Coverage and Vacuity in Network Formation Games

Gili Bielous
School of Computer Science and Engineering, The Hebrew University, Jerusalem, Israel
gili.bielous@mail.huji.ac.il

Orna Kupferman
School of Computer Science and Engineering, The Hebrew University, Jerusalem, Israel
orna@cs.huji.ac.il

Abstract

The frameworks of coverage and vacuity in formal verification analyze the effect of mutations applied to systems or their specifications. We adopt these notions to network formation games, analyzing the effect of a change in the cost of a resource. We consider two measures to be affected: the cost of the Social Optimum and extremums of costs of Nash Equilibria. Our results offer a formal framework to the effect of mutations in network formation games and include a complexity analysis of related decision problems. They also tighten the relation between algorithmic game theory and formal verification, suggesting refined definitions of coverage and vacuity for the latter.

2012 ACM Subject Classification Theory of computation → Algorithmic game theory; Theory of computation → Network games; Theory of computation → Network formation; Software and its engineering → Formal methods

Keywords and phrases Network Formation Games, Vacuity, Coverage

Digital Object Identifier 10.4230/LIPIcs.CSL.2020.10

Related Version A full version of the paper is available at `https://www.cs.huji.ac.il/~ornak/publications/csl20.pdf`.

1 Introduction

Following the emergence of the Internet, there has been an explosion of studies employing game-theoretic analysis to explore applications such as network formation and routing in computer networks [21, 1, 20, 4]. In *network-formation games* (for a survey, see [37]), the network is modeled by a weighted graph. The weight of an edge indicates the cost of activating the transition it models, which is independent of the number of times the edge is used. Players have reachability objectives, each given by a source and a target vertex. Under the common Shapley cost-sharing mechanism, the cost of an edge is shared evenly by the players that use it. The players are selfish agents who attempt to minimize their own costs, rather than to optimize some global objective. In network-design settings, this would mean that the players selfishly select a path instead of being assigned one by a central authority. The study of networks from a game-theoretic point of view focuses on optimal strategies for the underlying players, stable outcomes of a given setting, namely equilibrium points, and outcomes that are optimal for the society as a whole.

A different type of reasoning about networks is the study of their on-going behaviors. In particular, in recent years we see growing use of formal-verification methods in the context of software-defined networks [34, 33]. The study of networks from a formal-verification point of view focuses on specification and verification of their behavior. The primary problem here is *model checking*: given a system (in particular, a network) and a specification for its desired behavior, decide whether the system satisfies the specification [18]. Typically, the system is given by means of a labeled graph and the specification is given by a temporal-logic formula. An important element in model-checking methodologies is an assessment of the quality of the modeling of the system and the specifications as well as the exhaustiveness of the model-checking process. Researchers have developed a number of *sanity checks*, aiming to detect errors in the modeling [27]. Two leading sanity checks are *vacuity* and *coverage*. In vacuity, the goal is to detect cases where the system satisfies the specification in some unintended trivial way [10, 31, 14]. In coverage, the goal is to increase the exhaustiveness of the specification by detecting components of the system that do not play a role in the verification process [24, 25, 16, 15]. Both vacuity and coverage checks are based on analyzing the effect of applying *local mutations* to the system or the specification. The intuition is that model checking of an exhaustive well-formed specification should be sensitive to such mutations.

Beyond the practical importance of sanity checks, their study highlights some general important theoretical properties regarding the sensitivity of systems and specifications to mutations. Examples to such properties include *duality* between mutations applied to the system and the specification [29], and trade-offs between desired and undesired insensitivity to mutations (for example, fault tolerance is associated with a desired insensitivity to mutations) [17]. A fundamental property of mutations in the context of formal verification is *monotonicity*: mutations to temporal-logic formulas are monotone, in the sense that if ψ is a formula and φ is a sub-formula of ψ that appears in a positive polarity (that is, nested in an even number of negations), then when we mutate ψ to ψ' by replacing φ by φ', then $\psi' \rightarrow \psi$ iff $\varphi' \rightarrow \varphi$. Monotonicity turns out to be a very helpful property in the context of vacuity checking. Indeed, the basic notion in vacuity is of a subfumula φ *not affecting* the satisfaction of a specification ψ. Formally, consider a system $\mathcal{S}$ satisfying a specification ψ. A subformula φ of ψ *does not affect* (the satisfaction of) ψ in $\mathcal{S}$ if $\mathcal{S}$ also satisfies all specifications obtained by mutating φ to some other subformula [10]. Thanks to monotonicity, we can check whether φ affects ψ by examining only the most challenging mutation, namely one that replaces φ by false and the most helpful mutation, namely one that replaces φ by true.

Our goal in this paper is to examine the sensitivity of network-formation games (NFGs, for short) to mutations applied to costs. While our study adopts from formal verification the notion of mutation-based analysis, we examine the effect of mutations on measures from game theory: the cost of stable and optimal outcomes. Recall that a strategy of a player in an NFG is a path from a source to a target vertex. A *profile* in the game is a vector of strategies, one for each player. A *Social Optimum* (SO) is a profile that minimizes the total cost to all players. A *Nash equilibrium* (NE) is a profile in which no player can decrease her cost by a unilateral deviation from her current strategy, that is, assuming that the strategies of the other players do not change.

Consider an NFG N. We say that the edge e of N *SO-affects* N if a change in the cost of e leads to a change in the cost of the SO. Formally, there exists $x \geq 0$ such that the cost of the SO profiles in N is different from the cost of the SO profiles in $N[e \leftarrow x]$, that is N with e being assigned cost x. We consider the function $cost^{e}_{SO}(N) : \mathbb{R} \rightarrow \mathbb{R}$, mapping a cost $x \geq 0$ to the cost of the SO profiles in $N[e \leftarrow x]$. That is, $cost^{e}_{SO}(N)$ describes the cost of the SO in N as a function of the cost of the edge e. We say that $cost_{SO}$ is monotonically increasing if for every NFG N and edge e of N, the function $cost^{e}_{SO}(N)$ is monotonically increasing. Likewise, $cost_{SO}$ is continuous if for every NFG N and edge e, the function $cost^{e}_{SO}(N)$ is continuous. For the best and worst NEs, we similarly define when an edge e *bNE-affects* and *wNE-affects* N, and define the functions $cost_{bNE}$ and $cost_{wNE}$, which describe the cost of the best and worst NEs as a function of the cost of an edge.

Our first set of results concerns the way edge costs affect the SO. Here, the results are quite expected: $cost_{SO}$ is monotonically increasing and continuous, which leads to simple solutions to related decision problems: as is the case with model checking and temporal-logic specifications, we can decide whether an edge e SO-affects N by checking the cost of the SO in $N[e \leftarrow 0]$ and $N[e \leftarrow \infty_N]$, for a sufficiently large cost ∞_N. This leads to Δ_2^P and Θ_2^P upper bounds (depending on whether costs are given in binary or unary, respectively), which we show to be tight. Also, we show that it is NP-complete and DP-complete to decide whether we can mutate a cost in a way that would cause the SO to be below or agree exactly with, respectively, a given threshold. The technically challenging results here are the Δ_2^P-lower bound (it is tempting to believe that thanks to monotonicity, we could decide whether e SO-affects N using only logarithmically many queries to an NP oracle that bounds the SO) and the DP upper bound (the upper and lower bounds on the SO that we can obtain by querying an NP and a co-NP oracle need not be associated with the same edge).

Things become unexpected when we turn to study effects on the costs of the best and worst NEs. Here an edge may affect the bNE without participating in profiles that are NEs, and may thus affect the bNE both positively and negatively. In model checking, this is related to coverage and vacuity in a setting with multiple occurrences of subformulas. For example, the atomic proposition p appears in the formula $\psi = (\varphi_1 \rightarrow p) \wedge (p \rightarrow \varphi_2)$ both positively and negatively. Consequently, we cannot decide whether p affects the satisfaction of ψ by examining its replacement by only true or false (in the context of vacuity), and we do not know the effect of mutating p in the system on the satisfaction of ψ (in the context of coverage). We show that $cost_{bNE}$ is neither monotone nor continuous, and in fact a change in the cost of an edge may incentivize players in surprising ways. In particular (see Figure 5), an edge e may not participate in any bNE in $N[e \leftarrow x]$, for all $x \geq 0$, and still the bNE may decrease as we increase the cost of e. We show that these challenges can be overcome by more restricted notions such as piecewise monotonicity and monotonicity on the participation of the mutated edge in bNE profiles. In particular, we show that these notions produce the same (tight) complexity bounds for the analogous decision problems we introduce for the SO. We note that while the general phenomenon of non-monotonicity is known (e.g., Braess' Paradox [12], the effectiveness of burning money [23, 36] or tax increase [19]), we are the first, to the best of our knowledge, to provide a comprehensive study of effects caused by cost mutation.

Our results on NFGs give rise to two research directions in coverage and vacuity in formal verification. The first arises from the segmentation of $\mathbb{R}^+$ induced by the non-monotonicity of the bNE, which suggests a similar segmentation in the context of multi-valued specification formalisms [2]. The second is a study of coverage and vacuity in formalisms for specifying strategic on-going behaviors [3, 13]. We discuss these research directions in Section 5.

Due to lack of space, some of the proofs are omitted, and can be found in the full version, as listed above.

