Nash Equilibria in Concurrent Priced Games

Miroslav Klimoš[1], Kim G. Larsen[2], Filip Štefaňák[1], Jeppe Thaarup[2]

[1] Masaryk University, Faculty of Informatics, Czech Republic
[2] Aalborg University, Department of Computer Science, Denmark

Abstract. Concurrent game structures model multi-player games played on finite graphs where the players simultaneously choose their moves and collectively determine the next state of the game. We extend this model with prices on transitions for each player. We study pure Nash equilibria in this framework where each player's payoff is the accumulated price of all transitions until reaching their goal state. We provide a construction of a Büchi automaton accepting all Nash equilibria outcomes and show how this construction can be used to solve a variety of related problems, such as finding pareto-optimal equilibria. Furthermore, we prove the problem of deciding the existence of equilibria to be NP-complete.

1 Introduction

Games played on graphs have enjoyed much attention from computer scientists in the past decades. Traditionally, they have been used to model scenarios where an actor tries to find a course of action against an unpredictable environment. Games have proven to be a helpful formalism with many applications. Bisimulation, accepting conditions of alternating automata, satisfiability of predicate logic can all be expressed as a two-player game with antagonistic objectives. Only one player can win in this case and the focus is usually limited to finding out which player has a winning strategy.

Non-zero-sum Games. In *non-zero-sum* games the players have independent objectives. Each player only cares about their own objective and does not care about objectives of others. Furthermore, it is natural to consider more than two players in this context. Such generalization of games allows for more realistic expression of real world problems and has been prominently used in economics, evolutionary biology or political science. In computer science, they have been used to model network routing problems [4]. The non-zero-sum games have been the focus of the *game theory* branch of mathematics for many years. However, game theorists do not usually study games and strategies as objects with internal structure.

The objectives of players in non-zero-sum games can be qualitative or quantitative. In the qualitative setting, each player can either win or lose, so the game has a set of winners. In the quantitative setting, the result of each player is a number – the cost – which they try to minimize (or maximize – in this case, the number is called payoff). In our case of graph games, the moves of the game are equipped with individual prices for each player.

Concurrent Games. Traditionally, the players take their decicions with full information and the game is turn-based. If the result depends on simultaneous and secret choices of multiple players, such as in the game rock-paper-scissors, we call the game *concurrent* [1]. Concurrent games are sufficient to describe all turn-based games, but they can model additional interesting problems, such as the famous prisoners' dilemma.

Nash Equilibria. Rational players adjust their play to the play of their opponents to improve their own benefit. If the game is repeated, the course of the game changes until they reach a situation where no player can further improve by unilaterally changing their play. Such state is called a *Nash equilibrium* [6] and it is the game theorists' tool of choice for the analysis of non-zero-sum games. Barring pacts between the players, the situation always stabilizes in a state that is a Nash equilibrium. A *pure* Nash equilibrium does not always exist and is not always optimal, but a game can also have multiple equilibria.

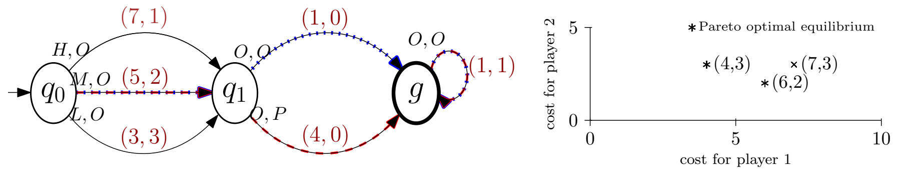


Fig. 1. A Priced Concurrent Games Structure and the Cost for its Equilibria

Our Contribution. In Section 2, we introduce Priced Concurrent Games Structures (PCGS), deterministic concurrent game graphs with non-negative integer prices on transitions for each player with individual reachability objectives. An example of a two-player PCGS can be seen in Fig. 1. From each state, a transition is determined by a choice of both players (pair of letters). Each transition is assigned a pair of numbers representing costs for the respective players. The goal state for both players is a bold circle.

A player provides a strategy that can consider the whole history of the game to choose a next move. The combination of strategies determines a run in the graph, which yields the cost for each player, defined as the accumulated price of all transitions until reaching their goal state.

