Robust Equilibria in Mean-Payoff Games

Romain Brenguier*

University of Oxford, UK

Abstract We study the problem of finding robust equilibria in multiplayer concurrent games with mean payoff objectives. A (k, t)-robust equilibrium is a strategy profile such that no coalition of size k can improve the payoff of one its member by deviating, and no coalition of size t can decrease the payoff of other players. While deciding whether there exists such equilibria is undecidable in general, we suggest algorithms for two meaningful restrictions on the complexity of strategies. The first restriction concerns memory. We show that we can reduce the problem of the existence of a memoryless robust equilibrium to a formula in the (existential) theory of reals. The second restriction concerns randomisation. We suggest a general transformation from multiplayer games to two-player games such that pure equilibria in the first game correspond to winning strategies in the second one. Thanks to this transformation, we show that the existence of robust equilibria can be decided in polynomial space, and that the decision problem is PSPACE-complete.

1 Introduction

Games are intensively used in computer science to model interactions in computerised systems. Two player antagonistic games have been successfully used for the synthesis of reactive systems. In this context, the opponent acts as a hostile environment, and winning strategies provide controllers that ensure correctness of the system under any scenario. In order to model complex systems in which several rational entities interact, multiplayer concurrent games come into the picture. Correctness of the strategies can be specified with different solution concepts, which describe formally what is a "good" strategy. In game theory, the fundamental solution concept is Nash equilibrium [15], where no player can benefit from changing its own strategy. The notion of robust equilibria refines Nash equilibria in two ways: 1. a robust equilibrium is *resilient*, i.e. when a "small" coalition of player changes its strategy, it can not improve the payoff of one of its participants; 2. it is *immune*, i.e. when a "small" coalition changes its strategy, it will not lower the payoff of the non-deviating players. The size of what is considered a small coalition is determined by a bound k for resilience and another t for immunity. When a strategy is both k-resilient and t-immune, it is called a (k, t)-robust equilibrium. We also generalise this concept to the notion (k, t, r)-robust equilibrium, where if t players are deviating, the others should not have their payoff decrease by more than r.

* Work supported by ERC Starting Grant inVEST (279499) and EPSRC grant EP/M023656/1.

Example In the design of network protocols, when many users are interacting, coalitions can easily be formed and resilient strategies are necessary to avoid deviation. It is also likely that some clients are faulty and begin to behave unexpectedly, hence the need for immune strategies.

As an example, consider a program for a server that distributes files, of which a part is represented in Figure 1. The functions `listen` and `send_files` will be run in parallel by the server. Some choices in the design of these functions have not been fixed yet and we wish to analyse the robustness of the different alternatives.

This program uses a table `clients` to keep track of the clients which are connected. Notice that the table has fixed size 2, which means that if 3 clients try to connect at the same time, one of them may have its socket overwritten in the table and will have to reconnect later to get the file. We want to know what strategy the clients should use and how robust the protocol will be: can clients exploit the protocol to get their files faster than the normal usage, and how will the performance of the over clients be affected.

We consider different strategies to chose between the possible alternatives in the program of Figure 1. The strategy that chooses alternatives 1 and 3 does not give 1-resilient equilibria even for just two clients: Player 1 can always reconnect just after its socket was closed, so that `clients[0]` points to player 1 once again. In this way, he can deviate from any profile to never have to wait for the file. Since the second player could do the same thing, no profile is 1-resilient (nor 1-immune). For the same reasons, the strategy $2, 3$ does not give 1-resilient equilibria. The strategy $1, 4$ does not give 1-resilient equilibria either, since player 1 can launch a new connection after player 2 to overwrite `clients[1]`.

The strategy $2, 4$ is the one that may give the best solution. We modelled the interaction of this program as a concurrent game for a situation with 2 potential clients in Figure 2. The labels represent the content of the table `clients`: 0 means no connection, 1 means connected with player 1 and 2 connected with player 2; and the instruction that the function `send_files` is executing. Because the game is quite big we represent only the part where player 1 connects before player 2 and the rest of the graph can be deduced by symmetry. The actions of the players are either to wait (action `w`) or to connect (action `ch` or `ct`). Symbol $*$ means any possible action. Note that in the case both player try to connect at the same time we simulate a matching penny game in order to determine which one will be treated first, this is the reason why we have two different possible actions to connect (`ch` for "head" and `ct` for "tail"). Clients have a positive reward when we send them the file they requested, this corresponds for Player i to states labelled with `send(i)`.

If both clients try to connect in the same slot, we use a matching penny game to decide which request was the first to arrive. For a general method to transform a game with continuous time into a concurrent game, see [6, Chapter 6].

Related works and comparison with Nash equilibria and secure equilibria Other solution concepts have been proposed as concepts for synthesis of distributed systems, in particular Nash equilibrium [18,19,4], subgame perfect equilibria [17,10],

```
clients = new socket[2];

void listen() {
  while(true) {
    Socket socket = serverSocket.accept();
    if(clients[0].isConnected())
    //// Two possible alternatives:
    | 1) clients[1] = socket;
    | 2) if(socket.remoteSocketAddress()
          != clients[0].remoteSocketAddress())
    |      clients[1] = socket;
    else
      clients[0] = socket;
  } }

void send_files() {
  while(true) {
    if(clients[0].isConnected()) {
      send(clients[0]);
      clients[0].close();
    }
    //// Two possible alternatives :
    | 3) else if(clients[1].isConnected()) {
    | 4) if(clients[1].isConnected()) {
      send(clients[1]);
      clients[1].close();
} } }
```

Figure 1: Example of a server program.

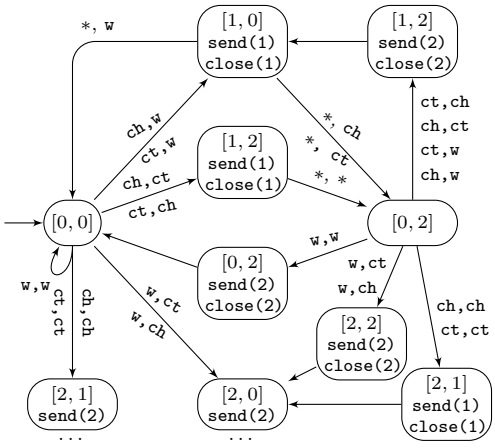

Figure 2: Example of a concurrent game generated from the program of Figure 1.

and secure equilibria [12]. A subgame perfect equilibria is a particular kind of Nash equilibria, where at any point in the game, if we forget the history the players are still playing a Nash equilibrium. In a secure equilibria, we ask that no player can benefit or keep the same reward while reducing the payoff of other players by changing its own strategy. However these concepts present two weaknesses: 1. There is no guarantee when two (or more) users deviate together. It can happen on a network that the same person controls several devices (a laptop and a phone for instance) and can then coordinate there behaviour. In that case, the devices would be considered as different agents and Nash equilibria offers no guarantee. 2. When a deviation occurs, the strategies of the equilibrium can punish the deviating user without any regard for payoffs of the others. This can result in a situation where, because of a faulty device, the protocol is totally blocked. By comparison, finding resilient equilibria with k greater than 1, ensures that clients have no interest in forming coalitions (up to size k), and finding immune equilibria with t greater than 0 ensures that other clients will not suffer from some agents (up to t) behaving differently from what was expected.

Note that the concept of robust equilibria for games with LTL objectives is expressible in logics such as strategy logic [13] or ATL* [2]. However, satisfiability in these logic is difficult: it is 2EXPTIME-complete for ATL* and undecidable for strategy logic in general (2EXPTIME-complete fragments exist [14]). Moreover, these logics cannot express equilibria in quantitative games such as mean-payoff.

Contributions In this paper, we study the problem of finding robust equilibria in multiplayer concurrent games. This problem is undecidable in general (see Section 2.3). In Section 3, we show that if we look for stationary (but randomised) strategies, then the problem can be decided using the theory of reals. We then turn to the case of pure (but memoryful) strategies. In Section 4, we describe a generic transformation from multiplayer games to two-player games. The resulting two-player game is called the *deviator game*. We show that pure equilibria in the original game correspond to winning strategies in the second one. In Section 5, we study quantitative games with mean-payoff objectives. We show that the game obtained by our transformation is equivalent to a multidimensional mean-payoff game. We then show that this can be reduced to a value problem with linear constraints in multidimensional mean-payoff games. We show that this can be solved in polynomial space, by making use of the structure of the deviator game. In Section 7, we prove the matching lower bound which shows the robustness problem is PSPACE-complete. Due to space constraints, most proofs have been omitted from this paper; they can be found in the appendix.

2 Definitions

2.1 Weighted concurrent games

We study concurrent game as defined in [2] with the addition of weights on the edges.

Concurrent games. A *weighted concurrent game* (or simply a *game*) $\mathcal{G}$ is a tuple $\langle \mathsf{Stat}, s_0, \mathsf{Agt}, \mathsf{Act}, \mathsf{Tab}, (w_A)_{A\in\mathsf{Agt}}\rangle$, where: Stat is a finite set of *states* and $s_0 \in \mathsf{Stat}$ is the *initial state*; Agt is a finite set of *players*; Act is a finite set of *actions*; a tuple $(a_A)_{A\in\mathsf{Agt}}$ containing one action a_A for each player A is called a *move*; $\mathsf{Tab} : \mathsf{Stat} \times \mathsf{Act}^{\mathsf{Agt}} \to \mathsf{Stat}$ is the *transition function*, it associates with a given state and a given move, the resulting state; for each player $A \in \mathsf{Agt}$, $w_A \colon \mathsf{Stat} \mapsto \mathbb{Z}$ is a *weight function* which assigns to each agent an integer weight.

In a game $\mathcal{G}$, whenever we arrive at a state s, the players simultaneously select an action. This results in a move a_{Agt}; the next state of the game is then $\mathsf{Tab}(s, a_{\mathsf{Agt}})$. This process starts from s_0 and is repeated to form an infinite sequence of states.

An example of a game is given in Figure 2. It models the interaction of two clients A_1 and A_2 with the program presented in the introduction. The weight functions for this game are given by $w_{A_1} = 1$ in states labelled by `send(1)` and $w_{A_1} = 0$ elsewhere, similarly $w_{A_2} = 1$ in states labelled by `send(2)`.

History and plays. A *history* of the game $\mathcal{G}$ is a finite sequence of states and moves ending with a state, i.e. an word in $(\mathsf{Stat} \cdot \mathsf{Act}^{\mathsf{Agt}})^* \cdot \mathsf{Stat}$. We write h_i the i-th state of h, starting from 0, and $\mathsf{move}_i(h)$ its i-th move, thus $h = h_0 \cdot \mathsf{move}_0(h) \cdot h_1 \cdots \mathsf{move}_{n-1}(h) \cdot h_n$. The length $|h|$ of such a history is $n+1$. We write $\mathtt{last}(h)$ the last state of h, i.e. $h_{|h|-1}$. A *play* ρ is an infinite sequence of

states and moves, i.e. an element of $(\mathsf{Stat} \cdot \mathsf{Act}^{\mathsf{Agt}})^\omega$. We write $\rho_{\leq n}$ for the prefix of ρ of length $n+1$, i.e. the history $\rho_0 \cdot \mathsf{move}_0(\rho) \cdots \mathsf{move}_{n-1}(\rho) \cdot \rho_n$.

The *mean-payoff* of weight w along a play ρ is the average of the weights along the play: $\mathsf{MP}_w(\rho) = \liminf_{n\to\infty} \frac{1}{n} \sum_{0\leq k\leq n} w(\rho_k)$. The *payoff* for agent $A \in \mathsf{Agt}$ of a play ρ is the mean-payoff of the corresponding weight: $\mathsf{payoff}_A(\rho) = \mathsf{MP}_{w_A}(\rho)$. Note that it only depends on the sequence of states, and not on the sequence of moves. The *payoff vector* of the run ρ is the vector $p \in \mathbb{R}^{\mathsf{Agt}}$ such that for all players $A \in \mathsf{Agt}$, $p_A = \mathsf{payoff}_A(\rho)$; we simply write $\mathsf{payoff}(\rho)$ for this vector.

Strategies. Let $\mathcal{G}$ be a game, and $A \in \mathsf{Agt}$. A *strategy* for player A maps histories to probability distributions over actions. Formally, a strategy is a function $\sigma_A \colon (\mathsf{Stat} \cdot \mathsf{Act}^{\mathsf{Agt}})^* \cdot \mathsf{Stat} \to \mathcal{D}(\mathsf{Act})$, where $\mathcal{D}(\mathsf{Act})$ is the set of probability distributions over Act. For an action $a \in \mathsf{Act}$, we write $\sigma_A(a \mid h)$ the probability assigned to a by the distribution $\sigma(h)$. A *coalition* $C \subseteq \mathsf{Agt}$ is a set of players, its size is the number of players it contains and we write it $|C|$. A strategy $\sigma_C = (\sigma_A)_{A\in C}$ for a coalition $C \subseteq \mathsf{Agt}$ is a tuple of strategies, one for each player in C. We write σ_{-C} for a strategy of coalition $\mathsf{Agt} \setminus C$. A *strategy profile* is a strategy for Agt. We will write (σ_{-C}, σ'_C) for the strategy profile σ''_{Agt} such that if $A \in C$ then $\sigma''_A = \sigma'_A$ and otherwise $\sigma''_A = \sigma_A$. We write $\mathsf{Strat}_{\mathcal{G}}(C)$ for the set of strategies of coalition C. A strategy σ_A for player A is said *deterministic* if it does not use randomisation: for all histories h there is an action a such that $\sigma(a \mid h) = 1$. A strategy σ_A for player A is said *stationary* if it depends only on the last state of the history: for all histories h, $\sigma_A(h) = \sigma_A(\mathtt{last}(h))$.

Outcomes. Let C be a coalition, and σ_C a strategy for C. A history h is *compatible* with the strategy σ_C if, for all $k < |h| - 1$, $(\mathsf{move}_k(h))_A = \sigma_A(h_{\leq k})$ for all $A \in C$, and $\mathsf{Tab}(h_k, \mathsf{move}_k(h)) = h_{k+1}$. A play ρ is *compatible* with the strategy σ_C if all its prefixes are. We write $\mathsf{Out}_{\mathcal{G}}(s, \sigma_C)$ for the set of plays in $\mathcal{G}$ that are compatible with strategy σ_C of C and have initial state s, these paths are called *outcomes* of σ_C from s. We simply write $\mathsf{Out}_{\mathcal{G}}(\sigma_C)$ when $s = s_0$ and $\mathsf{Out}_{\mathcal{G}}$ is the set of plays that are compatible with some strategy. Note that when the coalition C is composed of all the players and the strategies are deterministic the outcome is unique.

An *objective* Ω is a set of plays and a strategy σ_C is said *winning* for objective Ω if all its outcomes belong to Ω.

Probability measure induced by a strategy profile. Given a strategy profile σ_{Agt}, the *conditional probability* of a_{Agt} given history $h \in (\mathsf{Stat} \cdot \mathsf{Act}^{\mathsf{Agt}})^* \cdot \mathsf{Stat}$ is $\sigma_{\mathsf{Agt}}(a_{\mathsf{Agt}} \mid h) = \prod_{A\in\mathsf{Agt}} \sigma_A(a_A \mid h)$. The probabilities $\sigma_{\mathsf{Agt}}(a_{\mathsf{Agt}} \mid h)$ induce a probability measure on the Borel σ-algebra over $(\mathsf{Stat} \cdot \mathsf{Act}^{\mathsf{Agt}})^\omega$ as follows: the probability of a basic open set $h \cdot (\mathsf{Stat} \cdot \mathsf{Act}^{\mathsf{Agt}})^\omega$ equals the product $\prod_{j=1}^{n} \sigma_{\mathsf{Agt}}(h^{\mathsf{Act}}_{j,A} \mid h_{\leq j})$ if $h_0 = s_0$ and $\mathsf{Tab}(h_j, h^{\mathsf{Act}}_{\mathsf{Agt},j}) = h_{j+1}$ for all $1 \leq j < n$; in all other cases, this probability is 0. By Carathodory's extension theorem, this extends to a unique probability measure assigning a probability to every Borel subset of $(\mathsf{Stat} \cdot \mathsf{Act}^{\mathsf{Agt}})^\omega$, which we denote by $\mathtt{Pr}^{\sigma_{\mathsf{Agt}}}$. We denote by $\mathrm{E}^{\sigma_{\mathsf{Agt}}}$ the

expectation operator that corresponds to $\mathtt{Pr}^{\sigma_{\mathsf{Agt}}}$, that is $\mathrm{E}^{\sigma_{\mathsf{Agt}}}(f) = \int f \, \mathrm{d}\mathtt{Pr}^{\sigma_{\mathsf{Agt}}}$ for all Borel measurable functions $f\colon (\mathsf{Stat} \cdot \mathsf{Act}^{\mathsf{Agt}})^{\omega} \mapsto \mathbb{R} \cup \{\pm\infty\}$. The expected payoff for player A of a strategy profile σ_{Agt} is $\mathsf{payoff}_A(\sigma_{\mathsf{Agt}}) = \mathrm{E}^{\sigma_{\mathsf{Agt}}}(\mathsf{MP}_{w_A})$.