2 Preliminaries

2.1 Network formation games

A *network formation game* (NFG) is $N = \langle k, V, E, c, \gamma \rangle$, where k is a number of players, V is a set of vertices, $E \subseteq V \times V$ is a set of directed edges, $c : E \rightarrow \mathbb{R}^+$, where $\mathbb{R}^+$ is the set of positive real numbers including 0, is a cost function that maps each edge to the cost of forming it, and $\gamma = \{\langle s_1, t_1 \rangle, ..., \langle s_k, t_k \rangle\}$ is a set of objectives, each specifying a source and a target vertex per player. Thus, for all $1 \leq i \leq k$, the objective of player i is to form a path from s_i to t_i. A *strategy* for player i is a simple path $\pi_i \subseteq E$ from s_i to t_i. Note that since the path is simple, then π_i is indeed a subset of E. A *profile* $P = \langle \pi_1, ..., \pi_k \rangle$ is a vector of strategies, one for each player. For an edge $e \in E$, we denote by $used_P(e)$ the number of players that use e in their strategy in P, thus these with $e \in \pi_i$. We say that $e \in P$ if $used_P(e) > 0$.

Players pay the cost of forming edges they use. If players share an edge, they also share its cost. Thus, the cost of a strategy π_i in a profile P is $cost_{N,P}(\pi_i) = \sum_{e \in \pi_i} \frac{c(e)}{used_P(e)}$. Note that since c is positive, it is indeed sufficient to consider only simple paths as strategies. The *cost* of P in N is the sum of costs of its strategies, that is $cost(N,P) = \sum_{i=1}^{k} cost_{N,P}(\pi_i)$. Equivalently, $cost(N,P) = \sum_{e \in P} c(e)$.

A *Social Optimum (SO)* of N is a profile with minimal cost. That is, a profile P is an SO if for every other profile P' we have that $cost(N,P) \leq cost(N,P')$. Note that there may be several profiles that are a social optimum. We denote by $SO(N)$ and $cost_{SO}(N)$ the set of such profiles and their cost, respectively.

We say that the profile P is a *Nash Equilibrium (NE)* in N if no player can decrease her cost by deviating to another strategy assuming the other players stay in their strategies[1]. Formally, for all $1 \leq i \leq k$ and every $\pi'_i \neq \pi_i$, the cost of π'_i in $P' = \langle \pi_1, ..., \pi_{i-1}, \pi'_i, \pi_{i+1}, ..., \pi_k \rangle$ is no lower than the cost of π_i in P, i.e. $cost_{N,P}(\pi_i) \leq cost_{N,P'}(\pi'_i)$. A *best NE (bNE)* in N is an NE profile with minimal cost, i.e. a profile P is bNE iff P is an NE, and for every profile P' that is an NE, we have $cost(N,P) \leq cost(N,P')$. We denote by $bNE(N)$ and $cost_{bNE}(N)$ the set of profiles that are bNE, and their cost, respectively.

We dually define a *worst NE (wNE)* to be an NE profile with maximal cost, and denote by $wNE(N)$ and $cost_{wNE}(N)$ the set of such profiles and their cost, respectively. The *Price of Stability (PoS)* of N is the ratio between the cost of the bNE and the SO, that is, $PoS(N) = \frac{cost_{bNE}(N)}{cost_{SO}(N)}$.

▶ **Example 1.** Consider the NFG N appearing in Figure 1.

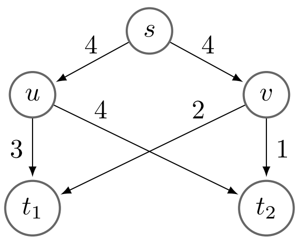

Figure 1 The NFG N.

Table 1 Players' costs in N.

Player 2 / Player 1	π_2^1 $s \to u \to t_2$	π_2^2 $s \to v \to t_2$
π_1^1 $s \to u \to t_1$	6 5	5 7
π_1^2 $s \to v \to t_1$	8 6	3 4

Assume that N is formed by two players. The first has objective $\langle s, t_1 \rangle$. The available strategies for her are $\pi_1^1 = \{(s,u),(u,t_1)\}$ and $\pi_1^2 = \{(s,v),(v,t_1)\}$. The second player has objective $\langle s, t_2 \rangle$. The available strategies for her are $\pi_2^1 = \{(s,u),(u,t_2)\}$ and $\pi_2^2 = \{(s,v),(v,t_2)\}$. If Player 1 choses the strategy π_1^1 and Player 2 uses the strategy π_2^1, then they share the cost of the edge (s,u), and their costs are $\frac{4}{2} + 3 = 5$ and $\frac{4}{2} + 4 = 6$ respectively. Table 1 describes the costs of the two players in the different profiles.

The profile with the lowest cost is $P = \langle \pi_1^2, \pi_2^2 \rangle$. Therefore, $SO(N) = \{P\}$, with cost $cost_{SO}(N) = 7$. Note that P is also the only NE in N. It is an NE since for the deviation $P' = \langle \pi_1^1, \pi_2^2 \rangle$, it holds that $4 = cost_{N,P}(\pi_1^2) < cost_{N,P'}(\pi_1^1) = 7$ and for the deviation

[1] Throughout this paper, we consider pure strategies and pure deviations, as is the case for the vast literature on cost-sharing games.

$P'' = \langle \pi_1^2, \pi_2^1 \rangle$ it holds that $3 = cost_{N,P}(\pi_2^2) < cost_{N,P''}(\pi_2^1) = 8$. It is the only NE in N since for every other profile there is a beneficial deviation. Therefore, P is both a bNE and a wNE. Since the bNE and the SO coincide, it follows that $PoS(N) = 1$. ◀

Consider an edge $e \in E$ and a value $x \in \mathbb{R}^+$. We denote by $c[e \leftarrow x]$ the cost function that agrees with c on every edge except e, which is assigned x. That is, $c[e \leftarrow x](e) = x$, and for all edge $e' \neq e$, we have $c[e \leftarrow x](e') = c(e')$. Let $N = \langle k, V, E, c, \gamma \rangle$, and let $e \in E$. We denote by $N[e \leftarrow x]$ the network obtained from N by changing the cost of e to x. Thus, $N[e \leftarrow x] = \langle k, V, E, c[e \leftarrow x], \gamma \rangle$.

Let c_1 and c_2 be cost functions. We say that c_2 *bounds* c_1 *from above*, denoted $c_1 \leq c_2$, if for all $e \in E$, we have $c_1(e) \leq c_2(e)$. We extend the notation to NFGs. Let $N_1 = \langle k, V, E, c_1, \gamma \rangle$ and $N_2 = \langle k, V, E, c_2, \gamma \rangle$ be two NFGs that differ only on their cost functions. If $c_1 \leq c_2$, we say that N_2 *bounds* N_1 *from above*, denoted $N_1 \leq N_2$.

▶ **Lemma 2.** *Let N_1 and N_2 be two NFGs that differ only on their cost functions. If $N_1 \leq N_2$, then for every profile P, we have $cost(N_1, P) \leq cost(N_2, P)$.*

2.2 Affecting edges in NFGs

Consider an NFG N and an edge e of N. We say that the edge e *SO-affects* N if there exists $x \geq 0$ such that $cost_{SO}(N[e \leftarrow x]) \neq cost_{SO}(N)$. That is, when changing the cost of e to x, the cost of the SO profiles of N changes. We define *bNE-affects*, *wNE-affects*, and *PoS-affects* in a similar way, referring to the costs of the best and worst NEs, and the PoS.

▶ **Example 3.** Consider the NFG N from Example 1, and consider the edge $e = (s, v)$. The edge e SO-affects N, since, for example, for $N[e \leftarrow 2]$ we have that $\langle \pi_1^2, \pi_2^2 \rangle$ is an SO with cost $5 < 7 = cost_{SO}(N)$. As another example, for $N[e \leftarrow 10]$ we have that $\langle \pi_1^1, \pi_2^1 \rangle$ is an SO with cost $11 > 7 = cost_{SO}(N)$. Next, consider the edge $e = (u, t_1)$. For every $x \geq 0$, we have $cost(N[e \leftarrow x], \langle \pi_1^1, \pi_2^1 \rangle) = x + 8$, $cost(N[e \leftarrow x], \langle \pi_1^1, \pi_2^2 \rangle) = x + 9$, $cost(N[e \leftarrow x], \langle \pi_1^2, \pi_2^1 \rangle) = 14$, and $cost(N[e \leftarrow x], \langle \pi_1^2, \pi_2^2 \rangle) = 7$. Therefore, $cost_{SO}(N[e \leftarrow x]) = \min\{x + 8, x + 9, 14, 7\} = 7 = cost_{SO}(N)$, and so e does not SO-affect N.

We proceed to bNE and wNE. Here, the change may affect the stability of profiles, and not just their cost. Consider the edge $e = (s, u)$. Table 2 describes the costs of the different profiles of $N[e \leftarrow (1 - \varepsilon)]$, for some $0 < \varepsilon < 1$.

Table 2 Costs in $N[\langle s, u \rangle \leftarrow (1 - \varepsilon)]$.