The studied problem is to characterize all Nash equilibria of a given game. Variants of this problem include deciding existence of an equilibrium and limiting the search to equilibria with the costs of players constrained by bounds. In the chart in Fig. 1, we can see plotted costs of all Nash equilibria of the example.

In sections 3 and 4 we identify all Nash equilibrium strategy profiles by constructing a Büchi automaton accepting precisely the language of outcomes of all equilibrium strategy profiles satisfying a bound vector. Such characterization also allows for simple reduction of other similar problems, such as deciding an existence of any equilibrium by checking emptiness of a language of a Büchi automaton.

In Section 5, we characterize the complexity of the decision problem by proving that it is NP-complete in its several variants, except for the special case of turn-based games without bounds. We prove that an equilibrium always exists in turn-based games, which makes the general decision problem trivial.

Related Work Bouyer et. al. explore in [2] the existence of Nash equilibria in multiplayer concurrent games with reachability objectives. They include timed games, but they only consider qualitative reachability objectives. Brihaye et. al. study in [3] turn-based quantitative multiplayer games with reachability objectives. They prove existence of finite-memory Nash equilibria in such games. We confirm this in our framework as a corollary of our main result. However, it is necessary to point out that the formalisms are not completely equivalent. Most recently, Ummels et. al. study Nash equilibria in [7], using concurrent games but only with respect to limit-average objectives. The important distinction is that the initial part of the game is irrelevant to them. Thus, although related, the studied problems are quite different.

2 Preliminaries

We start with definitions of Concurrent Game Structures, computations, (full-memory) strategies and strategy profiles and outcomes of strategies. Then we introduce Priced Concurrent Game Structures by adding prices to transitions. Afterwards we formally define Nash equilibria on priced games. As the formalism of Büchi automata is well known, we only include the exact definition of Büchi automata and runs in the Appendix.

We use the symbol $\mathbb{N}_\infty = \mathbb{N} \cup \{\infty\}$ for the set of natural numbers with zero and infinity. The a-th projection of a vector X is denoted by X_a. We use the notation $X_{-a} = (X_1 \dots X_{a-1}, X_{a+1} \dots)$ for the vector X without its a-th element. We define a vector extension operator $\rhd_a$, which adds the first argument to the position a in the vector given as the second argument, i.e. $x_a \rhd_a (x_1 \dots x_{a-1}, x_{a+1} \dots x_n) = (x_1 \dots x_n)$ and $X_a \rhd_a X_{-a} = X$.

A word over alphabet Σ is a (finite or infinite) sequence of elements from Σ. Given a word w and $i \leq |w|$, $w[i]$ denotes the i-th element of w, w_i is a prefix of w of length i, and w^i is the i-th suffix s. t. $w = w_i.w^i$. An empty word is denoted by ϵ.

For the rest of the article we use standard comparison operators over natural numbers including zero and special symbols ∞ and $\perp$. For the sake of comparison, ∞ is the largest element and $\perp$ is the smallest element. Addition or subtraction involving ∞ results in ∞ (except when $\perp$ is involved). Addition or subtraction involving $\perp$ always results in $\perp$.

Definition 1 (Concurrent Game Structures). *A Concurrent Game Structure (CGS) is a tuple $(K, Q, q_0, \Phi, \phi, \mathbb{M}, \Delta, \delta)$ with the following components:*

- *A natural number $K \geq 1$ of* players*. We identify the players with numbers $1, \dots, K$ and we use notation $\Omega = \{1, \dots, K\}$ for the set of players.*
- *A finite set Q of* states*.*
- *An* initial state *$q_0 \in Q$.*
- *A finite set Φ of* propositions*.*
- *A* labeling function *$\phi : Q \to 2^\Phi$, such that for each state $q \in Q$, $\phi(q) \subseteq \Phi$ is a set of propositions true at q.*
- *A non-empty, finite set $\mathbb{M}$ of* moves*.*
- *A* move function *$\Delta : \Omega \times Q \to 2^\mathbb{M} \setminus \{\emptyset\}$, that defines a set of possible moves for each player and each state. For each state $q \in Q$, a move vector at q is a vector $J \in \mathbb{M}^k$ such that $J_a \in \Delta_a(q)$ for each player a. Given a state $q \in Q$ we write $\Delta(q) = \prod_{a \in \Omega} \Delta_a(q)$ for the set of all move vectors. We denote by $\Delta_{-b}(q) = \prod_{a \in \Omega, a \neq b} \Delta_a(q)$ the set of vectors of the possible moves of all players except b.*
- *A* transition function *δ, such that for each state $q \in Q$ and each move vector $J \in \Delta(q)$, it determines a state $\delta(q, J) \in Q$ that results from state q if every player $a \in \Omega$ chooses move J_a.*