2.2 Equilibria notions

We now present the different solution concepts we will study. Solution concepts are formal descriptions of "good" strategy profiles. The most famous of them is Nash equilibrium [15], in which no single player can improve the outcome for its own preference relation, by only changing its strategy. This notion can be generalised to consider coalitions of players, it is then called a resilient strategy profile. Nash equilibria correspond to the special case of 1-resilient strategy profiles.

Resilience [3]. Given a coalition $C \subseteq \mathsf{Agt}$, a strategy profile σ_{Agt} is *C-resilient* if for all agents A in C, A cannot improve her payoff even if all agents in C change their strategies, i.e. σ_{Agt} is said C-resilient when:

$$\forall \sigma'_C \in \mathsf{Strat}_{\mathcal{G}}(C).\ \forall A \in C.\ \mathsf{payoff}_A(\sigma_{-C}, \sigma'_C) \leq \mathsf{payoff}_A(\sigma_{\mathsf{Agt}})$$

Given an integer k, we say that a strategy profile is k-resilient if it is C-resilient for every coalition C of size k.

Immunity [1]. Immune strategies ensure that players *not* deviating are not too much affected by deviation. Formally, a strategy profile σ_{Agt} is *(C, r)-immune* if all players not in C, are not worse off by more than r if players in C deviates, i.e. when:

$$\forall \sigma'_C \in \mathsf{Strat}_{\mathcal{G}}(C).\ \forall A \in \mathsf{Agt} \setminus C.\ \mathsf{payoff}_A(\sigma_{\mathsf{Agt}}) - r \leq \mathsf{payoff}_A(\sigma_{-C}, \sigma'_C)$$

Given an integer t, a strategy profile is said *(t, r)-immune* if it is (C, r)-immune for every coalition C of size t. Note that t-immunity as defined in [1] corresponds to $(t, 0)$-immunity.

Robust Equilibrium [1]. Combining resilience and immunity, gives the notion of robust equilibrium: a strategy profile is a *(k, t, r)-robust equilibrium* if it is both k-resilient and (t, r)-immune.

The aim of this article is to characterise robust equilibria in order to construct the corresponding strategies, and precisely describe the complexity of the following decision problem for mean-payoff games.

Robustness Decision Problem. Given a game $\mathcal{G}$, integers k, t, rational r does there exist a profile σ_{Agt}, that is a (k, t, r)-robust equilibrium σ_{Agt} in $\mathcal{G}$?

2.3 Undecidability

We show that allowing strategies that can use both randomisation and memory leads to undecidability. The problem of existence of Nash equilibria has been shown to be undecidable if we put constraints on the payoffs in [19]. The proof was improved to involve only 3 players and no constraint on the payoffs for games with terminal-reward (the weights are non-zero only in terminal vertices of the game) [5]. This corresponds to the particular case where $k = 1$, $t = 0$ for the robustness decision problem. Therefore, the following is a corollary of [5, Thm.13].

Theorem 1. *For randomised strategies, the robustness problem is undecidable even for 3 players, $k = 1$ and $t = 0$.*

To recover decidability, two restrictions are natural. In the next section, we will show that for randomised strategies with no memory, the problem is decidable. The rest of the article is devoted to the second restriction, which concerns pure strategies.

3 Stationary strategies

We use the *existential theory of reals*, which is the set of all existential first-order sentences that hold in the ordered field $\mathfrak{R} := (\mathbb{R}, +, \cdot, 0, 1, \leq)$, to show that the robustness decision problem is decidable. The associated decision problem is in the PSPACE complexity class [11]. However, since the system of equation we produce is of exponential size, we can only show that the robustness problem for (randomised) stationary strategies is in EXPSPACE.

Encoding of strategies We encode a stationary strategy σ_A, by a tuple of real variables $(\varsigma_{s,a})_{s\in\mathsf{Stat},a\in\mathsf{Act}} \in \mathbb{R}^{\mathsf{Stat}\times\mathsf{Act}}$ such that $\varsigma_{s,a} = \sigma_A(a \mid s)$ for all $s \in \mathsf{Stat}$ and $a \in \mathsf{Act}$. We then write a formula saying that these variables describe a correct stationary strategy. The following lemma is a direct consequence of the definition of strategy.

Lemma 1. *Let $(\varsigma_{s,a})_{s\in S,a\in\mathsf{Act}}$ be a tuple of real variables. The mapping $\sigma\colon s, a \mapsto \varsigma_s$ is a stationary strategy if, and only if, ς is a solution of the following equation:*

$$\mu(\varsigma) := \bigwedge_{s\in S} \left(\sum_{a\in\mathsf{Act}} \varsigma_{s,a} = 1 \wedge \bigwedge_{a\in\mathsf{Act}} \varsigma_{s,a} \geq 0\right)$$

Payoff of a profile We now give an equation which links a stationary strategy profile and its payoff. For this we notice that a concurrent game where the stationary strategy profile has been fixed corresponds to a *Markov reward process* [16]. We recall that a Markov reward process is a tuple $\langle S, P, r\rangle$ where: 1. S is a set of states; 2. $P \in \mathbb{R}^{S\times S}$ is a transition matrix; 3. $r\colon S \mapsto \mathbb{R}$ is a reward function. The expected value is then the expectation of the average reward of r. This value for each state is described by a system of equations in [16, Theorem 8.2.6]. We reuse these equations to obtain Thm. 2. Details of the proof are in the appendix.

Theorem 2. *Let σ_{Agt} be a stationary strategy profile. The expectation for MP_w of σ_{Agt} from s is the component γ_s of the solution γ of the equation:*
$\psi(\gamma, \sigma_{\mathsf{Agt}}, w) := \exists \beta \in \mathbb{R}^S.\ \bigwedge_{s \in S} \left(\gamma_s = \sum_{a_{\mathsf{Agt}} \in \mathsf{Act}^{\mathsf{Agt}}} \gamma_{\mathsf{Tab}(s, a_{\mathsf{Agt}})} \cdot \prod_{A \in \mathsf{Agt}} \sigma_A(a_A \mid s)\right)$
$\wedge \bigwedge_{s \in S} \left(\gamma_s + \beta_s = w(s) + \sum_{a_{\mathsf{Agt}} \in \mathsf{Act}^{\mathsf{Agt}}} \beta_{\mathsf{Tab}(s, a_{\mathsf{Agt}})} \cdot \prod_{A \in \mathsf{Agt}} \sigma_A(a_A \mid s)\right)$

Optimal payoff of a deviation We now want to keep the strategy profile fixed but also allow deviations and investigate what is the maximum payoff that a coalition can achieve by deviating. The system that is obtained is then a *Markov decision processes*. We recall that a Markov decision process (MDP) is a tuple $\langle S, A, P, r\rangle$, where: 1. S is a the non-empty, countable set of *states*. 2. A is a set of *actions*. 3. $P\colon S \times A \times S \mapsto [0, 1]$ is the *transition relation*. It is such that for each $s \in S \setminus S_\exists$ and $a \in A$, $\sum_{(s,a,s') \in S \times A \times S} P(s, a, s') = 1$. 4. $r\colon S \mapsto \mathbb{R}$ is the *reward function*. The *optimal value* is then maximal expectation that can be obtained by a strategy. The optimal values in such systems can be described by a linear program [16, Section 9.3]. We reuse this linear program to characterise optimal values against a strategy profile C. Details of the proof can be found in the appendix.

Theorem 3. *Let σ_{Agt} be a stationary strategy profile, C a coalition, s a state and w a weight function. The highest expectation $\mathsf{Agt} \setminus C$ can obtain for w in $\mathcal{G}$ from s: $\sup_{\sigma_{-C}} E^{\sigma_C, \sigma_{-C}}(\mathsf{MP}_w, s)$, is the smallest γ_s component of a solution of the system of inequation:*
$\phi(\gamma, \sigma_C, w) := \bigwedge_{a_{-C} \in \mathsf{Act}^{\mathsf{Agt} \setminus C}} \bigwedge_{s \in S} \gamma_s \geq \sum_{a_C \in \mathsf{Act}^C} \gamma_{\delta(s, a_C, a_{-C})} \cdot \prod_{A \in C} \sigma_A(a_A \mid s)$
$\wedge \bigwedge_{a_{-C} \in \mathsf{Act}^{\mathsf{Agt} \setminus C}} \bigwedge_{s \in S} \gamma_s \geq w(s) - \beta_s + \sum_{a_C \in \mathsf{Act}^C} \left(\beta_{\delta(s, a_C, a_{-C})} \cdot \prod_{A \in C} \sigma_A(a_A \mid s)\right)$

Expressing the existence of robust equilibria Putting these results together, we obtain the results. Intuitively, $\psi(\gamma, \sigma_{\mathsf{Agt}}, w_A)$ ensures that γ correspond to the payoff for σ_{Agt} of A, and $\phi(\gamma', \sigma_{-C}, w_A)$ makes γ' correspond to the payoff for a deviation of σ_C.

Theorem 4. *Let σ_{Agt} be a strategy profile.*

- *It is C-resilient if, and only if, it satisfies the formula:*
  $\rho(C, \sigma_{\mathsf{Agt}}) := \bigwedge_{A \in C} \exists \gamma, \gamma'.\ (\psi(\gamma, \sigma_{\mathsf{Agt}}, w_A) \wedge \phi(\gamma', \sigma_{-C}, w_A) \wedge (\gamma' \leq \gamma))$
- *It is C, r-immune if, and only if, it satisfies equation:*
  $\iota(C, \sigma_{\mathsf{Agt}}) := \bigwedge_{A \notin C} \exists \gamma, \gamma'.\ (\psi(\gamma, \sigma_{\mathsf{Agt}}, A) \wedge \phi(\gamma', \sigma_{-C}, -w_A) \wedge \gamma - r \leq -\gamma')$
- *There is a robust equilibria if, and only if, the following equation is satisfiable:*
  $\exists \varsigma \in \mathbb{R}^{\mathsf{Agt} \times \mathsf{Stat} \times \mathsf{Act}}.\ \mu(\varsigma) \wedge \bigwedge_{C \subseteq \mathsf{Agt} || C| \leq k} \rho(C, \varsigma) \wedge \bigwedge_{C \subseteq \mathsf{Agt} || C| \leq t} \iota(C, \varsigma)$

Theorem 5. *The robustness problem is in* EXPSPACE *for stationary strategies.*

Proof. By Thm. 4, the existence of a robust equilibria is equivalent to the satisfiability of a formula in the existential theory of reals. This formula can be of exponential size with respect to k and t, since a conjunction over coalitions of these size is considered. The best known upper bound for the theory of the reals in PSPACE [11], which gives the EXPSPACE upper bound for our problem.

4 Deviator Game

We now turn to the case of non-randomised strategies. In order to obtain simple algorithms for the robustness problem, we use a correspondence with zero-sum two-players game. Winning strategies has been well studied in computer science and we can make use of existing algorithms. We present the deviator game, which is a transformation of multiplayer game into a turn-based zero-sum game, such that there are strong links between robust equilibria in the first one and winning strategies in the second one. This is formalised in Thm. 6. Note that the proofs of this section are independent from the type of objectives we consider, and the result could be extended beyond mean-payoff objectives.

Deviator. The basic notion we use to solve the robustness problem is that of deviators. It identifies players that cause the current deviation from the expected outcome. A *deviator* from move a_{Agt} to a'_{Agt} is a player $D \in \mathsf{Agt}$ such that $a_D \neq a'_D$. We write this set of deviators: $\mathsf{Dev}(a_{\mathsf{Agt}}, a'_{\mathsf{Agt}}) = \{A \in \mathsf{Agt} \mid a_A \neq a'_A\}$. We extend the definition to histories and strategies by taking the union of deviator sets, formally $\mathsf{Dev}(h, \sigma_{\mathsf{Agt}}) = \bigcup_{0 \leq i < |h|} \mathsf{Dev}(\mathsf{move}_i(h), \sigma_{\mathsf{Agt}}(h_{\leq i}))$. It naturally extends to plays: if ρ is a play, then $\mathsf{Dev}(\rho, \sigma_{\mathsf{Agt}}) = \bigcup_{i \in \mathbb{N}} \mathsf{Dev}(\mathsf{move}_i(\rho), \sigma_{\mathsf{Agt}}(\rho_{\leq i}))$.

Intuitively, given an play ρ and a strategy profile σ_{Agt}, deviators represent the agents that need to change their strategies from σ_{Agt} in order to obtain the play ρ. The intuition is formalised in the following lemma.

Lemma 2. *Let ρ be a play, σ_{Agt} a strategy profile and $C \subseteq \mathsf{Agt}$ a coalition. Coalition C contains $\mathsf{Dev}(\rho, \sigma_{\mathsf{Agt}})$ if, and only if, there exists σ'_C such that $\rho \in \mathsf{Out}_{\mathcal{G}}(\rho_0, \sigma'_C, \sigma_{-C})$.*

4.1 Deviator Game

We now use the notion of deviators to draw a link between multiplayer games and a two-player game that we will use to solve the robustness problem. Given a concurrent game structure $\mathcal{G}$, we define the deviator game $\mathcal{D}(\mathcal{G})$ between two players called `Eve` and `Adam`. Intuitively `Eve` needs to play according to an equilibrium, while `Adam` tries to find a deviation of a coalition which will profit one of its player or harm one of the others. The states are in $\mathsf{Stat}' = \mathsf{Stat} \times 2^{\mathsf{Agt}}$; the second component records the deviators of the current history. The game starts in $(s_0, \varnothing)$ and then proceeds as follows: from a state (s, D), `Eve` chooses an action profile a_{Agt} and `Adam` chooses another one a'_{Agt}, then the next state is $(\mathsf{Tab}(s, a'_{\mathsf{Agt}}), D \cup \mathsf{Dev}(a_{\mathsf{Agt}}, a'_{\mathsf{Agt}}))$. In other words, `Adam` chooses the move that will apply, but this can be at the price of adding players to the D component when he does not follow the choice of `Eve`. The weights of a state (s, D) in this game are the same than that of s in $\mathcal{G}$. The construction of the deviator arena is illustrated in Figure 3.

We now define some transformations between the different objects used in games $\mathcal{G}$ and $\mathcal{D}(\mathcal{G})$. We define projections π_{Stat}, π_{Dev} and π_{Act} from Stat' to Stat, from Stat' to 2^{Agt} and from $\mathsf{Act}^{\mathsf{Agt}} \times \mathsf{Act}^{\mathsf{Agt}}$ to $\mathsf{Act}^{\mathsf{Agt}}$ respectively. They are given

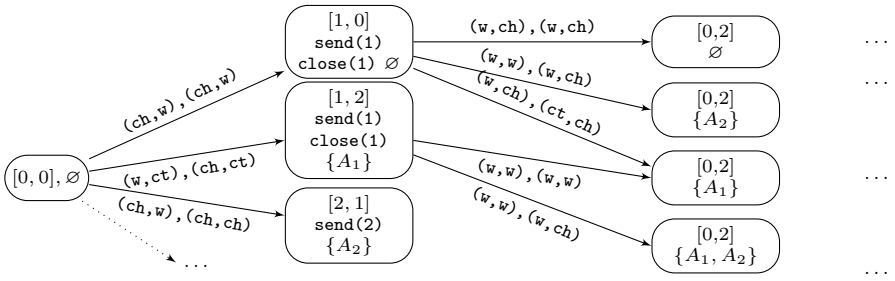

Figure 3: Part of the deviator game construction for the game of Figure 2. Labels on the edges correspond to the action of `Eve` and the action of `Adam`. Labels inside the states are the state of the original game and the deviator component.

by $\pi_{\mathsf{Stat}}(s, D) = s$, $\pi_{\mathsf{Dev}}(s, D) = D$ and $\pi_{\mathsf{Act}}(a_{\mathsf{Agt}}, a'_{\mathsf{Agt}}) = a'_{\mathsf{Agt}}$. We extend these projections to plays in a natural way, letting $\pi_{\mathsf{Out}}(\rho) = \pi_{\mathsf{Stat}}(\rho_0) \cdot \pi_{\mathsf{Act}}(\mathsf{move}_0(\rho)) \cdot \pi_{\mathsf{Stat}}(\rho_1) \cdot \pi_{\mathsf{Act}}(\mathsf{move}_1(\rho)) \cdots$ and $\pi_{\mathsf{Dev}}(\rho) = \pi_{\mathsf{Dev}}(\rho_0) \cdot \pi_{\mathsf{Dev}}(\rho_1) \cdots$. Note that for any play ρ, and any index i, $\pi_{\mathsf{Dev}}(\rho_i) \subseteq \pi_{\mathsf{Dev}}(\rho_{i+1})$, therefore $\pi_{\mathsf{Dev}}(\rho)$ seen as a sequence of sets of coalitions is increasing and bounded by Agt, its limit $\delta(\rho) = \cup_{i \in \mathbb{N}} \pi_{\mathsf{Dev}}(\rho_i)$ is well defined. Moreover to a strategy profile σ_{Agt} in $\mathcal{G}$, we can naturally associate a strategy $\kappa(\sigma_{\mathsf{Agt}})$ for `Eve` in $\mathcal{D}(\mathcal{G})$ such that for all histories h by $\kappa(\sigma_{\mathsf{Agt}})(h) = \sigma_{\mathsf{Agt}}(\pi_{\mathsf{Out}}(h))$.