Player 2 Player 1	π_2^1 $s \to u \to t_2$	π_2^2 $s \to v \to t_2$
π_1^1 $s \to u \to t_1$	$4\frac{1}{2} - \frac{\varepsilon}{2}$ $3\frac{1}{2} - \frac{\varepsilon}{2}$	5 $4 - \varepsilon$
π_1^2 $s \to v \to t_1$	$5 - \varepsilon$ 6	3 4

Table 3 Costs in $N[\langle u, t_1 \rangle \leftarrow x]$.

Player 2 Player 1	π_2^1 $s \to u \to t_2$	π_2^2 $s \to v \to t_2$
π_1^1 $s \to u \to t_1$	6 $2 + x$	5 $4 + x$
π_1^2 $s \to v \to t_1$	8 6	3 4

We previously saw that the only NE profile in N is $P = \langle \pi_1^2, \pi_2^2 \rangle$, with cost 7, and therefore it is both the bNE and the wNE. We can see that the cost of P is minimal for $N[e \leftarrow (1 - \varepsilon)]$. However, P is no longer an NE. Indeed, for the profile $P' = \langle \pi_1^1, \pi_2^2 \rangle$, obtained by a deviation of Player 1, we have that $4 - \varepsilon = cost_{N[e \leftarrow 1-\varepsilon], P'}(\pi_1^1) < cost_{N[e \leftarrow 1-\varepsilon], P}(\pi_1^2) = 4$. For $N[e \leftarrow (1 - \varepsilon)]$, the only NE profile is $\langle \pi_1^1, \pi_2^1 \rangle$, with cost $8 - \varepsilon$. For $0 < \varepsilon < 1$ it therefore holds that $7 = cost_{bNE}(N) < cost_{bNE}(N[e \leftarrow 1 - \varepsilon]) = 8 - \varepsilon$, and the same for wNE.

Therefore, the edge e both bNE-affects and wNE-affects N. Furthermore, e PoS-affects N, as $PoS(N) = 1$ and $PoS(N[e \leftarrow 1 - \varepsilon]) = \frac{8-\varepsilon}{7} > 1$.

Next, consider the edge $e = (u, t_1)$. We show that e does not bNE-affect nor does it wNE-affect N. To see this, consider the costs of the different profiles of $N[e \leftarrow x]$ for $x \geq 0$, described in Table 3. It can be easily verified that, for all $x \geq 0$, the only NE in $N[e \leftarrow x]$ is $\langle \pi_1^2, \pi_2^2 \rangle$. Therefore, $cost_{bNE}(N[e \leftarrow x]) = cost_{wNE}(N[e \leftarrow x]) = 7$. As e neither SO-affect nor bNE-affect N, it follows that e does not PoS-affect N.

It is also worth noting that it is not always the case that an edge either both bNE-affects and wNE-affects or both does not bNE-affect and wNE-affect N. As an example, consider the edge $e = (u, t_2)$. The cost table of $N[e \leftarrow x]$ appears in Table 4.

Table 4 Costs in $N[\langle u, t_2 \rangle \leftarrow x]$.

Player 2 / Player 1	π_2^1 $s \to u \to t_2$	π_2^2 $s \to v \to t_2$
π_1^1 $s \to u \to t_1$	$2 + x$ / 5	5 / 7
π_1^2 $s \to v \to t_1$	$4 + x$ / 6	3 / 4

It is not hard to see that for $0 \leq x \leq 3$, it holds that $P_1 = \langle \pi_1^1, \pi_2^1 \rangle$ and $P_2 = \langle \pi_1^2, \pi_2^2 \rangle$ are NEs in $N[e \leftarrow x]$. However, $cost(N[e \leftarrow x], P_1) = 7 + x$ and $cost(N[e \leftarrow x], P_2) = 7$. Therefore, $cost_{bNE}(N[e \leftarrow x]) = \min\{7 + x, 7\} = 7$, and $cost_{wNE}(N[e \leftarrow x]) = \max\{7 + x, 7\} = 7 + x$. Since for all $x > 3$, the profile P_2 is the only NE in $N[e \leftarrow x]$, it follows that e does not bNE-affect N, and e wNE-affects N. ◀

2.3 Monotonicity and continuity

Consider a function $f : \mathbb{R} \to \mathbb{R}$. We say that f is *monotonically increasing* if for all $x_1, x_2 \in \mathbb{R}$, we have that $x_1 \leq x_2$ implies $f(x_1) \leq f(x_2)$. For $x_0 \in \mathbb{R}$, we say that f is *continuous at* x_0 if for every $\varepsilon > 0$ there exists $\delta > 0$ such that for all $x \in \mathbb{R}$, if $|x - x_0| < \delta$ then $|f(x) - f(x_0)| < \varepsilon$. Then, we say that f is *continuous* if f is continuous at x_0 for all $x_0 \in \mathbb{R}$.

For an edge $e \in E$, we define the function $cost_{SO}^e(N) : \mathbb{R} \to \mathbb{R}$ by $cost_{SO}^e(N)(x) = cost_{SO}(N[e \leftarrow x])$ if $x \geq 0$, and $cost_{SO}^e(N)(x) = cost_{SO}(N[e \leftarrow 0])$ otherwise. That is, $cost_{SO}^e(N)$ is the cost of the SO in N as a function of the cost of the edge e. We say that $cost_{SO}$ is monotonically increasing, if for every NFG N and edge e of N, the function $cost_{SO}^e(N)$ is monotonically increasing. That is, $cost_{SO}$ is monotonically increasing if an increase in the cost of any edge, for any NFG, can only cause an increase in the cost of the SO. Likewise, $cost_{SO}$ is continuous, if for every NFG N and edge e, the function $cost_{SO}^e(N)$ is continuous. We define the monotonicity and the continuity of $cost_{bNE}$, $cost_{wNE}$ and PoS in a similar way.

3 Affecting the Social Optimum

In this section we study the sensitivity of the SO to cost mutations. We first study the monotonicity and continuity of $cost_{SO}$, and then the complexity of relevant decision problems.

3.1 Monotonicity and continuity of the SO

▶ **Theorem 4** ($cost_{SO}$ is monotone)**.** *For every NFG N and edge e of N, the function $cost^e_{SO}(N)$ is monotone.*

Proof. Let N_1 and N_2 be NFGs that differ only in their cost functions. We prove that if $N_1 \leq N_2$, then $cost_{SO}(N_1) \leq cost_{SO}(N_2)$. In particular, this holds for N_1 and N_2 being N with cost functions that differ only in the cost of e. Let $P_1 \in SO(N_1)$ and let $P_2 \in SO(N_2)$. By the minimality of the SO for N_1, we get that $cost(N_1, P_1) \leq cost(N_1, P_2)$. By Lemma 2, as $N_1 \leq N_2$, we have that $cost(N_1, P_2) \leq cost(N_2, P_2)$. Therefore, $cost(N_1, P_1) \leq cost(N_2, P_2)$, and hence $cost_{SO}(N_1) \leq cost_{SO}(N_2)$. ◀

Since $cost_{SO}$ is monotonically increasing, a sufficient condition for an edge not to SO-affect the network is based on comparing the cost of the SO in the two extreme costs for the edge. The lowest cost is 0. For the highest cost, let ∞_N be a sufficiently large value for a cost of an edge to be considered extreme in N, in the sense that if an edge e with cost ∞_N is in some strategy, then the cost of that strategy is guaranteed to be larger than the cost of all strategies that do not contain e. For example, we can define ∞_N to be $1 + \sum_{e \in E} c(e)$.

▶ **Lemma 5.** *For every NFG N and edge e of N, the edge e does not SO-affect N iff $cost_{SO}(N[e \leftarrow 0]) = cost_{SO}(N[e \leftarrow \infty_N])$.*

Proof. Since $N[e \leftarrow 0] \leq N[e \leftarrow \infty_N]$ and the function $cost_{SO}(N)$ is monotonically increasing, then $cost_{SO}(N[e \leftarrow 0]) = cost_{SO}(N[e \leftarrow \infty_N])$ implies that for all $x \geq 0$, we have $cost_{SO}(N[e \leftarrow 0]) = cost_{SO}(N[e \leftarrow x]) = cost_{SO}(N[e \leftarrow \infty_N])$. Thus, for all $x \geq 0$, we have $cost_{SO}(N) = cost_{SO}(N[e \leftarrow x])$, so the cost of e does not SO-affect N. For the other direction, if the cost of e does not SO-affect N, then, by definition, for all $x \geq 0$, we have that $cost_{SO}(N) = cost_{SO}(N[e \leftarrow x])$. In particular, $cost_{SO}(N[e \leftarrow 0]) = cost_{SO}(N[e \leftarrow \infty_N])$, and we are done. ◀

Note that it follows that for an NFG N and edge e in it, if there is a profile $P \in SO(N)$ such that $e \in P$ and $c(e) > 0$, then e SO-affects N, as reducing its cost to 0 reduces also the cost of the SO.

In case e SO-affects N, we can characterize the behavior of $cost_{SO}(N[e \leftarrow x])$ as follows.

