} - 1.$$

Bribing c' might reduce the $\mathbb{P}$-ratings of c' and its adjacent new edge customers. However, this reduction is certainly less than 4 units. Therefore we may conclude that the movement of $1-\epsilon$ of bribe from c to c' increases the overall utility by an amount greater than $\left(\frac{n-1}{2} - 1\right) - 7$ units. This amount is strictly positive for n sufficiently large ($n \geq 14$). Therefore the strategy σ' that we have constructed yields a revenue greater than that of σ, in contradiction with its optimality.

Suppose now that *some new customer c has been bribed some amount δ such that* $0 < \delta < 1-\epsilon$. By a detailed case study – omitted for space constraints – we can again derive a contradiction with the optimality of σ. Therefore, we conclude that *no new customer have been bribed* in the revenue-equivalent optimal strategy σ.

Revenue equivalent strategy - old customers. We now turn our attention to old customers. The bribes on old customers can be enumerated in descending order as $1, 1, ..., 1, \delta_1, \delta_2, ..., \delta_m$, for some $m \geq 0$ and where $0 < \delta_i < 1$, for each $i = 1, 2, ..., m$, with possibly no bribes of 1. Without loss of generality, we may assume that $\sum_{i=1}^{m} \delta_i \leq 1$; otherwise, we would have that $m \geq 2$ and we could reduce the bribes $\delta_2, \delta_3, ..., \delta_m$, without making any equal to zero, so as to increase the bribe δ_1 to 1 and secure another fully bribed customer. The following result is needed:

Lemma 9. *Let (C, E) be some network with initial evaluation $eval_0$ and let σ be a bribing strategy. Let $c \in C$ be such that $eval_0(c) \neq *$ and $eval^{\sigma}(c) = \delta > 0$, but where for every customer $c'' \in \bigcup\{N(c') : c' \in N(c)\}$, we have that $\delta < eval^{\sigma}(c'')$. If σ_{-c} is the bribing strategy obtained from σ by removing the bribe from c, we have that $\mathbf{r}(\sigma_{-c}) \geq \mathbf{r}(\sigma)$.*

Therefore, we can assume that at most one old customer has not been fully bribed.

Suppose now that there is in fact *one bribed old customer that has not been fully bribed*. Let us call this old customer c and further suppose that it has been bribed δ where $0 < \delta < 1$. We will again show that this yields yet another contradiction with the optimality of σ. We have the capacity to increase this bribe to 1 at a cost of $1-\delta$ (which we can do, given the remaining resource). The $\mathbb{P}$-rating of all the customers within $N(c)$ will increase with the cumulative increase (only due to new pendant neighbours) being

$$n\frac{1+\epsilon}{2} - n\frac{\delta+\epsilon}{2} = n\frac{1-\delta}{2}.$$

Hence we obtain an increase in revenue for n sufficiently large ($n \geq 3$). This contradicts the optimality of σ. Henceforth, we assume that, without loss of generality, any optimal bribing strategy σ on (C, E), with initial evaluation $eval_0$, is necessarily such that only old customers are bribed and bribed old customers are fully bribed.

Suppose now that the bribing strategy σ *bribes less than k old customers*; so, there is an old customer c that has not been bribed. Let us amend σ to obtain a new bribing strategy σ' by bribing c so that $\sigma(c) = 1$. This costs us 1 unit of resource. There is no customer of C such that its $\mathbb{P}$-rating decreases, and the cumulative increase in $\mathbb{P}$-rating of the n new pendant customers adjacent to c is

$$n\left(\frac{1+\epsilon}{2} - \epsilon\right) = n\left(\frac{1-\epsilon}{2}\right) > \frac{n(2n+3)}{2(2n+5)} > \frac{n}{4}$$

which is strictly greater than 1 (the amount invested) for n sufficiently large ($n \geq 5$). This contradicts the optimality of σ. Furthermore, it is clear that more than k old customers could not have been bribed since the initial utility of the network totals only k and each old customer is bribed by 1.

Finding an independent set of size k. We have shown above that the optimal bribing strategy σ on (C, E) is such that only old customers are bribed, all bribed old customers are fully bribed, and exactly k old customers are bribed.