The following lemma states the correctness of the construction of the deviator game, in the sense that it records the set of deviators in the strategy profile suggested by `Adam` with respect to the strategy profile suggested by `Eve`.

Lemma 3. *Let $\mathcal{G}$ be a game and σ_{Agt} be a strategy profile and $\sigma_\exists = \kappa(\sigma_{\mathsf{Agt}})$ the associated strategy in the deviator game.*

1. *If $\rho \in \mathsf{Out}_{\mathcal{D}(\mathcal{G})}(\sigma_\exists)$, then $\mathsf{Dev}(\pi_{\mathsf{Out}}(\rho), \sigma_{\mathsf{Agt}}) = \delta(\rho)$.*
2. *If $\rho \in \mathsf{Out}_{\mathcal{G}}$ and $\rho' = ((\rho_i, \mathsf{Dev}(\rho_{\leq i}, \sigma_{\mathsf{Agt}})) \cdot (\sigma_{\mathsf{Agt}}(\rho_{\leq i}), \mathsf{move}_i(\rho)))_{i \in \mathbb{N}}$ then $\rho' \in \mathsf{Out}_{\mathcal{D}(\mathcal{G})}(\sigma_\exists)$*

4.2 Objectives of the deviator game

We now show how to transform equilibria notions into objectives of the deviator game. These objectives are defined so that winning strategies correspond to equilibria of the original game. First, we define an objective $\Omega(C, A, G)$ in the following lemma, such that a profile which ensures some quantitative goal $G \subseteq \mathbb{R}$ in $\mathcal{G}$ against coalition C corresponds to a winning strategy in the deviator game.

Lemma 4. *Let $C \subseteq \mathsf{Agt}$ be a coalition, σ_{Agt} be a strategy profile, $G \subseteq \mathbb{R}$ and A a player. We have that for all strategies σ'_C for coalition C, $\mathsf{payoff}_A(\sigma_{-C}, \sigma'_C) \in G$ if, and only if, $\kappa(\sigma_{\mathsf{Agt}})$ is winning in $\mathcal{D}(\mathcal{G})$ for objective $\Omega(C, A, G) = \{\rho \mid \delta(\rho) \subseteq C \Rightarrow \mathsf{payoff}_A(\pi_{\mathsf{Out}}(\rho)) \in G\}$.*

This lemma makes it easy to characterise the different kinds of equilibria, using objectives in $\mathcal{D}(\mathcal{G})$. For instance, we define a *resilience objective* where if there are more than k deviators then Eve has nothing to do; if there are exactly k deviators then she has to show that none of them gain anything; and if there are less than k then no player at all should gain anything. This is because if a new player joins the coalition, its size remains smaller or equal to k. Similar characterisations for immune and robust equilibria lead to the following theorem.

Theorem 6. *Let $\mathcal{G}$ be a concurrent game, σ_{Agt} a strategy profile in $\mathcal{G}$, $p = \mathit{payoff}(\mathsf{Out}(\sigma_{\mathsf{Agt}}))$ the payoff profile of σ_{Agt}, k and t integers, and r a rational.*

- *The strategy profile σ_{Agt} is k-resilient if, and only if, strategy $\kappa(\sigma_{\mathsf{Agt}})$ is winning in $\mathcal{D}(\mathcal{G})$ for the* resilience objective $\mathcal{R}e(k,p)$ *where $\mathcal{R}e(k,p)$ is defined by: $\mathcal{R}e(k,p) = \{\rho \mid |\delta(\rho)| > k\} \cup \{\rho \mid |\delta(\rho)| = k \wedge \forall A \in \delta(\rho).\ \mathit{payoff}_A(\pi_{\mathsf{Out}}(\rho)) \leq p(A)\} \cup \{\rho \mid |\delta(\rho)| < k \wedge \forall A \in \mathsf{Agt}.\ \mathit{payoff}_A(\pi_{\mathsf{Out}}(\rho)) \leq p(A)\}$*
- *The strategy profile σ_{Agt} is (t,r)-immune if, and only if, strategy $\kappa(\sigma_{\mathsf{Agt}})$ is winning for the* immunity objective $\mathcal{I}(t,r,p)$ *$\mathcal{I}(t,r,p)$ is defined by: $\mathcal{I}(t,r,p) = \{\rho \mid |\delta(\rho)| > t\} \cup \{\rho \mid \ \forall A \in \mathsf{Agt} \setminus \delta(\rho).\ p(A) - r \leq \mathit{payoff}_A(\pi_{\mathsf{Stat}}(\rho))\}$*
- *The strategy profile σ_{Agt} is a (k,t,r)-robust profile in $\mathcal{G}$ if, and only if, $\kappa(\sigma_{\mathsf{Agt}})$ is winning for the* robustness objective $\mathcal{R}(k,t,r,p) = \mathcal{R}e(k,p) \cap \mathcal{I}(t,r,p)$.

5 Reduction to multidimensional mean-payoff objectives

We first show that the deviator game reduces the robustness problem to a winning strategy problem in multidimensional mean-payoff games. We then solve this by requests to the polyhedron value problem of [9].

5.1 Multidimensional objectives

Multidimensional mean-payoff objective. Let $\mathcal{G}$ be a two-player game, $v\colon \mathsf{Stat} \mapsto \mathbb{Z}^d$ a multidimensional weight functions and $I, J \subseteq [\![1,d]\!]$[1] a partition of $[\![1,d]\!]$ (i.e. $I \uplus J = [\![1,d]\!]$). We say that Eve *ensures threshold* $u \in \mathbb{R}^d$ if she has a strategy $\sigma_\exists$ such that all outcomes ρ of $\sigma_\exists$ are such that for all $i \in I$, $\mathsf{MP}_{v_i}(\rho) \geq u_i$ and for all $j \in J$, $\overline{\mathsf{MP}}_{v_j}(\rho) \geq u_j$, where $\overline{\mathsf{MP}}_{v_j}(\rho) = \limsup_{n\to\infty} \frac{1}{n} \sum_{0\leq k\leq n} v_j(\rho_k)$. That is, for all dimensions $i \in I$, the limit inferior of the average of v_i is greater than u_i and for all dimensions $j \in J$ the limit superior of v_j is greater than u_j.

We consider two decision problems on these games: 1. The *value problem*, asks given $\langle \mathcal{G}, v, I, J\rangle$ a game with multidimensional mean-payoff objectives, and $u \in \mathbb{R}^d$, whether Eve can ensure u. 2. The *polyhedron value problem*, asks given $\langle \mathcal{G}, v, I, J\rangle$ a game with multidimensional mean-payoff objectives, and $(\lambda_1, \ldots, \lambda_n)$ a tuple of linear inequations, whether there exists a threshold u which Eve can ensure and that satisfies the inequation λ_i for all i in $[\![1,d]\!]$. We assume that all linear inequations are given by a tuple $(a_1, \ldots, a_d, b) \in \mathbb{Q}^{d+1}$ and

[1] We write $[\![i,j]\!]$ for the set of integers $\{k \in \mathbb{Z} \mid i \leq k \leq j\}$.

that a point $u \in \mathbb{R}^d$ satisfies it when $\sum_{i \in [\![1,d]\!]} a_i \cdot u_i \geq b$. The value problem was showed to be coNP-complete [20] while the polyhedron value problem is Σ_2P-complete [9]. Our goal is now to reduce our robustness problem to a polyhedron value problem for some well chosen weights.

In our case, the number d of dimensions will be equal to $4 \cdot |\mathsf{Agt}|$. We then number players so that $\mathsf{Agt} = \{A_1, \ldots, A_{|\mathsf{Agt}|}\}$. Let $W = \max\{|w_i(s)| \mid A_i \in \mathsf{Agt}, s \in \mathsf{Stat}\}$ be the maximum constant occurring in the weights of the game, notice that for all players A_i and play ρ, $-W - 1 < \mathsf{MP}_i(\rho) \leq W$. We fix parameters k, t and define our weight function $v \colon \mathsf{Stat} \mapsto \mathbb{Z}^d$. Let $i \in [\![1, |\mathsf{Agt}|]\!]$, the weights are given for $(s, D) \in \mathsf{Stat} \times 2^{\mathsf{Agt}}$ by:

1. if $|D| \leq t$ and $A_i \notin D$, then $v_i(s, D) = w_{A_i}(s)$;
2. if $|D| > t$ or $A_i \in D$, then $v_i(s, D) = W$;
3. if $|D| < k$, then for all $A_i \in \mathsf{Agt}$, $v_{|\mathsf{Agt}|+i}(s, D) = -w_{A_i}(s)$;
4. if $|D| = k$ and $A_i \in D$, then $v_{|\mathsf{Agt}|+i}(s, D) = -w_{A_i}(s)$;
5. if $|D| > k$ or $A_i \notin D$ and $|D| = k$, then $v_{|\mathsf{Agt}|+i}(s, D) = W$.
6. if $D = \varnothing$ then $v_{2\cdot|\mathsf{Agt}|+i}(s, D) = w_{A_i}(s) = -v_{3\cdot|\mathsf{Agt}|+i}(s, D)$;
7. if $D \neq \varnothing$ then $v_{2\cdot|\mathsf{Agt}|+i}(s, D) = W = v_{3\cdot|\mathsf{Agt}|+i}(s, D)$;

We take $I = [\![1, |\mathsf{Agt}|]\!] \cup [\![2 \cdot |\mathsf{Agt}| + 1, 3 \cdot |\mathsf{Agt}|]\!]$ and $J = [\![|\mathsf{Agt}| + 1, 2 \cdot |\mathsf{Agt}|]\!] \cup [\![3 \cdot |\mathsf{Agt}| + 1, 4 \cdot |\mathsf{Agt}|]\!]$. Intuitively, the components $[\![1, |\mathsf{Agt}|]\!]$ are used for immunity, the components $[\![|\mathsf{Agt}| + 1, 2 \cdot |\mathsf{Agt}|]\!]$ are used for resilience and components $[\![2 \cdot |\mathsf{Agt}| + 1, 4 \cdot |\mathsf{Agt}|]\!]$ are used to constrain the payoff in case of no deviation.

5.2 Correctness of the objectives for robustness

Let $\mathcal{G}$ be a concurrent game, ρ a play of $\mathcal{D}(\mathcal{G})$ and $p \in \mathbb{R}^{\mathsf{Agt}}$ a payoff vector. The following lemma links the weights we chose and our solution concepts.

Lemma 5. *Let ρ be a play, σ_{Agt} a strategy profile and $p = \mathsf{payoff}(\sigma_{\mathsf{Agt}})$.*

- *ρ satisfies objective $\delta(\rho) = \varnothing \Rightarrow \mathsf{MP}_{A_i}(\rho) = p_i$ if, and only if, $\mathsf{MP}_{2\cdot v_{|\mathsf{Agt}|+i}}(\rho) \geq p(A_i)$ and $\overline{\mathsf{MP}}_{3\cdot v_{|\mathsf{Agt}|+i}}(\rho) \geq -p(A_i)$.*
- *If ρ is an outcome of $\kappa(\sigma_{\mathsf{Agt}})$ then ρ satisfies objective $\mathcal{R}e(k, p)$ if, and only if, for all agents A_i, $\overline{\mathsf{MP}}_{v_{|\mathsf{Agt}|+i}}(\rho) \geq -p(A_i)$.*
- *If ρ is an outcome of $\kappa(\sigma_{\mathsf{Agt}})$, then ρ satisfies objective $\mathcal{I}(t, r, p)$ if, and only if, for all agents A_i, $\mathsf{MP}_{v_i}(\rho) \geq p(A_i) - r$.*
- *If ρ is an outcome of $\kappa(\sigma_{\mathsf{Agt}})$ with $\mathsf{payoff}(\sigma_{\mathsf{Agt}}) = p$, then play ρ satisfies objective $\mathcal{R}(k, t, r, p)$ if, and only if, for all agents A_i, $\mathsf{MP}_{v_i}(\rho) \geq p(A_i) - r$ and $\overline{\mathsf{MP}}_{v_{|\mathsf{Agt}|+i}}(\rho) \geq -p(A_i)$.*

Putting together this lemma and the correspondence between the deviator game and robust equilibria of Thm. 6 we obtain the following proposition.

Lemma 6. *Let $\mathcal{G}$ be a concurrent game with mean-payoff objectives. There is a (k, t, r)-robust equilibrium in $\mathcal{G}$ if, and only if, for the multidimensional mean-payoff objective given by v, $I = [\![1, |\mathsf{Agt}|]\!] \cup [\![2 \cdot |\mathsf{Agt}| + 1, 3 \cdot |\mathsf{Agt}|]\!]$ and $J = [\![|\mathsf{Agt}| + 1, 2 \cdot |\mathsf{Agt}|]\!] \cup [\![3 \cdot |\mathsf{Agt}| + 1, 4 \cdot |\mathsf{Agt}|]\!]$, there is a payoff vector p such that Eve can ensure threshold u in $\mathcal{D}(\mathcal{G})$, where for all $i \in [\![1, |\mathsf{Agt}|]\!]$, $u_i = p(A_i) - r$, $u_{|\mathsf{Agt}|+i} = -p(A_i)$, $u_{2\cdot|\mathsf{Agt}|+i} = p(A_i)$, and $u_{3\cdot|\mathsf{Agt}|+i} = -p(A_i)$.*

5.3 Formulation of the robustness problem as a polyhedron value problem

From the previous lemma, we can deduce an algorithm which works by querying the polyhedron value problem. Given a game $\mathcal{G}$ and parameters k, t, r, we ask whether there exists a payoff u that `Eve` can ensured in the game $\mathcal{D}(\mathcal{G})$ with multidimensional mean-payoff objective given by v, I, J, and such that for all $i \in [\![1, |\mathsf{Agt}|]\!]$, $u_i + r = -u_{|\mathsf{Agt}|+i} = u_{2\cdot|\mathsf{Agt}|+i} = -u_{3\cdot|\mathsf{Agt}|+i}$. As we will show in Thm. 7 thanks to Lem. 6, the answer to this question is yes if, and only if, there is a (k, t, r)-robust equilibrium. From the point of view of complexity, however, the deviator game on which we perform the query can be of exponential size compared to the original game. To describe more precisely the complexity of the problem, we remark by applying the bound of [8, Thm. 22], that given a query, we can find solutions which have a small representation.

Lemma 7. *If there is a solution to the polyhedron value problem in $\mathcal{D}(\mathcal{G})$ then there is one whose encoding is of polynomial size with respect to $\mathcal{G}$ and the polyhedron given as input.*

We can therefore enumerate all possible solutions in polynomial space. To obtain an polynomial space algorithm, we must be able to check one solution in polynomial space as well. This account to show that queries for the value problem in the deviator game can be done in space polynomial with respect to the original game. This is the goal of the next section, and it is done by considering small parts of the deviator game called fixed coalition games.

6 Fixed coalition game

Although the deviator game may be of exponential size, it presents a particular structure. As the set of deviators only increases during any run, the game can be seen as the product of the original game with a directed acyclic graph (DAG). The nodes of this DAG correspond to possible sets of deviators, it is of exponential size but polynomial degree and depth. We exploit this structure to obtain a polynomial space algorithm for the value problem and thus also for the polyhedron value problem and the robustness problem. The idea is to compute winning states in one component at a time, and to recursively call the procedure for states that are successors of the current component. We will therefore consider one different game for each component.