▶ **Lemma 6.** *Consider an NFG N and an edge e of N. If e SO-affects N, then there is a value $x \in \mathbb{R}$ such that the following hold.*

1. *For all values y with $y > x$, the edge e does not participate in any profile in $SO(N[e \leftarrow y])$ and $cost_{SO}(N[e \leftarrow y]) = x + cost_{SO}(N[e \leftarrow 0])$.*
2. *For all values y with $y < x$, the edge e participates in at least one profile in $SO(N[e \leftarrow y])$ and $cost_{SO}(N[e \leftarrow y]) = y + cost_{SO}(N[e \leftarrow 0])$.*
3. *The edge e participates in at least one profile in $SO(N[e \leftarrow x])$ and $cost_{SO}(N[e \leftarrow x]) = x + cost_{SO}(N[e \leftarrow 0])$.*

Proof. Since e SO-affects N, then, by Lemma 5, we have that $cost_{SO}(N[e \leftarrow 0]) < cost_{SO}(N[e \leftarrow \infty_N])$. It is not hard to see that taking x to be $\min\{y : cost_{SO}(N[e \leftarrow y]) = cost_{SO}(N[e \leftarrow \infty_N])\}$ satisfies the conditions in the lemma. In particular, when e participates in all profiles in the SO, then $x = \min \emptyset = \infty$. ◀

▶ **Theorem 7.** *For every NFG N and edge e of N, the function $cost^e_{SO}(N)$ is continuous.*

Proof. Consider an NFG N and edge e of N. First, if the edge e does not SO-affect N, then $cost^e_{SO}(N)$ is constant and therefore continuous. Otherwise, by Lemma 6, there is a value $x \in \mathbb{R}$ such that for all values y with $y \geq x$, we have that $cost_{SO}(N[e \leftarrow y]) = x + cost_{SO}(N[e \leftarrow 0])$, and for all values y with $y < x$, we have that $cost_{SO}(N[e \leftarrow y]) = y + cost_{SO}(N[e \leftarrow 0])$. Thus, continuity in all points except x follows immediately from continuity of linear functions. For the point x, Lemma 6 implies that for all $\varepsilon > 0$, we have that $f(x+\epsilon) - f(x) = 0$, and $f(x) - f(x-\epsilon) = \epsilon$, so $cost^e_{SO}(N)$ is continuous also at x. ◀

3.2 Decision problems

The SO-cost decision problem is the problem of deciding, given an NFG N and a threshold $\kappa \geq 0$, whether $cost_{SO}(N) \leq \kappa$. The SO-cost problem is NP-complete [37]. In this section we study the following related decision problems.

1. Edge-SO-affects: Given an NFG N and an edge e of N, does e SO-affect N? Thus, Edge-SO-affects $= \{\langle N, e\rangle \mid e$ SO-affects $N\}$.
2. Edge-SO-optimization: Given an NFG N, an edge e of N, and a threshold $\kappa \geq 0$, is there a value $x \geq 0$, such that $cost_{SO}(N[e \leftarrow x]) \leq \kappa$? Thus, Edge-SO-optimization $= \{\langle N, e, \kappa\rangle \mid$ there exists $x \geq 0$ such that $cost_{SO}(N[e \leftarrow x]) \leq \kappa\}$.
3. SO-optimization: Given an NFG N and a threshold $\kappa \geq 0$, is there an edge e of N and a value $x \geq 0$, such that $cost_{SO}(N[e \leftarrow x]) \leq \kappa$? Thus, SO-optimization$= \{\langle N, \kappa\rangle \mid$ there exist e and $x \geq 0$ such that $cost_{SO}(N[e \leftarrow x]) \leq \kappa\}$.
4. SO-control: Given an NFG N and a threshold $\kappa \geq 0$, is there an edge e of N and a value $x \geq 0$, such that $cost_{SO}(N[e \leftarrow x]) = \kappa$? Thus, SO-control$= \{\langle N, \kappa\rangle \mid$ there exist e and $x \geq 0$ such that $cost_{SO}(N[e \leftarrow x]) = \kappa\}$.

Analyzing the complexity of the problems, we assume that the costs of an NFG are given in binary. As we shall note below, this affects the complexity of the problems. In addition to the classes NP and co-NP, we are going to refer to the class $\Delta_2^P = P^{NP}$ (Θ_2^P), of decision problems that can be decided by a polynomial-time deterministic Turing machine that has access to polynomially many (logarithmically many, respectively) queries to an oracle to an NP-complete problem, and the class DP, of decision problems that are the intersection of an NP and a co-NP problem. That is, a decision problem $\mathcal{L}$ is in DP if there are decision problems L_1, L_2 such that $L_1 \in$ NP, $L_2 \in$ co-NP and $\mathcal{L} = L_1 \cap L_2$.

▶ **Theorem 8.** *The Edge-SO-affects problem is Δ_2^P-complete, and is Θ_2^P complete when costs are given in unary.*

Proof. We start with membership in Δ_2^P. Given an NFG N and an edge e in N, a deterministic Turing machine can use an oracle to SO-cost, calculate $cost_{SO}(N[e \leftarrow 0])$ and $cost_{SO}(N[e \leftarrow \infty_N])$ and compare them. Since the maximal cost of a profile is $\sum_{e \in E} c(e)$, and $cost_{SO}$ is the sum of costs of a subset of edges, rather than an arbitrary number in $\mathbb{R}$, the Turing machine can proceed by a binary search and thus the number of oracle calls is logarithmic in $\sum_{e \in E} c(e)$. When costs are given in binary, $\sum_{e \in E} c(e)$ is exponential in input, hence there are polynomially-many oracle calls. Thus, Edge-SO-affects$\in \Delta_2^P$. However, when costs are given in unary, $\sum_{e \in E} c(e)$ is polynomial in input, hence there are logarithmically-many oracle calls. Thus, Edge-SO-affects$\in \Theta_2^P$.

In the full version, we prove that the problem is Δ_2^P-hard by a reduction from maximum-satisfying-assignment, namely the problem of deciding, given a 3CNF formula φ if the lexicographically maximal assignment that satisfies φ has LSB that equals 1. It was shown by [26] that maximum-satisfying-assignment is Δ_2^P-complete. Essentially, given φ, we construct an NFG N such that profiles corresponds to assignments, and the cost of a profile decreases with lexicographically greater satisfying assignments. The edge e participates in profiles that correspond to assignments in which the LSB is 1, and is minimal only when the maximal lexicographic assignment has LSB 1. Consequently, $\langle N, e\rangle \in$ Edge-SO-affects iff $\varphi \in$ maximum-satisfying-assignment.

In the full version, we prove that when costs are given in unary, the problem is Θ_2^P-hard. The proof is by a reduction from VC-compare, namely the problem of deciding, given two undirected graphs $G_1 = \langle V_1, E_1\rangle$ and $G_2 = \langle V_2, E_2\rangle$, whether the size of a minimal vertex cover of G_1 is less than or equal to the size of a minimal vertex cover of G_2. Essentially, given G_1 and G_2, we construct an NFG N that subsumes both graphs and the objectives of the players are defined so that profiles correspond to choosing a vertex cover in one of the graphs. The edge e participates in profiles in which the players choose to proceed with a cover in G_1, which happens only when the size of a minimal vertex cover of G_1 is less than or equal to the size of a minimal vertex cover of G_2. Consequently, $\langle N, e\rangle \in$ Edge-SO-affects iff $\langle G_1, G_2\rangle \in$ VC-compare. ◀

We continue to the optimization problems. The proof is easy and can be found in the full version. In particular, the lower bounds are by a reduction from the SO-cost problem.

▶ **Theorem 9.** *The Edge-SO-optimization and SO-optimization problems are NP-complete.*

For the upper-bound of the SO-control problem, we first need the following lemma.

▶ **Lemma 10.** *Let N be an NFG and let $\kappa \geq 0$ be a threshold. If there are (not necessarily distinct) edges e_1 and e_2 of N such that $cost_{SO}(N[e_1 \leftarrow 0]) \geq \kappa$ and $cost_{SO}(N[e_2 \leftarrow \infty]) \leq \kappa$, then there is an edge e of N and a value $x \geq 0$ such that $cost_{SO}(N[e \leftarrow x]) = \kappa$.*

Proof. Assume towards contradiction that for all edges e of N and value $x \geq 0$, it holds that $cost_{SO}(N[e \leftarrow x]) \neq \kappa$. In particular, this means that $cost_{SO}(N[e_1 \leftarrow 0]) > \kappa$ and $cost_{SO}(N[e_2 \leftarrow \infty]) < \kappa$. Hence, by monotonicity of $cost^e_{SO}(N)$, we get that $cost_{SO}(N) = cost_{SO}(N[e_2 \leftarrow c(e_2)]) \leq cost_{SO}(N[e_2 \leftarrow \infty]) < \kappa < cost_{SO}(N[e_1 \leftarrow 0]) \leq cost_{SO}(N[e_1 \leftarrow c(e_1)]) = cost_{SO}(N)$. ◀

▶ **Theorem 11.** *The SO-control problem is DP-complete.*

Proof. We start with membership. Let $L_1 = \{\langle N, \kappa\rangle \mid$ there exist an edge e and $x \geq 0$ such that $cost_{SO}(N[e \leftarrow x]) \leq \kappa\}$ and $L_2 = \{\langle N, \kappa\rangle \mid$ there exist an edge e and $x \geq 0$ such that $cost_{SO}(N[e \leftarrow x]) \geq \kappa\}$. Note that L_1 is SO-optimization and is therefore in NP. We show that L_2 is in co-NP. The complement of L_2 is $L_2^c = \{\langle N, \kappa\rangle \mid$ for all edges e and $x \geq 0$ we have $cost_{SO}(N[e \leftarrow x] < \kappa)\}$. A witness for membership in L_2^c is a set S of $|E| = m$ profiles, one for each edge, satisfying $cost(N[e \leftarrow \infty], P_e) < \kappa$ for each $P_e \in S$. The witness is polynomial since we only require m profiles. By monotonicity, it holds that if such a profile P_e exists for an edge e, then for every $x \geq 0$, we have that $cost_{SO}(N[e \leftarrow x]) \leq cost(N[e \leftarrow x], P_e) \leq cost(N[e \leftarrow \infty], P_e) < \kappa$. If this holds for every edge, then $\langle N, \kappa\rangle \in L_2^c$. In the other direction, if there is an edge e such that for every profile P it holds that $cost(N[e \leftarrow \infty], P) \geq \kappa$, then $cost_{SO}(N[e \leftarrow \infty]) \geq \kappa$, and therefore $\langle N, \kappa\rangle \notin L_2^c$. Therefore, L_2^c is in NP, hence L_2 is in co-NP. We show that $L_1 \cap L_2 =$SO-control.