Consider now the revenue accruing from our optimal bribing strategy σ. Irrespective of which k old customers are fully-bribed, the increase in $\mathbb{P}$-rating due to these old customers is equal to:

$$\frac{(1+(n+3)\epsilon)}{n+4} - \epsilon = \frac{1-\epsilon}{n+4},$$

and the $\mathbb{P}$-rating of the pendant customers adjacent to each of these bribed old customers increases by:

$$\frac{1+\epsilon}{2} - \epsilon = \frac{1-\epsilon}{2}.$$

All that remains is to compute the revenue accruing due to the new edge customers adjacent to each of these bribed old customers (as the $\mathbb{P}$-rating of any other old or new customer does not change). However, this depends upon how many bribed old customers each new edge customer is adjacent to. Let m_i denote the number of new edge customers adjacent to i bribed old customers, for $i = 1, 2$. If a new edge customer c is adjacent to 1 bribed old customer then its increase in $\mathbb{P}$-rating is

$$\frac{(1+\epsilon)}{2} - \epsilon = \frac{(1-\epsilon)}{2}$$

and if it is adjacent to two bribed old customers then its increase in $\mathbb{P}$-rating is

$$\frac{(2+\epsilon)}{3} - \epsilon = \frac{2(1-\epsilon)}{3}.$$

So, the total increase in revenue is

$$m_1\frac{(1-\epsilon)}{2} + m_2\frac{2(1-\epsilon)}{3}.$$

We also know that by counting the edges joining bribed old customers and their adjacent new edge customers, we obtain that $3k = 2m_2 + m_1$. Hence, the total increase in $\mathbb{P}$-rating due to new edge customers is equal to

$$\begin{aligned} & m_1\frac{(1-\epsilon)}{2} + m_2\frac{2(1-\epsilon)}{3} \\ &= (3k - 2m_2)\frac{(1-\epsilon)}{2} + m_2\frac{2(1-\epsilon)}{3} \\ &= \frac{3k(1-\epsilon)}{2} - m_2\frac{(1-\epsilon)}{3}. \end{aligned}$$

So, the revenue due to the bribing strategy σ is:

$$\begin{aligned} & \frac{k(1-\epsilon)}{n+4} + \frac{nk(1-\epsilon)}{2} + \frac{3k(1-\epsilon)}{2} - m_2\frac{(1-\epsilon)}{3} - k \\ &= \quad (1-\epsilon)[\frac{k}{n+4} + \frac{k(n+3)}{2} - \frac{m_2}{3}] - k. \end{aligned}$$

Clearly this revenue is largest when m_2 is 0, and if $m_2 > 0$ then the revenue is less than this maximal value. Also, when m_2 is 0 this revenue is exactly equal to ρ. Hence, as we started with a yes-instance of NVKL, we must have that $m_2 = 0$, i.e., that no edge customer is adjacent to two bribed old customers. Thus, the k vertices of G corresponding to the k bribed old customers in C form an independent set, and (G, k) is a yes-instance of ISREG(3). □

As a direct consequence of Propositions 5 and 6 we obtain the following:

Theorem 10. BRIBE-NVKL *is* **NP**-*complete.*

In summary, we have been able to prove the **NP**-completeness of BRIBE-NVKL by giving a reduction from ISREG(3). This is an important finding, that significantly strengthens the value of personalised rating systems and their resistance to bribery, as we have demonstrated that we cannot compute an optimal bribing strategy, nor any strategy guaranteeing at least a given reward, in a reasonable amount of time; that is, of course, unless **P** = **NP**.

5 Conclusion

We have investigated the problem of manipulation in a network-based rating system, in which customers' form their personalised rating aggregating the opinion of their peers and an external attacker is allowed to elaborate bribing strategies to modify them. The framework, first elaborated by Grandi and Turrini (2016), has been shown to be manipulable when the attacker has full knowledge of the underlying network. In this paper we have shown that despite this fact, manipulation is intractable in practice, as the problem of computing the existence of a manipulation strategy guaranteeing a given reward and thus an optimal one, what we called BRIBE-NVKL, is **NP**-complete. This, we find, is a major strengthening for the practical applicability of the personalised rating framework.