We now present the details of the procedure. For a fixed set of deviator D, the possible successors of states of the component $\mathsf{Stat} \times D$ are the states in: $\mathsf{Succ}(D) = \{\mathsf{Tab}_{\mathcal{D}}((s, D), (m_{\mathsf{Agt}}, m'_{\mathsf{Agt}})) \mid s \in \mathsf{Stat}, m_{\mathsf{Agt}}, m'_{\mathsf{Agt}} \in \mathsf{Mov}(s)\} \setminus \mathsf{Stat} \times D\}$. Note that the size of $\mathsf{Succ}(D)$ is bounded by $|\mathsf{Stat}| \times |\mathsf{Tab}|$, hence it is polynomial. Let u be a payoff threshold, we want to know whether `Eve` can ensure u in $\mathcal{D}(\mathcal{G})$, for the multi-dimensional objective defined in Section 5.1. A winning path ρ from a state in $\mathsf{Stat} \times D$ is either: 1) such that $\delta(\rho) = D$; 2) or it reaches a state in $\mathsf{Succ}(D)$ and follow a winning path from there. Assume we

have computed all the states in $\mathsf{Succ}(D)$ that are winning. We can stop the game as soon as $\mathsf{Succ}(D)$ is reached, and declare $\mathtt{Eve}$ the winner if the state that is reached is a winning state of $\mathcal{D}(\mathcal{G})$. This process can be seen as a game $\mathcal{F}(D,u)$, called the *fixed coalition game*.

In this game the states are those of $(\mathsf{Stat} \times D) \cup \mathsf{Succ}(D)$; transitions are the same than in $\mathcal{D}(\mathcal{G})$ on the states of $\mathsf{Stat} \times D$ and the states of $\mathsf{Succ}(D)$ have only self loops. The winning condition is identical to $\mathcal{R}(k,t,p)$ for the plays that never leave $(\mathsf{Stat} \times D)$; and for a play that reach some $(s',D') \in \mathsf{Succ}(D)$, it is considered winning $\mathtt{Eve}$ has a winning strategy from (s',D') in $\mathcal{D}(\mathcal{G})$ and losing otherwise.

In the fixed coalition game, we keep the weights previously defined for states of $\mathsf{Stat} \times D$, and fix it for the states that are not in the same D component by giving a high payoff on states that are winning and a low one on the losing ones. Formally, we define a multidimensional weight function v^f on $\mathcal{F}(D,u)$ by: 1. for all $s \in \mathsf{Stat}$, and all $i \in [\![1, 4 \cdot |\mathsf{Agt}|]\!]$, $v_i^f(s,D) = v_i(s,D)$. 2. if $(s,D') \in \mathsf{Succ}(D)$ and $\mathtt{Eve}$ can ensure u from (s,D'), then for all $i \in [\![1, 4 \cdot |\mathsf{Agt}|]\!]$, $v_i^f(s,D') = W$. 3. if $(s,D') \in \mathsf{Succ}(D)$ and $\mathtt{Eve}$ cannot ensure u from (s,D'), then for all $i \in [\![1, 4 \cdot |\mathsf{Agt}|]\!]$, $v_i^f(s,D') = -W - 1$.

Lemma 8. *$\mathtt{Eve}$ can ensure payoff $u \in [\![-W, W]\!]^d$ in $\mathcal{D}(\mathcal{G})$ from (s,D) if, and only if, she can ensure u in the fixed coalition game $\mathcal{F}(D,p)$ from (s,D).*

Using this correspondence, we deduce a polynomial space algorithm to check that $\mathtt{Eve}$ can ensure a given value in the deviator game and thus a polynomial space algorithm for the robustness problem.

Theorem 7. *There is a polynomial space algorithm, that given a concurrent game $\mathcal{G}$, tells if there is a (k,t,r)-robust equilibrium.*

Proof. We first show that there is a polynomial space algorithm to solve the value problem in $\mathcal{D}(\mathcal{G})$. We consider a threshold u and a state (s,D). In the fixed coalition game $\mathcal{F}(D,u)$, for each $(s',D') \in \mathsf{Succ}(D)$, we can compute whether it is winning by recursive calls. Once the weights for all $(s',D') \in \mathsf{Succ}(D)$ have been computed for $\mathcal{F}(D,u)$, we can solve the value problem in $\mathcal{F}(D,u)$. Thanks to Lem. 8 the answer to value problem in this game is yes exactly when $\mathtt{Eve}$ can ensure u from (s,D) in $\mathcal{D}(\mathcal{G})$. There is a coNP algorithm [20] to check the value problem in a given game, and therefore there also is an algorithm which uses polynomial space. The size of the stack of recursive calls is bounded by $|\mathsf{Agt}|$, so the global algorithm uses polynomial space.

We now use this to show that there is a polynomial space algorithm for the polyhedron value problem in $\mathcal{D}(\mathcal{G})$. We showed in Lem. 7, that if the polyhedron value problem has a solution then there is a threshold u of polynomial size that is witness of this property. We can enumerate all the thresholds that satisfy the size bound in polynomial space. We can then test that these thresholds satisfy the given linear inequations, and that the algorithm for the value problem answers yes on this input, in polynomial space thanks to the previous algorithm. If this

is the case for one of the thresholds, then we answer yes for the polyhedron value problem. The correctness of this procedure holds thanks to Lem. 7.

We now use this to show that there is a polynomial space algorithm for the robustness problem. Given a game $\mathcal{G}$ and parameters (k, t, r), we define a tuple of linear equations, for all $i \in [\![1, |\mathsf{Agt}|]\!]$, $x_{2\cdot|\mathsf{Agt}|+i} = x_i + r \wedge x_{2\cdot|\mathsf{Agt}|+i} = -x_{|\mathsf{Agt}|+i} \wedge x_{2\cdot|\mathsf{Agt}|+i} = -x_{3\cdot|\mathsf{Agt}|+i}$ (each equation can be expressed by two inequations). Thanks to Lem. 6, there is a payoff which satisfies these constraints and which `Eve` can ensure in $\mathcal{D}(\mathcal{G})$ if, and only if, there is a (k, t, r)-robust equilibrium. Then, querying the algorithm we described for the polyhedron value problem in $\mathcal{D}(\mathcal{G})$ with our system of inequations, answers the robustness problem.

7 Hardness

In this section, we show a matching lower bound for the resilience problem. The lower bound holds for weights that are 0 in every states except on some terminal states where they can be 1. This is also called *simple reachability objectives*.

Theorem 8. *The robustness problem is* PSPACE*-complete.*

Note that we already proved PSPACE-membership in Thm. 7. We give the construction and intuition of the reduction to show hardness and leave the proof of correctness in the appendix. We encode QSAT formulas with n variable into a game with $2 \cdot n + 2$ players, such that the formula is valid if, and only if, there is n-resilient equilibria. We assume that we are given a formula of the form $\phi = \forall x_1.\exists x_2.\, \forall x_3 \cdot \exists x_n.\, C_1 \wedge \cdots \wedge C_k$, where each C_k is of the form $\ell_{1,k} \vee \ell_{2,k} \vee \ell_{3,k}$ and each $\ell_{j,k}$ is a literal (i.e. x_m or $\neg x_m$ for some m). We define the game $\mathcal{G}_\phi$ as illustrated by an example in Figure 4. It has a player A_m for each positive literal x_m, and a player B_m for each negative literal $\neg x_m$. We add two extra players `Eve` and `Adam`. `Eve` is making choices for the existential quantification and `Adam` for the universal ones. When they chose a literal, the corresponding player can either go to a sink state $\bot$ or continue the game to the next quantification. Once a literal has been chosen for all the variables, `Eve` needs to chose a literal for each clause. The objective for `Eve` and the literal players is to reach $\bot$. The objective for `Adam` is to reach $\top$. We ask whether there is a $(n+1)$-resilient equilibrium.

To a history $h = \mathtt{Adam}_1 \cdot X_1 \cdot \mathtt{Eve}_2 \cdot X_2 \cdot \mathtt{Adam}_3 \cdots \mathtt{Eve}_m \cdot X_m$ with $X_i \in \{A_i, B_i\}$, we associate a valuation v_h, such that $v_h(x_i) = \mathsf{true}$ if $X_i = B_i$ and $v_h(x_i) = \mathsf{false}$ if $X_i = A_i$. Intuitively, `Eve` has to find a valuation that makes the formula hold, while `Adam` tries to falsify it.

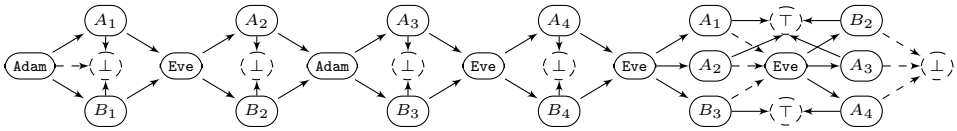

Figure 4: Encoding of a formula $\phi = \forall x_1.\exists x_2.\ \forall x_3.\ \exists x_4.\ (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_3 \vee x_4)$. The dashed edges represent the strategies in the equilibrium of the players other than `Eve`.

References

1. I. Abraham, D. Dolev, R. Gonen, and J. Halpern. Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. In *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pages 53–62. ACM, 2006.
2. R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM (JACM)*, 49(5):672–713, 2002.
3. R. Aumann. Acceptable points in general cooperative n-person games. *Topics in Mathematical Economics and Game Theory Essays in Honor of Robert J Aumann*, 23:287–324, 1959.
4. P. Bouyer, R. Brenguier, N. Markey, and M. Ummels. Concurrent games with ordered objectives. In L. Birkedal, editor, *FoSSaCS'12*, volume 7213 of *LNCS*, pages 301–315. Springer-Verlag, Mar. 2012.
5. P. Bouyer, N. Markey, and D. Stan. Mixed Nash Equilibria in Concurrent Terminal-Reward Games. In *34th International Conference on Foundation of Software Technology and Theoretical Computer Science (FSTTCS 2014)*, volume 29 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 351–363, Dagstuhl, Germany, 2014.
6. R. Brenguier. *Nash equilibria in concurrent games: application to timed games.* PhD thesis, Cachan, Ecole normale supérieure, 2012.
7. R. Brenguier. Robust equilibria in concurrent games. *CoRR*, abs/1311.7683, 2015.
8. R. Brenguier and J.-F. Raskin. Optimal values of multidimensional mean-payoff games, 2014. https://hal.archives-ouvertes.fr/hal-00977352/.
9. R. Brenguier and J.-F. Raskin. Pareto curves of multidimensional mean-payoff games. In *Computer Aided Verification*, pages 251–267. Springer, 2015.
10. T. Brihaye, V. Bruyère, and J. De Pril. Equilibria in quantitative reachability games. In F. Ablayev and E. Mayr, editors, *Computer Science Theory and Applications*, volume 6072 of *LNCS*, pages 72–83. Springer Berlin / Heidelberg, 2010.
11. J. Canny. Some algebraic and geometric computations in pspace. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 460–467. ACM, 1988.
12. K. Chatterjee, T. A. Henzinger, and M. Jurdziński. Games with secure equilibria. In *Formal Methods for Components and Objects*, pages 141–161. Springer, 2005.
13. K. Chatterjee, T. A. Henzinger, and N. Piterman. Strategy logic. *Information and Computation*, 208(6):677–693, 2010.
14. F. Mogavero, A. Murano, G. Perelli, and M. Y. Vardi. What makes ATL* decidable? a decidable fragment of strategy logic. In *CONCUR 2012–Concurrency Theory*, pages 193–208. Springer, 2012.

15. J. F. Nash, Jr. Equilibrium points in n-person games. *Proc. National Academy of Sciences of the USA*, 36(1):48–49, Jan. 1950.
16. M. L. Puterman. Markov decision processes: Discrete stochastic dynamic programming. 1994.
17. M. Ummels. The complexity of Nash equilibria in infinite multiplayer games. In *Proc. of FoSSaCS'12*, volume 4962 of *LNCS*, pages 20–34. Springer, 2008.
18. M. Ummels and D. Wojtczak. The complexity of Nash equilibria in simple stochastic multiplayer games. *Automata, Languages and Programming*, pages 297–308, 2009.
19. M. Ummels and D. Wojtczak. The complexity of Nash equilibria in limit-average games. *CONCUR 2011–Concurrency Theory*, pages 482–496, 2011.
20. Y. Velner, K. Chatterjee, L. Doyen, T. A. Henzinger, A. Rabinovich, and J.-F. Raskin. The complexity of multi-mean-payoff and multi-energy games. *CoRR*, abs/1209.3234, 2012.

A Appendix for Section 3

We will recall how to compute values of *Markov reward process* (Lem. **??**), make explicit the correspondence with our problem on concurrent games (Lem. **??**), and then give equations describing the payoff of strategy profile.

Definition 1 (Markov reward processes). *A* Markov reward process *is a tuple $\langle S, P, r\rangle$ where: 1. S is a set of states; 2. $P \in \mathbb{R}^{S\times S}$ is a transition matrix; 3. $r\colon S \mapsto \mathbb{R}$ is a reward function. The probability of a history $h \in S^*$, is $\prod_{0\leq i<|h|} P(h_i, h_{i+1})$. The expected value of a Markov reward process is then defined similarly to the expected value of a concurrent game, as the expectation of MP_r. The* gain $g \in \mathbb{R}^S$ *is a vector such that for each state s, $g(s)$ is the expected value from state s.*

A.1 Proof of Thm. 2

We recall the result of [16] that we will use.

Theorem 9 ([16, Theorem 8.2.6]). *Let $\langle S, P, r\rangle$ be Markov reward process. We consider real variables of the form γ_s and β_s for each $s \in S$. The gain is uniquely characterised has the solution for γ of the equations $(I - P)\cdot\gamma = 0$ and $\gamma + (I - P)\cdot\beta = r$ where I is the identity matrix. This solution is such that γ is equal to the gain*[2]*.*

Thanks to this result, we can show that solutions of the following equation are such that the mapping $g\colon s \mapsto \gamma_s$ corresponds to the gain:

$$\exists \beta \in \mathbb{R}^S.\ \bigwedge_{s\in S}\left(\gamma_s = \sum_{s'\in S} P_{s,s'}\cdot\gamma_{s'}\right) \wedge \bigwedge_{s\in S}\left(\gamma_s + \beta_s = r(s) + \sum_{s'\in S} P_{s,s'}\cdot\beta_{s'}\right) \tag{1}$$

Lemma 9. *In a Markov reward process $\langle S, P, r\rangle$, the solution γ of equation (**??**) is such that for each state s, $g(s) = \gamma_s$ where g is the gain function.*

Proof. By using **??**, g (the gain) and b (the bias) are the solutions for γ and β of the equations of $(I - P)\cdot\gamma = 0$ and $\gamma + (I - P)\cdot\beta = r$. We can rewrite the equation and since we are only interested in g we can abstract b:

$$\exists \beta \in \mathbb{R}^S.\ \bigwedge_{s\in S}\left(\gamma_s - \sum_{s'\in S} P_{s,s'}\cdot\gamma_{s'} = 0\right) \wedge \bigwedge_{s\in S}\left(\gamma_s + \beta_s - \sum_{s'\in S} P_{s,s'}\cdot\beta_{s'} = r(s)\right)$$

Which we then rewrite in the form of equation (**??**).

We now make explicit the correspondence with concurrent games. Let $\mathcal{G}$ be a concurrent game, σ_{Agt} be a strategy profile and w a weight function. We define the Markov reward process $\mathtt{MRP}(\mathcal{G}, \sigma_{\mathsf{Agt}}, w) = \langle \mathsf{Stat}, P, w\rangle$ where for all $(s, s') \in \mathsf{Stat}\times\mathsf{Stat}$, $P(s, s') = \sum_{a_{\mathsf{Agt}}|\mathsf{Tab}(s,a_{\mathsf{Agt}})=s'} \prod_{A\in\mathsf{Agt}} \sigma_A(a_A \mid s)$.

[2] β is equal to a quantity called the bias [16]

Lemma 10. *The expectation for* MP_w *of the strategy profile* σ_{Agt} *is equal to the expected value of the Markov reward process* $\mathtt{MRP}(\mathcal{G}, \sigma_{\mathsf{Agt}}, w)$.

Proof. For proving the lemma, since the weight functions are the same, it is enough to prove that the probability of an history in the Markov reward process is the same as its probability in the concurrent game knowing that the strategy profile σ_{Agt}. This is done by induction over the length of history h. The case of length 1 is obvious. Now assume that h is a history that has same probability $p(h)$ in $\mathcal{G}$ knowing σ_{Agt} and in $\mathtt{MRP}(\mathcal{G}, \sigma_{\mathsf{Agt}}, A)$. The probability of $h \cdot s$ in $\mathcal{G}$ knowing σ_{Agt} is the sum over actions profiles a_{Agt} that lead from s to s' of the probability that the strategy profile σ_{Agt} chooses a_{Agt}. This equals $p(h) \cdot \sum_{a_{\mathsf{Agt}} | \mathsf{Tab}(s, a_{\mathsf{Agt}}) = s'} \prod_{A \in \mathsf{Agt}} \sigma_A(a_A \mid s)$, which is also equal to the probability of a transition from s to s' in $\mathtt{MRP}(\mathcal{G}, \sigma_{\mathsf{Agt}}, w)$ times $p(h)$. Therefore the probability of $h \cdot s$ is equal in $\mathcal{G}$ knowing σ_{Agt} and in $\mathtt{MRP}(\mathcal{G}, \sigma_{\mathsf{Agt}}, w)$. This shows the property.