For the first direction, let $\langle N, \kappa\rangle \in$ SO-control. Therefore, there is an edge $e \in E$ and a value $x \geq 0$ such that $cost_{SO}(N[e \leftarrow x]) = \kappa$. In particular, we have that $cost_{SO}(N[e \leftarrow x]) \leq \kappa$, therefore $\langle N \kappa\rangle \in L_1$. Furthermore, $cost_{SO}(N[e \leftarrow x]) \geq \kappa$, therefore $\langle N, \kappa\rangle \in L_2$. Hence, $\langle N, \kappa\rangle \in L_1 \cap L_2$.

For the other direction, let $\langle N, \kappa \rangle \in L_1 \cap L_2$. Since $\langle N, \kappa \rangle \in L_1$, there is $e_1 \in E$ and $x_1 \geq 0$ such that $cost_{SO}(N[e_1 \leftarrow x_1]) \leq \kappa$. If $cost_{SO}(N[e_1 \leftarrow \infty]) \geq \kappa$, then by continuity and the intermediate value theorem, there is $x \geq 0$ such that $cost_{SO}(N[e_1 \leftarrow x]) = \kappa$, hence $\langle N, \kappa \rangle \in$ SO-control. If $cost_{SO}(N[e_1 \leftarrow \infty]) < \kappa$, we use the fact that $\langle N, \kappa \rangle \in L_2$. Hence, there is $e_2 \in E$ and $x_2 \geq 0$ such that $cost_{SO}(N[e_2 \leftarrow x_2]) \geq \kappa$. If $cost_{SO}(N[e_2 \leftarrow 0]) \leq \kappa$, then again by continuity and the intermediate value theorem, there is $x \geq 0$ such that $cost_{SO}(N[e_2 \leftarrow x]) = \kappa$. If $cost_{SO}(N[e_2 \leftarrow 0]) > \kappa$, then since $cost_{SO}(N[e_1 \leftarrow \infty]) < \kappa$ by Lemma 10, there is an edge $e \in E$ and a value $x \geq 0$ such that $cost_{SO}(N[e \leftarrow x]) = \kappa$, and therefore $\langle N, \kappa \rangle \in$ SO-control.

We turn to prove that the problem is DP-hard. We reduce SAT-UNSAT to SO-control. SAT-UNSAT is the problem of deciding, given two 3CNF formulas φ_1 and φ_2, whether φ_1 is satisfiable and φ_2 is not satisfiable. That is, $\langle \varphi_1, \varphi_2 \rangle \in$ SAT-UNSAT iff there exists an assignment f_1 to the variables of φ_1 such that f_1 satisfies φ_1, and for all assignments f_2 to the variables of φ_2, it holds that f_2 does not satisfy φ_2. It was shown in [35] that SAT-UNSAT is DP-complete.

We propose the following reduction. For each formula φ_i, with $i \in \{1, 2\}$, we add a fresh variable z_i. We first construct a new formula φ'_i in the following way. For each clause, we disjunct the clause with z_i. We also conjunct the entire formula with $\neg z_i$. Note that if φ_i is satisfied by an assignment f_i, then φ'_i is satisfied by the assignment that agrees with f_i on all the variables in φ_i, and has $z_i = \boldsymbol{false}$. Furthermore, if φ_i is unsatisfiable, then φ'_i is unsatisfiable. Indeed, an assignment that satisfies φ'_i must have $z_i = \boldsymbol{false}$, implying that all other clauses are satisfied by an assignment that satisfies φ_i as well. Next, we construct an NFG $N_i = \langle k_i, V_i, E_i, c_i, \gamma_i \rangle$, for $i \in \{1, 2\}$, as follows (see Figure 2).

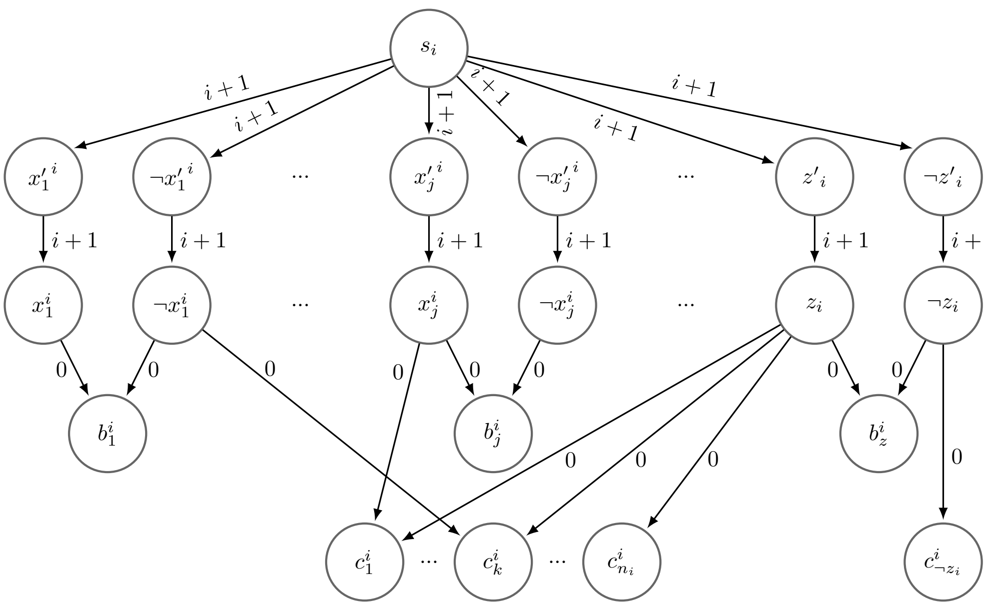

Figure 2 The NFG N_i; each edge denotes a set of two parallel edges with the same cost.

Let n_i be the number of variables in φ_i, and let m_i be the number of clauses in φ_i. Thus, the number of variables in φ'_i is $n_i + 1$, and the number of clauses in φ'_i is $m_i + 1$. We define $V_i = \bigcup_{1 \leq j \leq n_i+1} \{x^i_j, \neg x^i_j, {x'_j}^i, \neg {x'_j}^i, b^i_j\} \bigcup_{1 \leq k \leq m_i+1} \{c^i_k\} \cup \{s_i\}$. That is, for each variable x^i_j of φ'_i, we have in V_i two vertices for the variable x^i_j, denoted $x^i_j, {x'_j}^i$, two vertices for its negation $\neg x^i_j$, denoted $\neg x^i_j, \neg {x'_j}^i$, and another vertex, denoted b^i_j. We also have a vertex for each clause, and a source vertex. The edges and costs are as follows. There are two parallel edges, each with cost $i+1$, from s_i to both ${x'}_j^i, \neg {x'}_j^i$ for every variable x_j^i of φ'_i. There are two parallel edges, each with cost $i+1$, from ${x'}_j^i$ to x_j^i and from $\neg {x'}_j^i$ to $\neg x_j^i$ for every variable x_j^i of φ'_i. There are two parallel edges, each with cost 0 from both $x_j^i, \neg x_j^i$ to b_j^i. Finally, for every clause c_k^i, there are two parallel edges, each with cost 0, from every literal appearing in c_k^i to the vertex c_k^i. Note that, in particular, this means that there are two parallel edges with cost 0 from z_i to all clauses except the clause $\neg z_i$. Finally, we have $k_i = n_i + 1 + m_i + 1$ players. The first $n_i + 1$ players are clause players, and the objective of Player $1 \leq k \leq n_i + 1$ is $\langle s_i, c_k^i \rangle$. The rest are variable players, and the objective of Player $n_i + 2 \leq j \leq n_i + m_i + 2$ is $\langle s_i, b_j^i \rangle$. To complete the construction, we fix $N = \langle k_1 + k_2, V_1 \cup V_2, E_1 \cup E_2, c_1 \cup c_2, \gamma_1 \cup \gamma_2 \rangle$ and $\kappa = 4n_1 + 6n_2 + 16$.