However, it has to be emphasised that our results are confined to worst-case complexity analysis and it is therefore necessary to analyse alternative methods for manipulation.

These include studying the parameterised complexity of various sub-problems (see, e.g., Faliszewski and Niedermeier, 2014). Alternatively, we can think of devising ways to compute an approximate or satisfactory solution that yields at least a positive return. More specifically, we may still be able to salvage something of the $\mathbb{P}$-greedy approach from Grandi and Turrini (2016). We saw through our Example 2 that whilst not yielding the optimal amount of revenue, we can still compute a profitable return. We can approach this question from a slightly less formal but nevertheless important angle and seek to obtain a number of experimental results concerning the performance of greedy algorithms.

Acknowledgments

Umberto Grandi acknowledges the support of the Labex CIMI project *"Social Choice on Networks"* (ANR-11-LABX-0040-CIMI).

References

Alon, N.; Feldman, M.; Lev, O.; and Tennenholtz, M. 2015. How robust is the wisdom of the crowds? In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI)*.

Andersen, R.; Borgs, C.; Chayes, J.; Feige, U.; Flaxman, A.; Kalai, A.; Mirrokni, V.; and Tennenholtz, M. 2008. Trust-based recommendation systems: An axiomatic approach. In *Proceedings of the 17th International Conference on World Wide Web (WWW)*.

Bobadilla, J.; Ortega, F.; Hernando, A.; and Gutierrez, A. 2013. Recommender systems survey. *Knowledge-Based Systems* 46:109 – 132.

Brill, M.; Conitzer, V.; Freeman, R.; and Shah, N. 2016. False-name-proof recommendations in social networks. In *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Bu, N. 2013. Unfolding the mystery of false-name-proofness. *Economics Letters* 120(3):559–561.

Conitzer, V., and Walsh, T. 2016. Barriers to manipulation in voting. In Brandt, F.; Conitzer, V.; Endriss, U.; Lang, J.; and Procaccia, A. D., eds., *Handbook of Computational Social Choice*. Cambridge University Press. chapter 6.

Conitzer, V.; Immorlica, N.; Letchford, J.; Munagala, K.; and Wagman, L. 2010. False-Name-Proofness in Social Networks. In *6th International Workshop on Internet and Network Economics (WINE)*.

Conte, R., and Paolucci, M. 2002. *Reputation in Artificial Societies: Social Beliefs for Social Order*. Kluwer Academic Publishers.

Faliszewski, P., and Niedermeier, R. 2014. Parameterization in computational social choice. In Kao, M.-Y., ed., *Encyclopedia of Algorithms*. Springer Berlin Heidelberg.

Faliszewski, P., and Rothe, J. 2016. Control and bribery in voting. In Brandt, F.; Conitzer, V.; Endriss, U.; Lang, J.; and Procaccia, A. D., eds., *Handbook of Computational Social Choice*. Cambridge University Press. chapter 7.

Garcin, F.; Faltings, B.; and Jurca, R. 2009. Aggregating reputation feedback. In *Proceedings of the International Conference on Reputation*.

Garey, M. R., and Johnson, D. S. 1990. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co.

Gibbard, A. 1973. Manipulation of voting schemes: A general result. *Econometrica* 41(4):587–601.

Grandi, U., and Turrini, P. 2016. A network rating system and its resistance to bribery. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence, (IJCAI)*.

Grandi, U. 2017. Social choice and social networks. In Endriss, U., ed., *Trends in Computational Social Choice*. AI Access. chapter 9, 169–184.

Lev, O., and Tennenholtz, M. 2017. Group recommendations: Axioms, impossibilities, and random walks. *CoRR* abs/1707.08755.

Sabater, J., and Sierra, C. 2005. Review on computational trust and reputation models. *Artificial Intelligence Review* 24(1):33–60.

Satterthwaite, M. A. 1975. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory* 10(2):187 – 217.

Tennenholtz, M. 2008. Game-theoretic recommendations: some progress in an uphill battle. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Todo, T., and Conitzer, V. 2013. False-name-proof matching. In *Proceedings of the 12th International Conference on Autonomous Agents and Multi-agent Systems (AAMAS)*.