Theorem 10. *(Thm. 2 in the body of the paper) Let* σ_{Agt} *be a stationary strategy profile. The expectation for* MP_w *of* σ_{Agt} *from* s *is the component* γ_s *of the solution* γ *of the equation:*

$$\psi(\gamma, \sigma_{\mathsf{Agt}}, w) := \exists \beta \in \mathbb{R}^S.\ \bigwedge_{s \in S} \left(\gamma_s = \sum_{a_{\mathsf{Agt}} \in \mathsf{Act}^{\mathsf{Agt}}} \gamma_{\mathsf{Tab}(s, a_{\mathsf{Agt}})} \cdot \prod_{A \in \mathsf{Agt}} \sigma_A(a_A \mid s) \right)$$
$$\wedge \bigwedge_{s \in S} \left(\gamma_s + \beta_s = w(s) + \sum_{a_{\mathsf{Agt}} \in \mathsf{Act}^{\mathsf{Agt}}} \beta_{\mathsf{Tab}(s, a_{\mathsf{Agt}})} \cdot \prod_{A \in \mathsf{Agt}} \sigma_A(a_A \mid s) \right)$$

Proof. Thanks to Lem. **??** the expected payoff is the same than the value of $\mathtt{MRP}(\mathcal{G}, \sigma_{\mathsf{Agt}}, w)$. If we apply Lem. **??** to this Markov reward process, and replace P by its expression given from σ_{Agt}, r by w, we obtain that its gain is given by the equation:

$$\exists \beta \in \mathbb{R}^S.\ \bigwedge_{s \in S} \left(\gamma_s = \sum_{s' \in S} \sum_{\{a_{\mathsf{Agt}} | \mathsf{Tab}(s, a_{\mathsf{Agt}}) = s'\}} \gamma_{s'} \cdot \prod_{A \in \mathsf{Agt}} \sigma_A(a_A \mid s) \right)$$
$$\wedge \bigwedge_{s \in S} \left(\gamma_s + \beta_s = w(s) + \sum_{s' \in S} \sum_{\{a_{\mathsf{Agt}} | \mathsf{Tab}(s, a_{\mathsf{Agt}}) = s'\}} \beta_{s'} \cdot \prod_{A \in \mathsf{Agt}} \sigma_A(a_A \mid s) \right)$$

We can then rewrite this, noticing that $\sum_{s' \in S} \sum_{\{a_{\mathsf{Agt}} | \mathsf{Tab}(s, a_{\mathsf{Agt}}) = s'\}} f(s, s', a_{\mathsf{Agt}})$ is the same as $\sum_{a_{\mathsf{Agt}} \in \mathsf{Act}^{\mathsf{Agt}}} f(s, \mathsf{Tab}(s, a_{\mathsf{Agt}}), a_{\mathsf{Agt}})$ and we obtain the desired expression.

We will recall how to compute values of such processes (Lem. **??**), draw the link with our problem(Lem. **??**), and then give equations linking the maximum payoff a coalition can achieve with the given strategy profile (Thm. 3).

Definition 2 ([16,19]). *A* Markov decision process *(MDP) is given by a tuple* $\langle S, A, P, r \rangle$*, where:*

- S *is a the non-empty, countable set of* states.

– A *is a set of* actions.
– $P\colon S \times A \times S \mapsto [0,1]$ *is the* transition relation. *It is such that for each* $s \in S \setminus S_\exists$ *and* $a \in A$, $\sum_{(s,a,s')\in S\times A\times S} P(s,a,s') = 1$.
– $r\colon S \mapsto \mathbb{R}$ *is the* reward function.

Given a policy $\sigma\colon S \to A$, *the probability of a history* $h \in S^*$, *is* $\prod_{0\leq i<|h|} \sum_{a\in A} P(h_i, \sigma(a \mid h_i), h_{i+1})$. *The* expected value $E^\sigma(r)$ *of a policy* σ *in a Markov reward process is then defined similarly to the expected value of a concurrent game. The* optimal average reward *is the maximum over the policies of the expected value: the optimal average reward of the MDP* M *from state* s *is written* $v(M,s)$ *and equals* $\sup_{\{\sigma\colon S\to A\}} E^\sigma(\mathsf{MP}_r)$.

In [16, Section 9.3] it is shown that the optimal average reward of a MDP is characterised by the following linear programs:

Minimise $\sum_{s\in S} \alpha_s \cdot \gamma_s$ where for all states s, $\alpha_s > 0$ and $\sum_{s\in S} \alpha_s = 1$ subject to:

$$\begin{aligned} \bigwedge_{a\in\mathsf{Act}} \bigwedge_{s\in S} \gamma_s &\geq \textstyle\sum_{s'\in S} P(s,a,s') \cdot \gamma_{s'} \\ \wedge \bigwedge_{a\in\mathsf{Act}} \bigwedge_{s\in S} \gamma_s &\geq r(s) + \textstyle\sum_{s'\in S} P(s,a,s') \cdot \beta_{s'} - \beta_s \end{aligned} \tag{2}$$

Thanks to this linear program we can characterise optimal values against a strategy profile C.

We reformulate the results of [16, Section 9.3] in the following lemma.

Lemma 11. *Equation* (**??**) *is fulfilled by* $\gamma = (v(M,s))_{s\in S}$ *and if it is fulfilled by some vector* γ *then* $\gamma_s \geq v(M,s)$ *for all states* s.

Proof. That equation (**??**) if fulfilled by $v(M,s)$ is obvious because it is the solution of the linear program. Assume now that some γ satisfies ϕ.

First notice that $\sum_{s\in S} \gamma_s - v(M,s) \geq 0$ because γ is a solution of equation **??** and $v(M,s)$ is the solution that minimises $\sum_{s\in S} \frac{\gamma_s}{|S|}$ (taking $\alpha_s = \frac{1}{|S|}$ for all $s \in S$).

Towards a contradiction assume there is a state t, such that $\gamma_t < v(M,t)$. Given some ε with $0 < \varepsilon < 1$, we let $\alpha_t = 1-\varepsilon$ and $\alpha_s = \frac{\varepsilon}{|S|-1}$ for each $s \neq t$. We have that the constraints $\alpha_s > 0$ for all states s and $\sum_{s\in S} \alpha_s = 1$ are satisfied.

Moreover:

$$\begin{aligned}
&\sum \alpha_s \cdot \gamma_s - \sum \alpha_s \cdot v(M,s) \\
&\quad = \sum \alpha_s \cdot (\gamma_s - v(M,s)) \\
&\quad = \alpha_t \cdot (\gamma_t - v(M,t)) + \sum_{s \neq t} \alpha_s \cdot (\gamma_s - v(M,s)) \\
&\quad = (\gamma_t - v(M,t)) - \varepsilon \cdot (\gamma_t - v(M,t)) + \sum_{s \neq t} \varepsilon \cdot \frac{(\gamma_s - v(M,s))}{|S| - 1} \\
&\quad = (\gamma_t - v(M,t)) - \varepsilon \cdot \left(\left(1 + \frac{1}{|S| - 1} \right) \cdot (\gamma_t - v(M,t)) - \sum_{s \in S} \frac{\gamma_s - v(M,s)}{|S| - 1} \right)
\end{aligned}$$

We write $\delta = \frac{\gamma_t - v(M,t)}{\left(1 + \frac{1}{|S|-1}\right) \cdot (\gamma_t - v(M,t)) - \sum_{s \in S} \frac{\gamma_s - v(M,s)}{|S|-1}}$, so that:

$$\sum \alpha_s \cdot \gamma_s - \sum \alpha_s \cdot v(M,s) = (\gamma_t - v(M,t)) \left(1 - \frac{\varepsilon}{\delta}\right)$$

We have that δ is greater than 0 because $\gamma_t - v(M,t) < 0$ by hypothesis, and we showed that $\sum_{s \in S} \gamma_s - v(M,s) \geq 0$ so $\delta > 0$. We can then consider some ε such that $0 < \varepsilon < \min\{1, \delta\}$. We obtain that

$$\sum \alpha_s \cdot \gamma_s - \sum \alpha_s \cdot v(M,s) < (\gamma_t - v(M,t))$$

Which means that $\sum \alpha_s \cdot \gamma_s < \sum \alpha_s \cdot v(M,s)$ and contradicts that $v(M,s)$ is the solution of ϕ that minimises $\sum \alpha_s \cdot \gamma_s$.

We conclude that if vector γ satisfies equation (**??**) then $\gamma_s \geq v(M,s)$ for all states s.

Given a coalition C, a stationary strategy profile σ_C for this coalition, and a weight function w, we define a Markov decision process $\mathtt{MDP}(\mathcal{G}, \sigma_C, w)$ which represents possible executions of the game when strategies for the coalition C have been fixed. We define $\mathtt{MDP}(\mathcal{G}, \sigma_C, w) = \langle \mathsf{Stat}, \mathsf{Act}^{\mathsf{Agt} \setminus C}, P, w \rangle$ where P is such that the probability of a transition from s to s' with action a_{-C} is:

$$P(s, a_{-C}, s') = \sum_{a_C \mid \delta(s, a_C, a_{-C}) = s'} \prod_{A \in C} \sigma_A(a_A \mid h).$$

Given a strategy σ_{-C} of the coalition $\mathsf{Agt} \setminus C$ in $\mathcal{G}$, we define its projection $\pi(\sigma_{-C})$ in $\mathcal{G}(\sigma_{-C})$: for all histories h, $\pi(\sigma_{-C})(a_{-C} \mid h) = \prod_{A \in \mathsf{Agt} \setminus C} \sigma_A(a_A \mid h)$.

Lemma 12. *The expectation for* MP_w *of profile* (σ_C, σ_{-C}) *in the game* $\mathcal{G}$ *is the same as the expected value of the policy* $\pi(\sigma_{-C})$ *in* $\mathtt{MDP}(\mathcal{G}, \sigma_C, w)$.

Proof. Since the weights are given by the same function, it is enough to prove that the probability of an history in $\mathcal{G}$ knowing (σ_C, σ_{-C}) is the same as the probability in $\mathtt{MDP}(\mathcal{G}, \sigma_C, w)$ knowing σ_{-C}. We do this by induction. This is obvious for a history h of length 1. We now assume that the probability $p(h)$ of some history h is the same in $\mathcal{G}$ knowing (σ_C, σ_{-C}) and in $\mathtt{MDP}(\mathcal{G}, \sigma_C, w)$ knowing σ_{-C}. Let s be a state, the probability of $h \cdot s$ in $\mathcal{G}$ knowing (σ_C, σ_{-C}) is: $p(h) \cdot \sum_{a_{\mathsf{Agt}} | \delta(\mathtt{last}(h), a_{\mathsf{Agt}}) = s} \prod_{A \in \mathsf{Agt}} \sigma_A(a_A \mid h)$. This can be rewritten as: $p(h) \cdot \sum_{a_{\mathsf{Agt}} | \delta(\mathtt{last}(h), a_{\mathsf{Agt}}) = s} \pi(\sigma_{-C})(a_{-C} \mid h) \prod_{A \in C} \sigma_A(a_A \mid h)$ which is equal to the probability of $h \cdot s$ in $\mathtt{MDP}(\mathcal{G}, \sigma_C, w)$ knowing σ_{-C}. This shows that the expectation for w of profile (σ_C, σ_{-C}) in the game $\mathcal{G}$ is the same as the expected payoff for A of the strategy $\pi(\sigma_{-C})$ in the $\mathtt{MDP}(\mathcal{G}, \sigma_C, w)$.

Theorem 11. *Let σ_{Agt} be a stationary strategy profile, C a coalition, s a state and w a weight function. The highest expectation $\mathsf{Agt} \setminus C$ can obtain for w in $\mathcal{G}$ from s: $\sup_{\sigma_{-C}} E^{\sigma_C, \sigma_{-C}}(\mathsf{MP}_w, s)$, is the smallest γ_s component of a solution of the system of inequation:*
$\phi(\gamma, \sigma_C, w) := \bigwedge_{a_{-C} \in \mathsf{Act}^{\mathsf{Agt} \setminus C}} \bigwedge_{s \in S} \gamma_s \geq \sum_{a_C \in \mathsf{Act}^C} \gamma_{\delta(s, a_C, a_{-C})} \cdot \prod_{A \in C} \sigma_A(a_A \mid s)$
$\wedge \bigwedge_{a_{-C} \in \mathsf{Act}^{\mathsf{Agt} \setminus C}} \bigwedge_{s \in S} \gamma_s \geq w(s) - \beta_s + \sum_{a_C \in \mathsf{Act}^C} \left(\beta_{\delta(s, a_C, a_{-C})} \cdot \prod_{A \in C} \sigma_A(a_A \mid s) \right)$

Proof. We proved in Lem. **??** that the expected payoff of a profile in $\mathcal{G}$ is the same as the expected payoff of its projection in $\mathtt{MDP}(\mathcal{G}, \sigma_C, w)$. Replacing the transition relation in the equation of ϕ by its expression in $\mathtt{MDP}(\mathcal{G}, \sigma_C, w)$, and r by w, we obtain by Lem. **??** that the expected payoff of the profile is the minimal solution of:

$$\bigwedge_{a_{-C} \in \mathsf{Act}^{-C}} \bigwedge_{s \in S} \gamma_s \geq \sum_{s' \in S} \sum_{\{a_C | \delta(s, a_C, a_{-C}) = s'\}} \gamma_{s'} \cdot \prod_{A \in C} \sigma_A(a_A \mid s)$$

$$\wedge \bigwedge_{a_{-C} \in \mathsf{Act}^{-C}} \bigwedge_{s \in S} \gamma_s \geq w(s) - \beta_s + \sum_{s' \in S} \sum_{\{a_C | \delta(s, a_C, a_{-C}) = s'\}} \left(\beta_{s'} \cdot \prod_{A \in C} \sigma_A(a_A \mid s) \right)$$

We notice that $\sum_{s' \in S} \sum_{\{a_C | \delta(s, a_C, a_{-C}) = s'\}} f(s, s', a_C)$ is the same as: $\sum_{a_C \in \mathsf{Act}^C} f(s, \delta(s, a_C, a_{-C}), a_C)$, and rewrite the equation:

$$\bigwedge_{a_{-C} \in \mathsf{Act}^{-C}} \bigwedge_{s \in S} \gamma_s \geq \sum_{a_C \in \mathsf{Act}^C} \gamma_{\delta(s, a_C, a_{-C})} \cdot \prod_{A \in C} \sigma_A(a_A \mid s)$$

$$\wedge \bigwedge_{a_{-C} \in \mathsf{Act}^{-C}} \bigwedge_{s \in S} \gamma_s \geq w(s) - \beta_s + \sum_{a_C \in \mathsf{Act}^C} \left(\beta_{\delta(s, a_C, a_{-C})} \cdot \prod_{A \in C} \sigma_A(a_A \mid s) \right)$$

Therefore $\sup_{\sigma_{-C}} E^{\sigma_C, \sigma_{-C}}(\mathsf{MP}_w, s)$ equals the minimal γ_s that is part of a solution of this equation.

A.2 Proof of Thm. 4

Lemma 13. *The strategy profile σ_{Agt} is C-resilient if, and only if, it satisfies the formula:*

$$\rho(C, \sigma_{\mathsf{Agt}}) := \bigwedge_{A \in C} \exists \gamma, \gamma'. \ (\psi(\gamma, \sigma_{\mathsf{Agt}}, w_A) \wedge \phi(\gamma', \sigma_{-C}, w_A) \wedge (\gamma' \leq \gamma))$$

Proof. $\boxed{\Rightarrow}$ Assume σ_{Agt} is C-resilient, and let $A \in C$. Consider γ the payoff of A in $\mathsf{Out}_{\mathcal{G}}(\sigma_{\mathsf{Agt}})$, and γ' the highest payoff C can obtain for A against σ_{-C}: $\gamma' = \sup_{\sigma_{-C}} E^{\sigma_C,\sigma_{-C}}(\mathsf{MP}_{w_A}, s)$. By Thm. 2, γ makes $\psi(\gamma, \sigma_{\mathsf{Agt}}, w_A)$ hold, and by Thm. 3, γ' makes $\phi(\gamma', \sigma_{-C}, w_A)$ hold. Now since σ_{Agt} is resilient, C cannot improve the payoff of A and therefore $\gamma' \leq \gamma$. Hence $(\psi(\gamma, \sigma_{\mathsf{Agt}}, w_A) \wedge \phi(\gamma', \sigma_{-C}, w_A) \wedge (\gamma' \leq \gamma))$ is satisfied by this choice of γ and γ'. As we can do the same for all $A \in C$, $\rho(C, \sigma_{\mathsf{Agt}})$ holds.