Let us remark that commonly studied turn-based games are a special case of CGS, where in every state each player but one has exactly one possible move. We define the extended transition function $\hat{\delta}$ over finite words of move vectors inductively: $\hat{\delta}(q, \epsilon) = q$ and $\hat{\delta}(q, J.\Lambda) = \hat{\delta}(\delta(q, J), \Lambda)$, where $J \in \mathbb{M}^K$, $\Lambda \in (\mathbb{M}^K)^*$.

Definition 2 (Computation). *Let Λ be a (finite or infinite) word over alphabet $\mathbb{M}^K$ and q a state of a CGS. We say that Λ is a* computation *from q if for each position $i \in \mathbb{N}$, $\Lambda[i+1] \in \Delta(\hat{\delta}(q, \Lambda_i))$.*

Lemma 1. *Let Λ be a computation from state q. Then Λ^i is a computation from state $\hat{\delta}(q, \Lambda_i)$.*

Definition 3 (Strategy). *Given a CGS T, $q \in Q$ and $a \in \Omega$, a function $f : (\mathbb{M}^K)^* \to \mathbb{M}$ is a* strategy from state q for player a, *if $r = \hat{\delta}(q, \Lambda)$ implies $f_a(\Lambda) \in \Delta_a(r)$.*

Thus, a strategy of a player is a function that for a finite history determines their next move.

A vector $(f_1, \ldots, f_K)$ is a strategy profile from state q, *if for each $a \in \Omega$, f_a is a strategy from state q for player a.*

The operator $\rhd$ allows us to change the strategy of a player in a strategy profile: If F is a strategy profile from state q and f is a strategy from state q for player a, $f \rhd_a F_{-a}$ is a strategy profile, where all players except a use the strategies according to F and player a uses the strategy f. Similarly, if $j \in \Delta_a(q)$ is a move of player a in state q and $J \in \Delta(q)$ is a move vector, $j \rhd_a J_{-a}$ is a move vector with changed move for player a.

Definition 4 (Outcome). *An* outcome *is a function λ from strategy profiles to outcomes, s. t. whenever F is a strategy profile from a state q, $\lambda(F)$ is an infinite computation from q s. t. $\lambda(F)[i+1] = (F_1(\lambda(F)_i),\ \ldots, F_K(\lambda(F)_i))$.*

That is, $\lambda(F)$ is a computation where each step consists of individual moves of strategies from F based on current history.

Definition 5 (Priced Concurrent Game Structures). *A Priced Concurrent Game Structure (PCGS) is a tuple $(K, Q, q_0, \Phi, \phi, \mathbb{M}, \Delta, \delta, \gamma)$ with the following differences to CGS:*

- *The set of propositions Φ always includes $g_1, \ldots, g_K$, which are used to represent* goal states *for the respective players.*
- *The* price *function γ assigns to each state q and each move vector $J \in \Delta(q)$ a K-tuple of non-negative natural numbers which correspond to the price for each player. γ_i denotes the i-th projection of γ, i.e. the price for player i.*

Definition 6 (Cost). *Given a PCGS T, a* cost function *is a function $\Gamma : Q \times (\mathbb{M}^K)^* \to \mathbb{N}^K_\infty$ such that for each player $a \in \Omega$, state $q \in Q$ and computation Λ from q,*

$$\Gamma_a(q, \Lambda) = \sum_{i=1}^{k} \gamma_a(\hat{\delta}(q, \Lambda_{i-1}), \Lambda[i]),$$

where k is the smallest position such that $g_a \in \phi(\hat{\delta}(q, \Lambda_k))$. If no such k exists, $\Gamma_a(q, \Lambda) = \infty$.