Note that since N_1 and N_2 are disjoint, it holds that $cost_{SO}(N) = cost_{SO}(N_1) + cost_{SO}(N_2)$. We argue that if φ_i, for $i \in [1,2]$, is satisfiable, then $cost_{SO}(N_i) = 2(i+1) \cdot (n_i+1)$, and otherwise $cost_{SO}(N_i) = 2(i+1) \cdot (n_i+2)$. Thus, N has a distinct SO-cost to every combination of {SAT, UNSAT} $\times$ {SAT, UNSAT}, which enables us to point to a threshold κ such that $\langle \varphi_1, \varphi_2 \rangle \in$ SAT-UNSAT iff $\langle N, \kappa \rangle \in$ SO-control. Details can be found in the full version. ◀

4 Affecting the Best Nash Equilibrium

In this section we study the sensitivity of the best NE to cost mutations. As we shall see, while the setting is less clean than in the SO case, we are able to obtain the same complexity bounds for analogous decision problems.

4.1 Monotonicity and continuity of the bNE

▶ **Theorem 12** ($cost_{bNE}$ is not monotone). *There is an NFG N and an edge e of N, such that the function $cost^e_{bNE}(N)$ is not monotone.*

Proof. Consider the NFG N appearing in Figure 3. The game is played between two players, with objectives $\langle s, t_1 \rangle$ and $\langle s, t_2 \rangle$. Let $e = \langle s, t_2 \rangle$. Table 5 describes the costs of the players in the possible four profiles of $N[e \leftarrow x]$. When $x \in [0,1)$, the only NE is $\langle \pi_1^2, \pi_2^1 \rangle$, with cost $x+2$. When $x > 1$, the only NE is $\langle \pi_1^2, \pi_2^2 \rangle$, with cost 2. So, for all $x \in (0,1)$, we have that $cost_{bNE}(N[e \leftarrow x]) = 2 + x > 2 = cost_{bNE}(N[e \leftarrow 1])$, and thus $cost^e_{bNE}(N)$ is not monotone. ◀

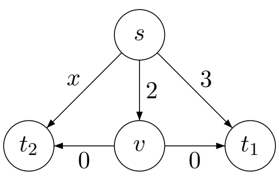

Figure 3 The NFG N.

Table 5 Players' costs in N.

Player 2 / Player 1	π_2^1 $s \to t_2$	π_2^2 $s \to v \to t_2$
π_1^1 $s \to t_1$	x 3	2 3
π_1^2 $s \to v \to t_1$	x 2	1 1

▶ **Theorem 13** ($cost_{bNE}$ is not continuous). *There is an NFG N and an edge e of N, such that the function $cost^e_{bNE}(N)$ is not continuous.*

Proof. We use the same NFG N and edge e as in the proof of Theorem 12. It is easy to see that $cost^e_{bNE}(N)$ is not continuous at 1. ◀

While $cost_{bNE}$ is neither monotonous nor continuous, we now show that it is composed of finitely many linear segments. We say that a function $f : \mathbb{R}^+ \to \mathbb{R}^+$ is *composed of linear segments* if there is a segmentation $0 = x_0 < x_1 < ... < x_n < x_{n+1} = \infty$ of $\mathbb{R}^+$, for some $n \geq 0$, such that for every $0 \leq i \leq n$ there is a linear function $f_i : \mathbb{R} \to \mathbb{R}$ such that for all $x \in [x_i, x_{i+1}]$ it holds that $f(x) = f_i(x)$. We call $x_0, x_1, ..., x_{n+1}$ the *edge points* of f. Given an NFG N, a profile P, and an edge e, the cost of P is a linear function with respect to the cost of e. Indeed, $cost(N, P) = \sum_{e' \in P \setminus \{e\}} c(e') + \mathbb{1}_{P,e} c(e)$, where $\mathbb{1}_{P,e} \in \{0, 1\}$ is an indicator of e being used in P. In particular, when $\mathbb{1}_{P,e} = 0$, then $cost(N, P)$ is a constant function.

▶ **Lemma 14.** *Given an NFG N, an edge e, and a profile P, the range of values x such that P is an NE in $N[e \leftarrow x]$ is a single (possibly empty) segment.*

Proof. By definition, a profile P is an NE if for every i and for every profile P' obtained from P by a deviation π'_i of Player i that $cost_{N,P}(\pi_i) \leq cost_{N,P'}(\pi'_i)$. Hence, P is an NE in $N[e \leftarrow x]$ in values x for which the set of constraints of the form $cost_{N,P}(\pi_i) \leq cost_{N,P'}(\pi'_i)$ holds. As each constraint is a linear inequality in a single variable (that is, x), the solution set is a single (perhaps empty) segment. ◀

We denote by $bumps(P)$ the set of edge points of the segment along which P is an NE in $N[e \leftarrow x]$. That is, $bumps(P) = \{a, b\}$ if P is an NE in $N[e \leftarrow x]$ for exactly all $a \leq x \leq b$. By Lemma 14, $bumps(P)$ contains at most two points. We further denote by $Bumps(N, e) = \bigcup_P bumps(P)$. Since the number of strategies per player and the number of players are finite, the number of profiles is finite as well. Hence, since $|bumps(P)| \leq 2$ for every profile P, we get that $Bumps(N, e)$ is finite.

Consider two profiles $P_1 \neq P_2$ in N. For an edge e, we say that a value $x \geq 0$ is an *intersection point* for e, P_1, and P_2, if $cost(N[e \leftarrow x], P_1) = cost(N[e \leftarrow x], P_2)$. Note that since $cost(N[e \leftarrow x], P)$ is linear for every profile P, there is at most one intersection point for every edge and two profiles. Let $Ints(N, e)$ be the set of all intersection points for e and pairs of profiles in N. Since the number of different profiles is finite, so is $Ints(N, e)$.

▶ **Theorem 15.** *Consider an NFG N and an edge e in N. Then, $cost_{bNE}(N[e \leftarrow x])$ is composed of finitely many linear segments, and is monotonically increasing within each segment.*

Proof. Recall that $cost^e_{bNE}(N)(x) = cost_{bNE}(N[e \leftarrow x]) = \min_{P \in bNE(N[e \leftarrow x])} cost(N[e \leftarrow x], P) = \min_{P \in bNE(N[e \leftarrow x])} \sum_{e' \in P \setminus \{e\}} c(e') + \mathbb{1}_{P,e} x$. Hence, $cost_{bNE}(N[e \leftarrow x])$ is composed of linear segments. The set of edge points refines $bumps(N, e) \cup Ints(N, e)$, and since it is finite, so are the number of segments. Furthermore, as $cost(N[e \leftarrow x], P)$ is monotonically increasing for every P, we get that $cost_{bNE}(N[e \leftarrow x])$ is monotonically increasing within each segment. ◀

Figure 4 below contains plots[2] of the function $cost_{bNE}(N[e \leftarrow x])$. The left plot describes $cost_{bNE}(N[e \leftarrow x])$ where N is the NFG from Example 1 and $e = \langle s, u \rangle$. To its right, we describe a three-player NFG N and the plot of $cost_{bNE}(N[e \leftarrow x])$ with $e = \langle s, v_2 \rangle$.

[2] The plots were generated by a simple Python program that gets as input an NFG by means of a NetworkX weighted directed graph, and naively follows the segmentation from Theorem 15.

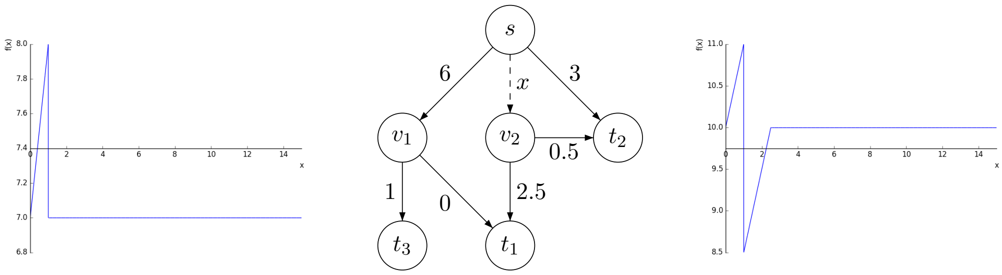

Figure 4 Plots for $cost_{bNE}(N[e \leftarrow x])$.

4.2 Decision problems

The bNE-cost decision problem is the problem of deciding, given an NFG N and a threshold $\kappa \geq 0$, whether $cost_{bNE}(N) \leq \kappa$. The bNE-cost problem is NP-complete [4]. In this section we study the following related decision problems.

1. Edge-bNE-affects: Given an NFG N and an edge e of N, does e bNE-affect N? Thus, Edge-bNE-affects $= \{\langle N, e\rangle \mid e \text{ bNE-affects } N\}$.
2. Edge-bNE-optimization: Given an NFG N, an edge e of N, and a threshold $\kappa \geq 0$, is there a value $x \geq 0$, such that $cost_{bNE}(N[e \leftarrow x]) \leq \kappa$? Thus, Edge-bNE-optimization $= \{\langle N, e, \kappa\rangle \mid \text{there exists } x \geq 0 \text{ such that } cost_{bNE}(N[e \leftarrow x]) \leq \kappa\}$.
3. bNE-optimization: Given an NFG N and a threshold $\kappa \geq 0$, is there an edge e of N and a value $x \geq 0$, such that $cost_{bNE}(N[e \leftarrow x]) \leq \kappa$? Thus, bNE-optimization$= \{\langle N, \kappa\rangle \mid \text{there exist } e \text{ and } x \geq 0 \text{ such that } cost_{bNE}(N[e \leftarrow x]) \leq \kappa\}$.