$\boxed{\Leftarrow}$ The requirement that $\psi(\gamma, \sigma_{\mathsf{Agt}}, w_A)$ holds, ensures that the payoff for A of σ_{Agt} is γ (see Thm. 2). The requirement $\phi(\gamma', \sigma_{-C}, w_A)$, ensures that the maximal payoff for A that coalition C can obtain against σ_{-C} is less than γ' (see Thm. 3). Finally the requirement $\gamma' \leq \gamma$, ensures that γ' and therefore the best value C can obtain, is smaller than γ, hence the coalition C cannot improved the payoff of A by deviating. This implies the resilience of σ_{Agt}.

Lemma 14. *The strategy profile σ_{Agt} is C, r-immune if, and only if, it satisfies equation:*

$$\iota(C, \sigma_{\mathsf{Agt}}) := \bigwedge_{A \notin C} \exists \gamma, \gamma'.\ (\psi(\gamma, \sigma_{\mathsf{Agt}}, A) \wedge \phi(\gamma', \sigma_{-C}, -w_A) \wedge \gamma - r \leq -\gamma')$$

Proof. $\boxed{\Rightarrow}$ Assume σ_{Agt} is C, r-immune, and let $A \notin C$. Consider γ the payoff of A in $\mathsf{Out}_{\mathcal{G}}(\sigma_{\mathsf{Agt}})$. By Thm. 2, $\psi(\gamma, \sigma_{\mathsf{Agt}}, A)$ holds. Consider also γ' the negation of lowest payoff C can obtain for A against σ_{-C}: $\gamma' = -\inf_{\sigma_{-C}} E^{\sigma_C,\sigma_{-C}}(\mathsf{MP}_{w_A}, s)$. We use the fact that the optimal values for limit inferior and superior coincide in MDPs (see for instance [16, Thm. 9.1.3] to get the following

$$\begin{aligned}
-\inf_{\sigma_{-C}} E^{\sigma_C,\sigma_{-C}}(\mathsf{MP}_{w_A}, s) &= \sup_{\sigma_{-C}} -E^{\sigma_C,\sigma_{-C}}(\mathsf{MP}_{w_A}, s) \\
&= \sup_{\sigma_{-C}} E^{\sigma_C,\sigma_{-C}}(-\mathsf{MP}_{w_A}, s) \\
&= \sup_{\sigma_{-C}} E^{\sigma_C,\sigma_{-C}}\left(\rho \mapsto \limsup_{n\to\infty} \frac{1}{n} \sum_{1\leq k\leq n} -w_A(\rho_k), s\right) \\
&= \sup_{\sigma_{-C}} E^{\sigma_C,\sigma_{-C}}(\mathsf{MP}_{-w_A}, s) && \text{(consequence of [16])}
\end{aligned}$$

By Thm. 3, γ' makes $\phi(\gamma', \sigma_{-C}, -w_A)$ hold.

Now since σ_{Agt} is C, r-immune, the deviation of C cannot make the payoff of $A \notin C$ decrease by more than r. Therefore $\inf_{\sigma_{-C}} E^{\sigma_C,\sigma_{-C}}(\mathsf{MP}_{w_A}, s) \geq \gamma - r$ and $-\gamma' \geq \gamma - r$.

$\boxed{\Leftarrow}$ The requirement that $\psi(\gamma, \sigma_{\mathsf{Agt}}, A)$ holds, ensures that the payoff for A of σ_{Agt} is γ (see Thm. 2). The requirement $\phi(\gamma', \sigma_{-C}, -w_A)$, ensures that $\gamma' \geq \sup_{\sigma_{-C}} E^{\sigma_C,\sigma_{-C}}(\mathsf{MP}_{-w_A}, s)$ (see Thm. 3). As we so in the proof of implication, $\sup_{\sigma_{-C}} E^{\sigma_C,\sigma_{-C}}(\mathsf{MP}_{-w_A}, s) = -\inf_{\sigma_{-C}} E^{\sigma_C,\sigma_{-C}}(\mathsf{MP}_{w_A}, s)$, thus $-\gamma' \leq \inf_{\sigma_{-C}} E^{\sigma_C,\sigma_{-C}}(\mathsf{MP}_{w_A}, s)$. Therefore coalition C against σ_{-C} cannot make the payoff of A lower than $-\gamma'$, so with the additional constraint that $\gamma - r \leq -\gamma'$,

the payoff of A cannot be decreased by more than r by a deviation of C. This implies the immunity of σ_{Agt}.

Theorem 12. *There is a robust equilibria if, and only if, the following equation is satisfiable:*

$$\exists \tau \in \mathbb{R}^{\mathsf{Agt}\times\mathsf{Stat}\times\mathsf{Act}}.\ \mu(\tau) \wedge \bigwedge_{C\subseteq\mathsf{Agt}||C|\leq k} \rho(C,\tau) \wedge \bigwedge_{C\subseteq\mathsf{Agt}||C|\leq t} \iota(C,\tau)$$

Proof. The formula $\mu(\tau)$ that for each $A \in \mathsf{Agt}$ the mapping $s, a \mapsto \tau_{A,s,a}$ described by τ corresponds to a stationary strategy (see Lem. 1). Formula $\rho(C,\tau)$, says that for each coalition C of size smaller than k, the profile described by τ is C-resilient (see Lem. **??**), hence it is k-resilient. Finally $\iota(C,\tau)$, says that for each coalition C of size smaller than t, the profile described by τ is (C,r)-immune (see Lem. **??**), hence it is (t,r)-immune. This is therefore equivalent to the (k,t,r)-robustness of the strategy profile described by τ.

B Appendix for Section 4

Lemma 15. *(Lem. 2 in the body of the paper) Let ρ be a play, σ_{Agt} a strategy profile and $C \subseteq \mathsf{Agt}$ a coalition. Coalition C contains $\mathsf{Dev}(\rho, \sigma_{\mathsf{Agt}})$ if, and only if, there exists σ'_C such that $\rho \in \mathsf{Out}_{\mathcal{G}}(\rho_0, \sigma'_C, \sigma_{-C})$.*

Proof. $\boxed{\Rightarrow}$ Let ρ be a play and C a coalition which contains $\mathsf{Dev}(\rho, \sigma_{\mathsf{Agt}})$. We define σ'_C to be such that for all i, $\sigma'_C(\rho_{\leq i}) = (\mathsf{move}_i(\rho))_C$. We have that for all indices i, $\mathsf{Dev}(\rho_{i+1}, \sigma_{\mathsf{Agt}}(\mathsf{move}_i(\rho))) \subseteq C$. Therefore for all agents $A \notin C$, $\sigma_A(\rho_{\leq i}) = (\mathsf{move}_i(\rho))_A$. Then $\mathsf{Tab}(\rho_i, \sigma'_C(\rho_{\leq i}), \sigma_{-C}(\rho_{\leq i})) = \rho_{i+1}$. Hence ρ is the outcome of the profile (σ_{-C}, σ'_C).

$\boxed{\Leftarrow}$ Let σ_{Agt} be a strategy profile, σ'_C a strategy for coalition C, and $\rho \in \mathsf{Out}_{\mathcal{G}}(\rho_0, \sigma_{-C}, \sigma'_C)$. We have for all indices i that $\mathsf{move}_i(\rho) = (\sigma_{-C}(\rho_{\leq i}), \sigma'_C(\rho_{\leq i}))$. Therefore for all agents $A \notin C$, $(\mathsf{move}_i(\rho))_A = \sigma_A(\rho_{\leq i})$. Then $\mathsf{Dev}(\mathsf{move}_i(\rho), \sigma_{\mathsf{Agt}}(\rho_{\leq i})) \subseteq C$. Hence $\mathsf{Dev}(\rho, \sigma_{\mathsf{Agt}}) \subseteq C$.

Lemma 16. *(Lem. 3 in the body of the paper) Let $\mathcal{G}$ be a game and σ_{Agt} be a strategy profile and $\sigma_\exists = \kappa(\sigma_{\mathsf{Agt}})$ the associated strategy in the deviator game.*

1. *If $\rho \in \mathsf{Out}_{\mathcal{D}(\mathcal{G})}(\sigma_\exists)$, then $\mathsf{Dev}(\pi_{\mathsf{Out}}(\rho), \sigma_{\mathsf{Agt}}) = \delta(\rho)$.*
2. *If $\rho \in \mathsf{Out}_{\mathcal{G}}$ and $\rho' = ((\rho_i, \mathsf{Dev}(\rho_{\leq i}, \sigma_{\mathsf{Agt}})) \cdot (\sigma_{\mathsf{Agt}}(\rho_{\leq i}), \mathsf{move}_i(\rho)))_{i\in\mathbb{N}}$ then $\rho' \in \mathsf{Out}_{\mathcal{D}(\mathcal{G})}(\sigma_\exists)$*

Proof (Proof of 1). We prove that for all i, $\mathsf{Dev}(\pi_{\mathsf{Out}}(\rho)_{\leq i}, \sigma_{\mathsf{Agt}}) = \pi_{\mathsf{Dev}}(\rho_{\leq i})$, which implies the property. The property holds for $i = 0$, since initially both

sets are empty. Assume now that it holds for $i \geq 0$.

$$
\begin{aligned}
&\mathsf{Dev}(\pi_{\mathsf{Out}}(\rho)_{\leq i+1}, \sigma_{\mathsf{Agt}}) \\
&= \mathsf{Dev}(\pi_{\mathsf{Out}}(\rho)_{\leq i}, \sigma_{\mathsf{Agt}}) \cup \mathsf{Dev}(\sigma_{\mathsf{Agt}}(\pi_{\mathsf{Out}}(\rho)_{\leq i}), \pi_{\mathsf{Act}}(\mathsf{move}_{i+1}(\rho))) && \text{(by definition of deviators)} \\
&= \pi_{\mathsf{Dev}}(\rho_{\leq i}) \cup \mathsf{Dev}(\sigma_{\mathsf{Agt}}(\pi_{\mathsf{Act}}(\rho)_{\leq i}), \pi_{\mathsf{Act}}(\mathsf{move}_{i+1}(\rho))) && \text{(by induction hypothesis)} \\
&= \pi_{\mathsf{Dev}}(\rho_{\leq i}) \cup \mathsf{Dev}(\sigma_{\exists}(\rho_{\leq i}), \pi_{\mathsf{Act}}(\mathsf{move}_{i+1}(\rho))) && \text{(by definition of } \sigma_{\exists}) \\
&= \pi_{\mathsf{Dev}}(\rho_{\leq i}) \cup \mathsf{Dev}(\mathsf{move}_{i+1}(\rho)) && \text{(by assumption } \rho \in \mathsf{Out}_{\mathcal{D}(\mathcal{G})}(\sigma_{\exists})) \\
&= \pi_{\mathsf{Dev}}(\rho_{\leq i+1}) && \text{(by construction of } \mathcal{D}(\mathcal{G}))
\end{aligned}
$$

Which concludes the induction.

Proof (Proof of 2). The property is shown by induction. It holds for the initial state. Assume it is true until index i, then

$$
\begin{aligned}
&\mathsf{Tab}'(\rho'_i, \sigma_{\exists}(\rho'_{\leq i}), \mathsf{move}_i(\rho)) \\
&\quad = \mathsf{Tab}'((\rho_i, \mathsf{Dev}(\rho_{\leq i}, \sigma_{\mathsf{Agt}})), \sigma_{\exists}(\rho'_{\leq i}), \mathsf{move}_i(\rho)) && \text{(by definition of } \rho') \\
&\quad = (\mathsf{Tab}(\rho_i, \mathsf{move}_i(\rho)), \mathsf{Dev}(\rho_{\leq i}, \sigma_{\mathsf{Agt}}) \cup \mathsf{Dev}(\sigma_{\exists}(\rho'_{\leq i}), \rho_{i+1})) && \text{(by construction of } \mathsf{Tab}') \\
&\quad = (\rho_{i+1}, \mathsf{Dev}(\rho_{\leq i}, \sigma_{\mathsf{Agt}}) \cup \mathsf{Dev}(\sigma_{\exists}(\rho'_{\leq i}), \rho_{i+1})) && \text{(since } \rho \text{ is an outcome of the game)} \\
&\quad = (\rho_{i+1}, \mathsf{Dev}(\rho_{\leq i}, \sigma_{\mathsf{Agt}}) \cup \mathsf{Dev}(\sigma_{\mathsf{Agt}}(\rho_{\leq i}), \rho_{i+1})) && \text{(by construction of } \sigma_{\exists}) \\
&\quad = (\rho_{i+1}, \mathsf{Dev}(\rho_{\leq i+1}, \sigma_{\mathsf{Agt}})) && \text{(by definition of deviators)} \\
&\quad = \rho'_{i+1}
\end{aligned}
$$

This shows that ρ' is an outcome of $\sigma_{\exists}$.

Lemma 17. *Let $C \subseteq \mathsf{Agt}$ be a coalition, σ_{Agt} be a strategy profile, $G \subseteq \mathbb{R}$ and A a player. We have that for all strategies σ'_C for coalition C, $\mathsf{payoff}_A(\sigma_{-C}, \sigma'_C) \in G$ if, and only if, $\kappa(\sigma_{\mathsf{Agt}})$ is winning in $\mathcal{D}(\mathcal{G})$ for objective $\Omega(C, A, G) = \{\rho \mid \delta(\rho) \subseteq C \Rightarrow \mathsf{payoff}_A(\pi_{\mathsf{Out}}(\rho)) \in G\}$.*

Proof. $\boxed{\Rightarrow}$ Let ρ be an outcome of $\sigma_{\exists} = \kappa(\sigma_{\mathsf{Agt}})$. By Lem. 3, we have that $\delta(\rho) = \mathsf{Dev}(\pi_{\mathsf{Out}}(\rho), \sigma_{\mathsf{Agt}})$. By Lem. 2, $\pi_{\mathsf{Out}}(\rho)$ is the outcome of $(\sigma_{-\delta(\rho)}, \sigma'_{\delta(\rho)})$ for some $\sigma'_{\delta(\rho)}$. If $\delta(\rho) \subseteq C$, then $\mathsf{payoff}_A(\pi_{\mathsf{Out}}(\rho)) = \mathsf{payoff}_A(\sigma_{-C}, \sigma_{C \setminus \delta(\rho)}, \sigma'_{\delta(\rho)}) = \mathsf{payoff}_A(\sigma_{-C}, \sigma''_C)$ where $\sigma''_A = \sigma'_A$ if $A \in \delta(\rho)$ and σ_A otherwise. By hypothesis, this payoff belongs to G. This holds for all outcomes ρ of $\sigma_{\exists}$, thus $\sigma_{\exists}$ is a winning strategy for $\Omega(C, A, G)$.

$\boxed{\Leftarrow}$ Assume $\sigma_{\exists} = \kappa(\sigma_{\mathsf{Agt}})$ is a winning strategy in $\mathcal{D}(\mathcal{G})$ for $\Omega(C, A, G)$. Let σ'_C be a strategy for C and ρ the outcome of (σ'_C, σ_{-C}). By Lem. 2, $\mathsf{Dev}(\rho, \sigma_{\mathsf{Agt}}) \subseteq C$. By Lem. 3, $\rho' = (\rho_j, \mathsf{Dev}(\rho_{\leq j}, \sigma_{\mathsf{Agt}}))_{j \in \mathbb{N}}$ is an outcome of $\sigma_{\exists}$. We have that $\delta(\rho') = \mathsf{Dev}(\rho, \sigma_{\mathsf{Agt}}) \subseteq C$. Since $\sigma_{\exists}$ is winning, ρ is such that $\mathsf{payoff}_A(\pi_{\mathsf{Out}}(\rho)) \in G$. Since $\mathsf{payoff}_A(\pi_{\mathsf{Stat}}(\rho')) = \mathsf{payoff}_A(\rho)$, this shows that for all strategies σ'_C, $\mathsf{payoff}_A(\sigma_{-C}, \sigma'_C) \in G$

B.1 Proof of Thm. 6

The proof of the theorem relies on the two following lemmas. The first one shows the correctness of the resilience objective. The second lemma shows the correctness of the immunity objective, its proof follows the same ideas than the first and can be found in the appendix.

Lemma 18. *Let $\mathcal{G}$ be a concurrent game and σ_{Agt} a strategy profile in $\mathcal{G}$. The strategy profile σ_{Agt} is k-resilient if, and only if, strategy $\kappa(\sigma_{\mathsf{Agt}})$ is winning in $\mathcal{D}(\mathcal{G})$ for objective $\mathcal{R}e(k,p)$ where $p = \mathsf{payoff}(\sigma_{\mathsf{Agt}})$.*

Proof. By Lem. 4, σ_{Agt} is k-resilient if, and only if, for each coalition C of size smaller than k, and each player A in C, $\kappa(\sigma_{\mathsf{Agt}})$ is winning for $\Omega(C, A,]-\infty, \mathsf{payoff}_A(\sigma_{\mathsf{Agt}})])$. We will thus in fact show that for each coalition C of size smaller than k, and each player A in C, $\kappa(\sigma_{\mathsf{Agt}})$ is winning for $\Omega(C, A,]-\infty, \mathsf{payoff}_A(\sigma_{\mathsf{Agt}})])$ if, and only if, $\kappa(\sigma_{\mathsf{Agt}})$ is winning for $\mathcal{R}e(k,p)$.

$\boxed{\Rightarrow}$ Let ρ be an outcome of $\kappa(\sigma_{\mathsf{Agt}})$.

- If $|\delta(\rho)| > k$, then ρ is in $\mathcal{R}e(k,p)$ by definition.
- If $|\delta(\rho)| = k$, then for all $A \in \delta(\rho)$, $\mathsf{payoff}_A(\pi_{\mathsf{Out}}(\rho)) \in]-\infty, p(A)]$ because $\kappa(\sigma_{\mathsf{Agt}})$ is winning for $\Omega(\delta(\rho), A,]-\infty, p(A)])$. Therefore ρ is in $\mathcal{R}e(k,p)$.
- If $|\delta(\rho)| < k$, then for all $A \in \mathsf{Agt}$, $C = \delta(\rho) \cup \{A\}$ is a coalition of size smaller than k, and $\mathsf{payoff}_A(\pi_{\mathsf{Out}}(\rho)) \in]-\infty, p(A)]$ because $\kappa(\sigma_{\mathsf{Agt}})$ is winning for $\Omega(C, A,]-\infty, p(A)])$. Therefore ρ is in $\mathcal{R}e(k,p)$.

This holds for all outcomes ρ of $\kappa(\sigma_{\mathsf{Agt}})$ and shows that $\kappa(\sigma_{\mathsf{Agt}})$ is winning for $\mathcal{R}e(k,p)$.

$\boxed{\Leftarrow}$ We now show that $\kappa(\sigma_{\mathsf{Agt}})$ is winning for $\Omega(C, A,]-\infty, p(A)])$ for each coalition C of size smaller or equal to k and player A in C. Let ρ be an outcome of $\kappa(\sigma_{\mathsf{Agt}})$. Let p be such that strategy $\kappa(\sigma_{\mathsf{Agt}})$ is winning for $\mathcal{R}e(k,p)$. We have $\rho \in \mathcal{R}e(k,p)$. We show that ρ belongs to $\Omega(C, A,]-\infty, p(A)])$:

- If $\delta(\rho) \not\subseteq C$, then $\rho \in \Omega(C, A,]-\infty, p(A)])$ by definition.
- If $\delta(\rho) \subseteq C$ and $|\delta(\rho)| = k$, then $\mathsf{Dev}(\rho) = C$. Since $\rho \in \mathcal{R}e(k,p)$, for all $A \in C$, $\mathsf{payoff}_A(\rho) \leq p(A)$ and therefore $\mathsf{payoff}_A(\rho) \in]-\infty, p(A)]$. Hence $\rho \in \Omega(C, A,]-\infty, p(A)])$.
- If $\delta(\rho) \subseteq C$ and $|\delta(\rho)| < k$, then since $\rho \in \mathcal{R}e(k,p)$, for all $A \in \mathsf{Agt}$, $\mathsf{payoff}_A(\rho) \leq p(A)$. Therefore $\rho \in \Omega(C, A,]-\infty, p(A)])$.

This holds for all outcomes ρ of $\kappa(\sigma_{\mathsf{Agt}})$ and shows it is winning for $\Omega(C, A,]-\infty, p(A)])$ for each coalition C and player A in C, which shows that σ_{Agt} is k-resilient.

Lemma 19. *Let $\mathcal{G}$ be a concurrent game and σ_{Agt} a strategy profile in $\mathcal{G}$. The strategy profile σ_{Agt} is (t,r)-immune if, and only if, strategy $\kappa(\sigma_{\mathsf{Agt}})$ is winning for objective $\mathcal{I}(t,r,p)$ where $p = \mathsf{payoff}(\sigma_{\mathsf{Agt}})$.*

Proof. By Lem. 4, σ_{Agt} is (t,r)-immune if, and only if, for each coalition C of size smaller than t, and each player A not in C, $\kappa(\sigma_{\mathsf{Agt}})$ is winning for $\Omega(C, A, [\mathsf{payoff}_A(\sigma_{\mathsf{Agt}}) - r, +\infty[)$. We will thus in fact show that for each coalition C of size smaller than t, and each player A not in C, $\kappa(\sigma_{\mathsf{Agt}})$ is winning for $\Omega(C, A, [\mathsf{payoff}_A(\sigma_{\mathsf{Agt}}) - r, +\infty[)$ if, and only if, $\kappa(\sigma_{\mathsf{Agt}})$ is winning for $\mathcal{R}e(t, r, p)$.

$\boxed{\Rightarrow}$ Let ρ be an outcome of $\kappa(\sigma_{\mathsf{Agt}})$.

- If $|\delta(\rho)| > t$, then ρ is in $\mathcal{I}(t, r, p)$ by definition.
- If $|\delta(\rho)| \leq t$, then $C = \delta(\rho)$ is a coalition of size smaller than t. As a consequence, for all $A \notin \delta(\rho)$, ρ is winning for $\Omega(C, A, [p_A - r, +\infty[)$. By definition of Ω, we have $\mathsf{payoff}_A(\rho) \geq p_A - r$. Thus ρ is in $\mathcal{I}(t, r, p)$.

$\boxed{\Leftarrow}$ We now show that $\kappa(\sigma_{\mathsf{Agt}})$ is winning for $\Omega(C, A, [\mathsf{payoff}_A(\sigma_{\mathsf{Agt}}) - r, +\infty[)$ for each coalition C of size smaller than t and player A not in C. Let ρ be an outcome of $\kappa(\sigma_{\mathsf{Agt}})$. Let p be such that strategy $\kappa(\sigma_{\mathsf{Agt}})$ is winning for $\mathcal{I}(t, r, p)$. We have $\rho \in \mathcal{I}(t, r, p)$. We show that ρ belongs to $\Omega(C, A, [p(A) - r, +\infty[)$:

- If $\delta(\rho) \not\subseteq C$, then $\rho \in \Omega(C, A, [p(A) - r, +\infty[)$ by definition.
- If $\delta(\rho) \subseteq C$, then since $\rho \in \mathcal{I}(t, r, p)$, for all $A \notin C$, $p(A) - r \leq \mathsf{payoff}_A(\pi_{\mathsf{Out}}(\rho))$. Therefore $\mathsf{payoff}_A(\rho) \in [p(A) - r, +\infty[$ and $\rho \in \Omega(C, A, [p(A) - r, +\infty[)$.

This holds for all outcomes ρ of $\kappa(\sigma_{\mathsf{Agt}})$ and shows it is winning for $\Omega(C, A, [p(A) - r, +\infty[)$ for each coalition C and player A in C, which shows that σ_{Agt} is (t, r)-immune.

Lemma 20. *Let $\mathcal{G}$ be a concurrent game and σ_{Agt} a strategy profile in $\mathcal{G}$. The strategy profile σ_{Agt} is a (k, t, r)-robust profile in $\mathcal{G}$ if, and only if, the associated strategy of $\mathtt{Eve}$ is winning for the objective $\mathcal{R}(k, t, r, \mathsf{Out}(\sigma_{\mathsf{Agt}})) = \mathcal{R}e(k, p) \cap \mathcal{I}(t, r, p)$ where $p = \mathsf{payoff}(\sigma_{\mathsf{Agt}})$.*

Proof. This is a simple consequence of Lem. **??** and Lem. **??**. Let σ_{Agt} be a (k, t, r)-robust strategy profile. It is k-resilient, so $\kappa(\sigma_{\mathsf{Agt}})$ is winning the resilience objective. It is also (t, r)-immune, so $\kappa(\sigma_{\mathsf{Agt}})$ is winning the immunity objective. Therefore any outcome of σ_{Agt} is in the intersection, and σ_{Agt} ensures the robustness objective.

In the other direction, assume $\kappa(\sigma_{\mathsf{Agt}})$ wins the robustness objective. Then σ_{Agt} wins both the k-resilience objective and the (t, r)-immunity objective. Using lemmas **??** and **??**, σ_{Agt} is k-resilient and (t, r)-immune; it is therefore (k, t, r)-robust.

C Appendix for Section 5

C.1 Proof of Lem. 5

Lemma 21. *Let ρ be a play. It satisfies objective $\delta(\rho) = \varnothing \Rightarrow \mathsf{MP}_{A_i}(\rho) = p_i$ if, and only if, $\mathsf{MP}_{2 \cdot v_{|\mathsf{Agt}|+i}}(\rho) \geq p(A_i)$ and $\overline{\mathsf{MP}}_{3 \cdot v_{|\mathsf{Agt}|+i}}(\rho) \geq -p(A_i)$.*

Proof. We distinguish two cases according to whether $\delta(\rho)$ is empty.

– If $\delta(\rho) \neq \varnothing$, the implication holds and we have that after some point in the execution $\pi_{\mathsf{Dev}}(\rho) \neq \varnothing$. By item 7 of the definition of v, the average weight on dimensions $2 \cdot |\mathsf{Agt}| + i$ and $3 \cdot |\mathsf{Agt}| + i$ will tend to W, which is greater than $p(A_i)$ and $-p(A_i)$. Therefore the equivalence holds.
– If $\delta(\rho) = \varnothing$, then along all the run the D component is empty. By item 6 of the definition of v, $\mathsf{MP}_{v_{2\cdot|\mathsf{Agt}|+i}}(\rho) = \mathsf{MP}_{w_{A_i}}(\rho)$ and $\overline{\mathsf{MP}}_{v_{3\cdot|\mathsf{Agt}|+i}}(\rho) = -\mathsf{MP}_{w_{A_i}}(\rho)$. Therefore $\mathsf{MP}_{A_i}(\rho) = p_i$ is equivalent to the fact $\mathsf{MP}_{2\cdot v_{|\mathsf{Agt}|+i}}(\rho) \geq p(A_i)$ and $\overline{\mathsf{MP}}_{3\cdot v_{|\mathsf{Agt}|+i}}(\rho) \geq -p(A_i)$.

Lemma 22. *If ρ is an outcome of $\kappa(\sigma_{\mathsf{Agt}})$ with $\mathsf{payoff}(\sigma_{\mathsf{Agt}}) = p$, then play ρ satisfies objective $\mathcal{R}e(k,p)$ if, and only if, for all agents A_i, $\overline{\mathsf{MP}}_{v_{|\mathsf{Agt}|+i}}(\rho) \geq -p(A_i)$.*

Proof. First notice the following equivalence:

$$\mathsf{payoff}_{A_i}(\rho) \leq p(A_i) \Leftrightarrow \liminf \frac{w_i(\rho_{\leq n})}{n} \leq p(A_i) \Leftrightarrow \limsup -\frac{w_i(\rho_{\leq n})}{n} \geq -p(A_i)$$

$\boxed{\Rightarrow}$ Let A_i be a player, and assume $\rho \in \mathcal{R}e(k,p)$. We distinguish three cases based on the size of $\delta(\rho)$:

– If $|\delta(\rho)| < k$ then for all indices j, $|\mathsf{Dev}(\rho_{\leq j'})| < k$. Therefore $v_{|\mathsf{Agt}|+i}(\rho_j) = -w_{A_i}(\rho_j)$ (item 3 of the definition). Then as ρ is in $\mathcal{R}e(k,p)$, $\mathsf{payoff}_{A_i}(\rho) \leq p(A_i)$ and therefore $\overline{\mathsf{MP}}_{v_{|\mathsf{Agt}|+i}}(\rho) \geq -p(A_i)$
– If $|\delta(\rho)| = k$, then we distinguish two cases:
  • If $A_i \notin \delta(\rho)$, then there is a j such that for all $j' \geq j$, $|\mathsf{Dev}(\rho_{\leq j})| = k$ and $A_i \notin \mathsf{Dev}(\rho_j)$. Therefore for all $j' \geq j$ we have that $v_{|\mathsf{Agt}|+i}(\rho_{j'}) = W$ (item 5 of the definition). Since $p(A_i) \geq -W$, $\overline{\mathsf{MP}}_{v_{|\mathsf{Agt}|+i}}(\rho) \geq -p(A_i)$.
  • Otherwise $A_i \in \delta(\rho)$, then there is a j such that for all $j' \geq j$, $A_i \in \mathsf{Dev}(\rho_{\leq j'})$. Therefore for all $j' \geq j$ we have that $v_{|\mathsf{Agt}|+i}(\rho_{j'}) = -w_{A_i}(\rho_{j'})$ (item 4 of the definition). Then $\overline{\mathsf{MP}}_{v_{|\mathsf{Agt}|+i}}(\rho) = \limsup -\frac{w_i(\rho_{\leq n})}{n}$. Then as ρ satisfies $\mathcal{R}e(k,p)$ $\mathsf{payoff}_{A_i}(\rho) \leq p(A_i)$ and therefore using the equivalence at the beginning of this proof $\overline{\mathsf{MP}}_{v_{|\mathsf{Agt}|+i}}(\rho) \geq -p(A_i)$.
– Otherwise $|\delta(\rho)| > k$. Then, there is some index j such that either $|\mathsf{Dev}(\rho_{\leq j})| > k$ or $|\mathsf{Dev}(\rho_{\leq j})| = k \wedge A \notin \mathsf{Dev}(\rho_{\leq j})$. Then, by monotonicity of Dev along ρ, for all $j' \geq j$, $v_{|\mathsf{Agt}|+i}(\rho_{j'}) = W$ (item 5 of the definition). Since $p(A_i) \geq -W$, $\overline{\mathsf{MP}}_{v_{|\mathsf{Agt}|+i}}(\rho) \geq -p(A_i)$.

$\boxed{\Leftarrow}$ Now assume that for all players A_i, $\overline{\mathsf{MP}}_{v_{|\mathsf{Agt}|+i}}(\rho) \geq -p(A_i)$.

– If $|\delta(\rho)| < k$, therefore for all i and j we have that $v_{|\mathsf{Agt}|+i}(\rho_j) = -w_{A_i}(\rho_j)$ then $\overline{\mathsf{MP}}_{v_{|\mathsf{Agt}|+i}}(\rho) = \limsup -\frac{w_i(\rho_{\leq n})}{n}$. Thus using the equivalence at the beginning of this proof $\mathsf{payoff}_{A_i}(\rho) \leq p(A_i)$ for all A_i.
– If $|\delta(\rho)| = k$. Let A_i be a player in $\delta(\rho)$. Then for all j, either $|\mathsf{Dev}(\rho_{\leq j})| < k$ or $A_i \in \mathsf{Dev}(\rho_{\leq j})$. Therefore for all j we have that $v_{|\mathsf{Agt}|+i}(\rho_j) = -w_{A_i}(\rho_j)$ then $\overline{\mathsf{MP}}_{v_{|\mathsf{Agt}|+i}}(\rho) = \limsup -\frac{w_i(\rho_{\leq n})}{n}$. Thus using the equivalence at the beginning of this proof $\mathsf{payoff}_{A_i}(\rho) \leq p(A_i)$. This being true for all players in $\delta(\rho)$ shows that ρ belongs to $\mathcal{R}e(k,p)$.

– Otherwise $|\delta(\rho)| > k$ and then $\rho \in \mathcal{R}e(k,p)$ by definition of $\mathcal{R}e(k,p)$.

We now show the immunity part.

Lemma 23. *If ρ is an outcome of $\kappa(\sigma_{\mathsf{Agt}})$ with $\mathsf{payoff}(\sigma_{\mathsf{Agt}}) = p$, then play ρ satisfies objective $\mathcal{I}(t,r,p)$ if, and only if, for all agents A_i, $\mathsf{MP}_{v_i}(\rho) \geq p(A_i) - r$.*

Proof. $\boxed{\Rightarrow}$ Let A_i be a player and assume $\rho \in \mathcal{I}(t,r,p)$. We distinguish two cases:

– If $|\delta(\rho)| \leq t \wedge A_i \notin \delta(\rho)$, then for all indices j, $v_i(\rho_j) = w_{A_i}(\rho_j)$. Therefore $\mathsf{MP}_{v_i}(\rho) = \mathsf{payoff}_{A_i}(\pi_{\mathsf{Stat}}(\rho))$. Then as ρ satisfies $\mathcal{I}(t,r,p)$, $p(A_i) - r \leq \mathsf{payoff}_{A_i}(\pi_{\mathsf{Stat}}(\rho)) = \mathsf{MP}_{v_i}(\rho)$.
– Otherwise there is some index j such that either $|\mathsf{Dev}(\rho_{\leq j})| > t$ or $A \in \mathsf{Dev}(\rho_{\leq j})$. Then, by monotonicity of Dev along ρ, for all $j' \geq j$, $v_i(\rho_{j'}) = W \geq p(A_i)$. Hence $\mathsf{MP}_{v_i}(\rho) \geq p(A_i)$.