Before we turn to analyze the complexity of the problems, let us illustrate the non-intuitive behavior of $cost_{bNE}$. Consider the NFG N appearing in Figure 5, and let $e = \langle s, v_2\rangle$. As can be seen in Table 6, the profile $\langle \pi_1^3, \pi_2^3\rangle$ is an NE with cost 10 independent of the value of x. Then, when $0 \leq x \leq \frac{1}{2}$, the profile $\langle \pi_1^2, \pi_2^1\rangle$ is an NE with cost $10.5 + x$, and when $x \geq \frac{1}{2}$, the profile $\langle \pi_1^1, \pi_2^1\rangle$ is an NE with cost 9. Accordingly, $cost_{bNE}(N[e \leftarrow x])$ is 10 when $0 \leq x < \frac{1}{2}$, and is 9 when $x \geq \frac{1}{2}$. Though observations of the non-intuitive behavior of network exists in literature (e.g., Braess' Paradox [12]), it is common that added/removed edges participate in equilibria profiles either before or after changing the network. In this example, however, the edge e, which bNE-affects N, does not participate in any bNE profile! Thus, $cost_{bNE}$ is fixed in the two segments $[0, \frac{1}{2})$ and $[\frac{1}{2}, \infty]$, yet still e bNE affects N.

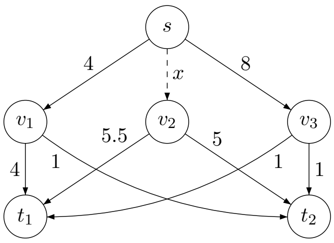

Figure 5 The NFG N.

Table 6 Players' costs in N.

Player 2 / Player 1	π_2^1 s, v_1, t_2	π_2^2 s, v_2, t_2	π_2^3 s, v_3, t_2
π_1^1 s, v_1, t_1	3 / 6	$5+x$ / 8	9 / 8
π_1^2 s, v_2, t_1	5 / $5.5+x$	$5+\frac{x}{2}$ / $5.5+\frac{x}{2}$	9 / $5.5+x$
π_1^3 s, v_3, t_1	5 / 9	$5+x$ / 9	5 / 5

▶ **Lemma 16.** *Let N be an NFG, and let e be an edge in N. If there is an NE profile P such that $e \notin P$, then for all $x \geq c(e)$, we have that P is an NE in $N[e \leftarrow x]$.*

Proof. Assume towards contradiction that there is $x > c(e)$ such that P is not an NE. Then, there is a player i with strategy π_i in P that has an incentive to unilaterally deviate to another strategy π'_i. Denote by P' the deviation profile resulting from i's deviation. Since P is an NE in N, we have that $cost_{N,P}(\pi_i) \leq cost_{N,P'}(\pi'_i)$. Since $e \notin P$, we have that $cost_{N[e\leftarrow x],P}(\pi_i) = cost_{N,P}(\pi_i)$. Since $x > c(e)$ we have that $cost_{N,P'}(\pi'_i) \leq cost_{N[e\leftarrow x],P'}(\pi'_i)$. Therefore $cost_{N[e\leftarrow x],P}(\pi_i) \leq cost_{N[e\leftarrow x],P'}(\pi'_i)$, in contradiction to the fact that Player i has an incentive to deviate. ◀

Lemma 16, together with the segmentation of $bNE(N[e \leftarrow x])$, is used for proving the following characterization of an edge that does not bNE-affect N. The proof is based on a careful consideration of all cases and can be found in the full version.

▶ **Theorem 17.** *Let N be an NFG. An edge e in N does not bNE-affect N iff there is a profile $P \in bNE(N[e \leftarrow 0])$ such that $e \notin P$ and for all $x \geq 0$ it holds that $cost_{bNE}(N[e \leftarrow x]) \geq cost_{bNE}(N[e \leftarrow 0])$.*

▶ **Theorem 18.** *The Edge-bNE-affects problem is Δ_2^P-complete, and is Θ_2^P-complete when costs are given in unary.*

Proof. We start with membership. First, note that given an NFG N, and edge e of N, and a value $\kappa \geq 0$, we can decide in NP whether there is a profile P such that $e \notin P$ and $cost(N, P) = \kappa$.

Let $OPT_0 = cost_{bNE}(N[e \leftarrow 0])$. As argued in the membership claim for Theorem 8, we can find OPT_0 using polynomially-many queries to an NP oracle when costs are given in binary, and using logarithmically-many queries when costs are given in unary. Then, using a single query to Edge-bNE-optimization (with modification to strictly smaller) with input N, e, and OPT_0, we can decide if there is a value $x \geq 0$ such that $cost_{bNE}(N[e \leftarrow x]) < OPT_0$. If so, then e affects N. Otherwise, use a single query to ask if there is a profile P such that $e \notin P$ and $cost(N[e \leftarrow 0], P) = OPT_0$. By Theorem 17, we have that e bNE-affects N iff the answer is no.

The hardness results for Δ_2^P and Θ_2^P can be found in the full version. In both cases we use the same reduction as in the hardness results for Theorem 8. In the case of Δ_2^P we make a slight variation. Then we show that the profiles described for the SO is a superset of the bNE profiles. ◀

Finally, for the optimization problems, the analysis is similar to the one in Theorem 9, except that we also have to argue that the witness value x is polynomial in input. The details can be found in the full version.

▶ **Theorem 19.** *The edge-bNE-optimization and bNE-optimization problems are NP-complete.*

▶ Remark 20 (On the PoS and the worst NE). Recall that $PoS(N) = \frac{cost_{bNE}(N)}{cost_{SO}(N)}$. If an edge e bNE-affects N, it does not necessarily imply that e PoS-affects N. Indeed, e may participate also in the SO. Nevertheless, the NFG N used in the proofs of Theorems 12 and 13 demonstrates that PoS is neither monotone nor continuous. To see this, note that for all $x \geq 0$, we have that $cost_{SO}(N[e \leftarrow x]) = 2$, we get that for $x \in [0, 1)$, we have that $PoS(N[e \leftarrow x]) = 1 + \frac{x}{2}$, and for $x \geq 1$, we have that $PoS(N[e \leftarrow x]) = 1$.

As for the worst NE, since the NFG N used in the proofs of Theorems 12 and 13 is such that $N[e \leftarrow x]$ has a single NE for all values of x, the considerations about the best and worst NE coincide, and thus N demonstrate that $cost_{wNE}$ is neither monotone nor continuous.

5 Discussion and Future Work

We studied the effect of mutations applied to the cost of edges in network formation games. Our results about monotonicity and continuity of the SO and NE are aligned with similar folk results in similar settings in game theory. We are, however, the first to introduce a formal framework to study these phenomena, and to provide a complexity analysis of the decision problems they induce. We also point to new surprising effects of the mutations.

The mutations we study for NFGs are of a restricted type: an unbounded change in the cost of a single resource in the game. As has been the case in coverage and vacuity in formal verification, richer types of mutations reflect practical bounds on the possible mutations. For example, it would be interesting to study how one can control the bNE by a budget-restricted mutation of several edges. Also, while our definition of affect is Boolean, namely an edge SO-, bNE-, or wNE-affects a network or it does not, it is interesting to examine a quantitative approach, where we care how much an edge affects these measures. Finally, while our optimization problems care about an upper bound to the costs of the SO and bNE, in some applications it is interesting to control these values by both an upper and lower bound. We leave the richer setting and variants for future research.

Both game theory and formal verification aim at reasoning about behaviors of interacting entities, yet consider different aspects of the interaction. We view this work as another chain in an exciting transfer of concepts and ideas between the two areas [28]. In the context of game theory, this includes an extension of NFGs to objectives that are richer than reachability [9], to a timed setting [6], and to a setting where the strategies of the players are dynamic [7]. Beyond richer settings, it is shown in [30, 5] how ideas used in formal verification for abstraction and symbolic presentation of huge systems can be used for reasoning about NFGs. In the other direction, concepts from game theory are used in the formalization of strategic behaviors in formal verification (e.g., rational verification and synthesis [22, 38]). In the more economic view, cost-sharing mechanisms from NFGs are used in [8] in order to augment the problem of synthesis from component libraries by cost considerations.

Our contribution here started with the transfer of concepts from formal verification to game theory, yet our results suggest new research directions in coverage and vacuity in formal verification, and logic in general. Studies of coverage and vacuity so far concern Boolean specification formalisms [27]. In contrast, the objectives of the players in typical game-theoretic settings, in particular NFGs, are quantitative. Recently, there is growing interest in *multi-valued* specification formalisms, which specify the *quality* of systems, and not only their correctness [2]. Moreover, the systems we reason about may be multi-valued too. For the multi-valued setting, we need to develop a theory of quantified multi-valued propositions. In particular, the segmentation of values in $\mathbb{R}^+$ we perform for bNE, is analogous to a segmentation of $[0,1]$ – the domain of values of atomic propositions and sub-formulas in typical multi-valued formalisms. Indeed, while mutations of sub-formulas that appear in a positive or negative polarity behave monotonically, sub-formulas with a mixed polarity may induce a non-trivial segmentation. Moreover, as has been the case with $bumps(P)$ in the bNE segmentation, the edge points of the segments may not be constants that appear in the formula. For example, when sub-formulas and atomic propositions take values in $[0,1]$, then the maximal satisfaction value of the formula $p \wedge (\neg p)$ is when the satisfaction value of p is $\frac{1}{2}$.