$\boxed{\Leftarrow}$ Assume that for all players A_i, $\mathsf{MP}_{v_i}(\rho) \geq p(A_i) - r$.

– If $|\delta(\rho)| \leq t$. Let $A_i \notin \delta(\rho)$, then for all j, $|\mathsf{Dev}(\rho_{\leq j})| \leq t$ and $A \notin \mathsf{Dev}(\rho_{\leq j})$. Therefore for all j we have that $v_i(\rho_j) = w_{A_i}(\rho_j)$ and thus $\mathsf{MP}_{v_i}(\rho) \geq p(A_i) - r \Leftrightarrow \mathsf{payoff}_{A_i}(\rho) - r \leq \mathsf{payoff}_{A_i}(\rho)$. This shows that ρ belongs to $\mathcal{I}(t,r,p)$.
– Otherwise $\delta(\rho)| > t$ and ρ belongs to $\mathcal{I}(t,r,p)$ by definition of $\mathcal{I}(t,r,p)$.

We now join the two preceding result to talk about the robustness objective.

Lemma 24. *If ρ is an outcome of $\kappa(\sigma_{\mathsf{Agt}})$ with $\mathsf{payoff}(\sigma_{\mathsf{Agt}}) = p$, then play ρ satisfies objective $\mathcal{R}(k,t,r,p)$ if, and only if, for all agents A_i, $\mathsf{MP}_{v_i}(\rho) \geq p(A_i) - r$ and $\overline{\mathsf{MP}}_{v_{|\mathsf{Agt}|+i}}(\rho) \geq -p(A_i)$.*

Proof.

$$\begin{aligned}
\rho \in \mathcal{R}(k,t,r,p) \Leftrightarrow & \rho \text{ satisfies } \mathcal{R}e(k,p) \text{ and } \mathcal{I}(t,r,p) \quad \text{(By definition of } \mathcal{R}(k,t,r,p)) \\
\Leftrightarrow & \rho \in \mathcal{R}e(k,p) \text{ and } \forall A_i \in \mathsf{Agt}.\ \mathsf{MP}_{v_i}(\rho) \geq p(A_i) - r \quad \text{(By Lem. ??)} \\
\Leftrightarrow & \forall A_i \in \mathsf{Agt}.\ \overline{\mathsf{MP}}_{v_{|\mathsf{Agt}|+i}}(\rho) \geq -p(A_i) \\
& \quad \text{and } \forall A_i \in \mathsf{Agt}.\ \mathsf{MP}_{v_i}(\rho) \geq p(A_i) - r \quad \text{(By Lem. ??)}
\end{aligned}$$

Lemma 25. *(Lem. 6 in the body of the paper) Let $\mathcal{G}$ be a concurrent game with mean-payoff objectives. There is a (k,t,r)-robust equilibrium in $\mathcal{G}$ if, and only if, for the multidimensional mean-payoff objective given by v, $I = [\![1, |\mathsf{Agt}|]\!] \cup [\![2 \cdot |\mathsf{Agt}| + 1, 3 \cdot |\mathsf{Agt}|]\!]$ and $J = [\![|\mathsf{Agt}| + 1, 2 \cdot |\mathsf{Agt}|]\!] \cup [\![3 \cdot |\mathsf{Agt}| + 1, 4 \cdot |\mathsf{Agt}|]\!]$, there is a payoff vector p such that **Eve** can ensure threshold u in $\mathcal{D}(\mathcal{G})$, where for all $i \in [\![1, |\mathsf{Agt}|]\!]$, $u_i = p(A_i) - r$, $u_{|\mathsf{Agt}|+i} = -p(A_i)$, $u_{2\cdot|\mathsf{Agt}|+i} = p(A_i)$, and $u_{3\cdot|\mathsf{Agt}|+i} = -p(A_i)$.*

Proof. $\boxed{\Rightarrow}$ Let σ_{Agt} be a robust equilibrium, using Thm. 6 $\kappa(\sigma_{\mathsf{Agt}})$ is a strategy of $\mathtt{Eve}$ in $\mathcal{D}(\mathcal{G})$ which ensures $\mathcal{R}(k,t,r,p)$ where $p = \mathsf{payoff}(\mathsf{Out}(\sigma_{\mathsf{Agt}}))$. Let ρ be an outcome of $\kappa(\sigma_{\mathsf{Agt}})$ in $\mathcal{D}(\mathcal{G})$. We will show that it is above the threshold u in all dimensions.

We first show that $\delta(\rho) = \varnothing \implies \mathsf{MP}_{A_i}(\rho) = p_i$. If $\delta(\rho) \neq \varnothing$ this is trivial. Otherwise $\delta(\rho) = \varnothing$, and by Lem. 3, there is ρ' such that $\mathsf{Dev}(\rho', \sigma_{\mathsf{Agt}}) = \varnothing$ and $\rho = \pi_{\mathsf{Out}}(\rho')$. Then by Lem. 2, $\rho' = \mathsf{Out}_{\mathcal{G}}(\sigma_{\mathsf{Agt}})$. Therefore $\rho = \pi_{\mathsf{Out}}(\mathsf{Out}_{\mathcal{G}}(\sigma_{\mathsf{Agt}}))$, thus $\mathsf{MP}_{A_i}(\rho) = \mathsf{payoff}_i(\sigma_{\mathsf{Agt}}) = p_i$ and the implication holds.

Then by Lem. **??**, we have that $\mathsf{MP}_{v_{2\cdot|\mathsf{Agt}|+i}}(\rho) \geq p(A_i)$ and $\overline{\mathsf{MP}}_{v_{3\cdot|\mathsf{Agt}|+i}}(\rho) \geq -p(A_i)$. This shows we ensure the correct thresholds on dimensions in $[\![2\cdot|\mathsf{Agt}|+1, 4\cdot|\mathsf{Agt}|]\!]$. Now, by Lem. **??**, for all agents A_i, $\mathsf{MP}_{v_i}(\rho) \geq p(A_i) - r$ and $\overline{\mathsf{MP}}_{v_{|\mathsf{Agt}|+i}}(\rho) \geq -p(A_i)$. This shows we ensure the correct thresholds on dimensions in $[\![1, 2\cdot|\mathsf{Agt}|]\!]$.

$\boxed{\Leftarrow}$ In the other direction, let p be a payoff vector such that there exist a strategy $\sigma_\exists$ in $\mathcal{D}(\mathcal{G})$ that ensure the threshold u. We define a strategy profile σ_{Agt} by induction, given a history h in $\mathcal{G}$: 1. if $|h| = 1$, then $h' = (h_0, \varnothing)$; 2. otherwise we assume σ_{Agt} has already been defined for histories shorter than h, we let $h' = (h_0, \varnothing)\cdot(\sigma_{\mathsf{Agt}}(h_{\leq 0}), \mathsf{move}_0(h))\cdot((h_i, \mathsf{Dev}(h_{\leq i}, \sigma_{\mathsf{Agt}})) \cdot (\sigma_{\mathsf{Agt}}(h_{\leq i}), \mathsf{move}_i(h)))_{0<i<|h|-1}\cdot (h, \mathsf{Dev}(h, \sigma_{\mathsf{Agt}}))$. Note the similarity with the second point of Lem. 3 and the fact that $\pi_{\mathsf{Out}}(h') = h$. We then set $\sigma_{\mathsf{Agt}}(h) = \sigma_\exists(h')$.

We show that $\mathsf{payoff}(\mathsf{Out}(\sigma_{\mathsf{Agt}})) = p$. Consider the strategy $\sigma_\forall$ of $\mathtt{Adam}$ in $\mathcal{D}(\mathcal{G})$ that always plays the same move as $\mathtt{Eve}$, the outcome $\rho = \mathsf{Out}_{\mathcal{D}(\mathcal{G})}(\sigma_\exists, \sigma_\forall)$ is such that $\delta(\rho) = \varnothing$. Since $\sigma_\exists$ ensures the threshold u, using Lem. **??**, for all agents A_i, $\mathsf{MP}_{A_i}(\rho) = p(A_i)$. We now show by induction that $\pi_{\mathsf{Out}}(\rho)$ is compatible with σ_{Agt}. Let $i \in \mathbb{N}$, we assume that the property holds for prefixes of $\pi_{\mathsf{Out}}(\rho)$ of length less than i. We have that $\pi_{\mathsf{Act}}(\rho_i) = \sigma_{\mathsf{Agt}}(\pi_{\mathsf{Out}}(\rho)_{\leq i}))$ because $\mathtt{Adam}$ plays the same move than $\mathtt{Eve}$ on this path. We also have that $\pi_{\mathsf{Stat}}(\rho_{i+1}) = \mathsf{Tab}(\pi_{\mathsf{Stat}}(\rho_i), \pi_{\mathsf{Act}}(\rho_i)) = \mathsf{Tab}(\pi_{\mathsf{Stat}}(\rho_i), \sigma_{\mathsf{Agt}}(\pi_{\mathsf{Out}}(\rho)_{\leq i}))$ by construction of $\mathcal{D}(\mathcal{G})$, and therefore $\pi_{\mathsf{Out}}(\rho_{i+1})$ is compatible with σ_{Agt}. Then as ρ is the projection of the outcome of σ_{Agt}, we have that $\mathsf{payoff}(\mathsf{Out}(\sigma_{\mathsf{Agt}})) = p$.

Let ρ be an outcome of $\sigma_\exists$. By Lem. **??**, we have that ρ satisfies the objective $\mathcal{I}(t,r,p)$, and by Lem. **??** it satisfies $\mathcal{R}e(k,p)$. Using Thm. 6, strategy σ_{Agt} is an equilibrium in $\mathcal{G}$ with the payoff p.

Proof. If we apply the bound of [8, Thm. 22], to the deviator game, then this shows that if there is a solution there is one of size bounded by $d\cdot P_1(\max\{||a_i, b_i|| \mid 1 \leq i \leq k\}, P_6(||W||, ||\mathsf{Stat} \times 2^{\mathsf{Agt}}||, d), d)$ where $||x||$ represents the size of the encoding of the object x, P_1 and P_6 are two polynomial functions, $(a_i, b_i)_{1\leq i\leq k}$ are the inequations defining the polyhedron, W is the maximal constant occurring in the weights and d is the number of dimension, which in our case is equal to $4\cdot|\mathsf{Agt}|$. The size $||\mathsf{Stat} \times 2^{\mathsf{Agt}}||$ is bounded by $||\mathsf{Stat}|| + |\mathsf{Agt}|$ (using one bit for each agent in our encoding). Thus the global bound is polynomial with respect to the game $\mathcal{G}$.

Lemma 26. *(Lem. 7 in the body of the paper) If there is a solution to the polyhedron value problem in $\mathcal{D}(\mathcal{G})$ then there is one whose encoding is of polynomial size with respect to $\mathcal{G}$ and the polyhedron given as input.*

D Appendix for Section 6

Lemma 27. *(Lem. 8 in the body of the paper)* $\texttt{Eve}$ *can ensure payoff* $u \in [\![-W, W]\!]^d$ *in* $\mathcal{D}(\mathcal{G})$ *from* (s, D) *if, and only if, she can ensure* u *in the fixed coalition game* $\mathcal{F}(D, p)$ *from* (s, D).

Proof. $\boxed{\Rightarrow}$ Let $\sigma_\exists$ be strategy which ensures u in $\mathcal{D}(\mathcal{G})$ from (s, D). If we apply the strategy in $\mathcal{F}$, any of its outcome ρ from (s, D) will either stay in the D component and correspond to an outcome of $\sigma_\exists$ in $\mathcal{D}(\mathcal{G})$ or reach a state (s', D') in $\mathsf{Succ}(D)$. Since $\texttt{Eve}$ can ensure the payoff u in $\mathcal{D}(\mathcal{G})$, and (s', D') is reached by one of its outcome, she can do so from (s', D'). Therefore (s', D') is a state where the weights are maximal (equal to W on all dimensions) and the outcome is winning in $\mathcal{F}(D, u)$.

$\boxed{\Leftarrow}$ Let $\sigma_\exists$ be strategy that ensures u in $\mathcal{F}(D, p)$. Every outcome of this strategy that get out of the D component, reach a state $(s', D') \in \mathsf{Succ}(D)$ where the weights are greater than u. These states cannot be losing since the weights on any dimension i would be smaller than $-W - 1$, which is smaller than u_i. This means that these states are winning, and to each state (s', D') with $D' \neq D$ that is reached by an outcome of $\sigma_\exists$ we can associate a strategy $\sigma_{\texttt{Eve}}^{s',D'}$ that is winning from (s', D'). We consider the strategy $\sigma'_\exists$ that plays according to $\sigma_\exists$ as long has we stay in the D component and according to $\sigma_{\texttt{Eve}}^{s',D'}$ once we reach another component, where (s', D') is the first state outside the D component that was reached.

The construction is such that the strategy $\sigma'_\exists$ ensures u in $\mathcal{D}(\mathcal{G})$: let ρ be one of its outcome, if ρ stays in the D component, it is also an outcome of $\sigma_\exists$ with the same payoff which is above u; otherwise ρ reaches a state (s', D') in $\mathsf{Succ}(D)$ and from this point $\texttt{Eve}$ follows a strategy that ensures threshold u.

E Appendix for Section 7

Theorem 13. *The robustness problem is* PSPACE*-complete.*

Proof. Note first that if the outcome is going to the state winning for $\texttt{Adam}$, it is possible for a A_i to change its strategy and go to $\bot$, thus improving its payoff. Therefore a $(n+1)$-resilient equilibrium is necessarily losing for $\texttt{Adam}$ and winning for all the others.

$\boxed{\text{Validity} \Rightarrow \text{equilibrium.}}$ Assume that ϕ is valid, we will show that there is a $(n+1)$-resilient equilibrium. We define a strategy of $\texttt{Eve}$ such that if v_h makes $\exists x_m.\, \forall x_{m+1} \cdots \exists x_n.\, C_1 \wedge \cdots \wedge C_k$ valid, then $\sigma_\exists(h) = X_m$ such that $v_{h \cdot X_m}$ makes $\forall x_{m+1} \cdots \exists x_n.\, C_1 \wedge \cdots \wedge C_k$ valid. As ϕ is valid, we know that for all outcomes h of $\sigma_\exists$ of the form $\texttt{Adam}_1 \cdot X_1 \cdots \texttt{Eve}_k \cdot X_k$, v_h makes $C_1 \wedge \cdots \wedge C_k$ valid. Then from X_m, $\texttt{Eve}$ can choose for each close a state Y that is different from all $X_1 \ldots X_m$. We also fix the strategy of all players A_i and B_i and $\texttt{Adam}$ to go to the state $\bot$. This defines a strategy profile that we will write σ_{Agt}.

Consider a strategy profile σ'_{Agt} where at most $(n+1)$ strategies are different from the ones in σ_{Agt}. Assume σ'_{Agt} reaches $\texttt{Eve}_m$. We know that in σ'_{Agt} at

least $n+1$ strategies are different from the ones σ_{Agt} and $\{A \in \mathsf{Agt} \mid \sigma'_A \neq \sigma_A\} = \{\mathtt{Adam}, X_1, \ldots, X_m\}$. Then, by the choice of the strategy for $\mathtt{Eve}$, the states that are seen in the following are controlled by players that are different from $X_1, \ldots, X_m$. Thus the run ends in $\bot$.

Equilibrium $\implies$ validity. Assume that $\sigma_\exists$ is part of a $(n+1)$-resilient equilibrium, we will show that ϕ is valid. Given a partial valuation $v_m \colon \{x_1, \ldots, x_m\} \mapsto \{\mathsf{true}, \mathsf{false}\}$, we define the function $f(v_m)$ such that:

$$f(v_m) \Leftrightarrow \sigma_\exists(\mathtt{Adam}_1 \cdot X_1 \cdots \mathtt{Adam}_m \cdot X_m) = B_{m+1}.$$

We will show that every valuation v, such that $v(x_{2k}) = f(v_{|2k-1})$, makes the formula $C_1 \wedge \cdots \wedge C_k$ valid, which shows that the formula ϕ is valid.

For all such valuations v, we can define strategies of $\mathtt{Adam}$ and players X_i such that $X_i = A_i$ if $v(x_i) = \mathsf{false}$ and $X_i = B_i$ otherwise, such that keeping all other strategies similar to σ_{Agt}, the state $\mathtt{Eve}_m$ is reached. Then, if we see a state belonging to one of the X_i, we can make the strategy go to the $\top$ state. Since the profile is $(n+1)$-resilient, this is impossible. Which shows that $\sigma_{\mathtt{Eve}}$ chooses for each clause a literal such that $v(\ell) = \mathsf{true}$. Therefore v makes the formula $C_1 \wedge \cdots \wedge C_k$ valid.