Furthermore, the need to reason formally about multi-agent systems has led to a development of specification formalisms that enable reasoning about on-going strategic behaviors [3, 13, 32, 11]. Essentially, these formalisms, most notably ATL, ATL*, and Strategy Logic (SL), include quantification of strategies of the different agents and of the computations they may force the system into, making it possible to specify concepts like SO and NE.

While coverage and vacuity are traditionally viewed as sanity checks in model checking, in the context of SL specifications, they can also be used for revealing properties of games and strategic behaviors. Out work demonstrates how SL formulas that specify concepts like SO and NE explain properties like monotonicity. Indeed, non-monotonicity of the bNE corresponds to the mixed polarity of the objectives in the SL formula that describes an NE: a negative occurrence (left-hand side of an implication) when we refer to a deviation and a positive one (right-hand side of that implication) in for the current strategy. In contrast, in the formula for the SO, all occurrences of the objectives are positive, implying monotonicity. Moreover, for a specific given game, reasoning about the effect of mutations can be reduced to checking the coverage of SL formulas that specify properties of the game. Thus, a framework for coverage and vacuity in SL is interesting for both formal verification and game theory.

References

1 S. Albers, S. Elits, E. Even-Dar, Y. Mansour, and L. Roditty. On Nash Equilibria for a Network Creation Game. In *7th ACM-SIAM Symposium on Discrete Algorithms*, 2006.

2 S. Almagor, U. Boker, and O. Kupferman. Formalizing and Reasoning About Quality. *Journal of the ACM*, 63(3), 2016.

3 R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, 2002.

4 E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden. The Price of Stability for Network Design with Fair Cost Allocation. *SIAM J. Comput.*, 38(4):1602–1623, 2008.

5 G. Avni, S. Guha, and O. Kupferman. An Abstraction-Refinement Methodology for Reasoning about Network Games. In *Proc. 26th Int. Joint Conf. on Artificial Intelligence*, pages 70–76, 2017.

6 G. Avni, S. Guha, and O. Kupferman. Timed Network Games. In *42nd Int. Symp. on Mathematical Foundations of Computer Science*, volume 83 of *LIPIcs*, pages 37:1–37:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2017.

7 G. Avni, T.A. Henzinger, and O. Kupferman. Dynamic Resource Allocation Games. In *Algorithmic Game Theory - 9th International Symposium*, volume 9928 of *Lecture Notes in Computer Science*, pages 153–166. Springer, 2016.

8 G. Avni and O. Kupferman. Synthesis from Component Libraries with Costs. In *Proc. 25th Int. Conf. on Concurrency Theory*, volume 8704 of *Lecture Notes in Computer Science*, pages 156–172. Springer, 2014.

9 G. Avni, O. Kupferman, and T. Tamir. Network-formation games with regular objectives. *Information and Computation*, 251:165–178, 2016.

10 I. Beer, S. Ben-David, C. Eisner, and Y. Rodeh. Efficient detection of vacuity in ACTL formulas. *Formal Methods in System Design*, 18(2):141–162, 2001.

11 P. Bouyer-Decitre, O. Kupferman, N. Markey, B. Maubert, A. Murano, and G. Perelli. Reasoning about Quality and Fuzziness of Strategic Behaviours. In *Proc. 28th Int. Joint Conf. on Artificial Intelligence*, 2019.

12 D. Braess. Über ein paradoxon aus der verkehrsplanung. *Unternehmensforschung*, 12(1):258–268, 1968.

13 K. Chatterjee, T. A. Henzinger, and N. Piterman. Strategy logic. *Information and Computation*, 208(6):677–693, 2010.

14 H. Chockler, A. Gurfinkel, and O. Strichman. Beyond vacuity: towards the strongest passing formula. *Formal Methods in System Design*, 43(3):552–571, 2013.

15 H. Chockler, O. Kupferman, R.P. Kurshan, and M.Y. Vardi. A Practical Approach to Coverage in Model Checking. In *Proc. 13th Int. Conf. on Computer Aided Verification*, volume 2102 of *Lecture Notes in Computer Science*, pages 66–78. Springer, 2001.

16 H. Chockler, O. Kupferman, and M.Y. Vardi. Coverage Metrics for Temporal Logic Model Checking. In *Proc. 7th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems*, number 2031 in Lecture Notes in Computer Science, pages 528–542. Springer, 2001.

17 H. Chockler, O. Kupferman, and M.Y. Vardi. Coverage Metrics for Formal Verification. *Software Tools for Technology Transfer*, 8(4-5):373–386, 2006.

18 E.M. Clarke, O. Grumberg, D. Kroening, D. Peled, and H. Veith. *Model Checking, Second Edition.* MIT Press, 2018.

19 R. Cole, Y. Dodis, and T. Roughgarden. How much can taxes help selfish routing? *J. Comput. Syst. Sci.*, 72(3):444–467, 2006.

20 J. Correa, A. Schulz, and N. Stier Moses. Selfish Routing in Capacitated Networks. *Math. Oper. Res.*, 29(4):961–976, 2004.

21 A. Fabrikant, A. Luthra, E. Maneva, C. Papadimitriou, and S. Shenker. On a Network Creation Game. In *ACM Symposium on Principles of Distributed Computing*, 2003.

22 D. Fisman, O. Kupferman, and Y. Lustig. Rational Synthesis. In *Proc. 16th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems*, volume 6015 of *Lecture Notes in Computer Science*, pages 190–204. Springer, 2010.

23 J.D. Hartline and T. Roughgarden. Optimal mechanism design and money burning. In *Proc. 40th ACM Symp. on Theory of Computing*, pages 75–84, 2008.

24 Y. Hoskote, T. Kam, P.-H Ho, and X. Zhao. Coverage estimation for symbolic model checking. In *Proc. 36th Design Automation Conf.*, pages 300–305, 1999.

25 S. Katz, D. Geist, and O. Grumberg. "Have I written enough properties ?" A method of comparison between specification and implementation. In *Proc. 10th Conf. on Correct Hardware Design and Verification Methods*, volume 1703 of *Lecture Notes in Computer Science*, pages 280–297. Springer, 1999.

26 M.W. Krentel. The complexity of optimization problems. *Journal of Computer and Systems Science*, 36:490–509, 1988.

27 O. Kupferman. Sanity Checks in Formal Verification. In *Proc. 17th Int. Conf. on Concurrency Theory*, volume 4137 of *Lecture Notes in Computer Science*, pages 37–51. Springer, 2006.

28 O. Kupferman. Examining classical graph-theory problems from the viewpoint of formal-verification methods. In *Proc. 49th ACM Symp. on Theory of Computing*, page 6, 2017.

29 O. Kupferman, W. Li, and S.A. Seshia. A Theory of Mutations with Applications to Vacuity, Coverage, and Fault Tolerance. In *Proc. 8th Int. Conf. on Formal Methods in Computer-Aided Design*, pages 1–9, 2008.

30 O. Kupferman and T. Tamir. Hierarchical Network Formation Games. In *Proc. 22nd Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems*, volume 10205 of *Lecture Notes in Computer Science*, pages 229–246. Springer, 2017.

31 O. Kupferman and M.Y. Vardi. Vacuity detection in temporal model checking. *Software Tools for Technology Transfer*, 4(2):224–233, 2003.

32 F. Mogavero, A. Murano, G. Perelli, and M. Y. Vardi. Reasoning About Strategies: On the Model-Checking Problem. *ACM Transactions on Computational Logic*, 15(4):34:1–34:47, 2014.

33 O. Padon, N. Immerman, A. Karbyshev, O. Lahav, M. Sagiv, and S. Shoham. Decentralizing SDN Policies. In *Proc. 42nd ACM Symp. on Principles of Programming Languages*, pages 663–676, 2015.

34 A. Panda, K. Argyraki, M. Sagiv, M. Schapira, and S. Shenker. New Directions for Network Verification. In *1st Summit on Advances in Programming Languages*, volume 32 of *LIPIcs*, pages 209–220, 2015.

35 C.H. Papadimitriou and M. Yannakakis. The complexity of facets (and some facets of complexity). In *Proc. 14th ACM Symp. on Theory of Computing*, pages 255–260, 1982.

36 F. Souza and L.C. Rego. Mixed equilibrium in 2 x 2 normal form games: when burning money is rational. *Pesquisa Operacional*, 36:81–99, April 2016.

37 E. Tardos and T. Wexler. *Algorithmic Game Theory*. Cambridge University Press, 2007. Chapter 19: Network Formation Games and the Potential Function Method.

38 M. Wooldridge, J. Gutierrez, P. Harrenstein, E. Marchioni, G. Perelli, and A. Toumi. Rational Verification: From Model Checking to Equilibrium Checking. In *Proc. of 30th National Conf. on Artificial Intelligence*, pages 4184–4190, 2016.