We omit the state q since it is usually clear from the context of the computation Λ and write just $\Gamma(\Lambda)$.

Lemma 2. *Let Λ be a computation from q. If $g_a \in \phi(q)$, then $\Gamma_a(\Lambda) = 0$. Otherwise, $\Gamma_a(\Lambda) = \Gamma_a(\Lambda^1) + \gamma_a(q, \Lambda[1])$.*

Definition 7 (Nash Equilibrium). *Given a PCGS with initial state q_0, we say that strategy profile F from state q_0 is a* Nash equilibrium, *if for each player $a \in \Omega$ and for all strategies f_a of player a,*

$$\Gamma_a(\lambda(f_a \rhd_a F_{-a})) \geq \Gamma_a(\lambda(F)).$$

That is, no player can reduce their cost Γ_a by changing their strategy.

We say that a Nash equilibrium F satisfies bounds *$B \in \mathbb{N}^K_\infty$, if $B_a \geq \Gamma_a(\lambda(F))$.*

Example 1. Let us go back to the example in Fig. 1. Consider the following strategies f_H, f_M, f_L for player 1 from q_0 (all possible strategies) and f_1, f_2, f_3 for player 2 from q_0 (3 out of 8 possible strategies). For any history $\Lambda \in (\mathbb{M}^2)^*$:

$$f_H(\Lambda) = \begin{cases} H & \text{if } \Lambda = \epsilon \\ O & \text{otherwise} \end{cases} \qquad f_1(\Lambda) = \begin{cases} O & \text{always} \end{cases}$$

$$f_M(\Lambda) = \begin{cases} M & \text{if } \Lambda = \epsilon \\ O & \text{otherwise} \end{cases} \qquad f_2(\Lambda) = \begin{cases} P & \text{if } \Lambda = (L, O) \\ O & \text{otherwise} \end{cases}$$

$$f_L(\Lambda) = \begin{cases} L & \text{if } \Lambda = \epsilon \\ O & \text{otherwise} \end{cases} \qquad f_3(\Lambda) = \begin{cases} P & \text{if } \Lambda = (L, O) \text{ or } (M, O) \\ O & \text{otherwise} \end{cases}$$

The outcome for the strategy profile (f_M, f_2) is $\lambda(f_M, f_2) = (M, O)(O, O)^\omega$ (the blue dotted path). The cost of this outcome is $\Gamma(\lambda(f_M, f_2)) = (6, 2)$. The outcome for the strategy profile (f_L, f_2) is $\lambda(f_L, f_2) = (L, O)(O, P)(O, O)^\omega$ (the red dashed path). The cost of this outcome is $\Gamma(\lambda(f_L, f_2)) = (7, 3)$. Note that although the strategy for player 2 remains the same, their move is different in the second step.

Let us now examine some of the strategy profiles for equilibria. Profile (f_M, f_2) is a Nash equilibrium as no player can reduce their cost. Particularly, if player 1 uses f_L, he gets lower cost for the first step, but suffers even worse penalty in the second step. Other equilibria include (f_L, f_1), (f_M, f_1), and (f_L, f_3). Profiles (f_M, f_1) and (f_M, f_3) are not Nash equilibria. Player 1 may switch to f_L without any penalty. Profile (f_H, f_3) is not a Nash equilibrium. While switching to f_M does not do player 1 any good, switching to f_L yields an immediate benefit that is greater than the received penalty from player 2.

Problem 1 (Nash equilibria problem). Given a PCGS and bound vector $(b_1 \dots b_K) \in \mathbb{N}_\infty^K$, find all Nash equilibria F satisfying bounds $(b_1 \dots b_K)$.

Problem 2 (Decision variant of Nash equilibria problem). Given a PCGS and a bound vector $(b_1 \dots b_K) \in \mathbb{N}_\infty^K$, decide whether there is a Nash equilibrium F satisfying bounds $(b_1 \dots b_K)$. If all b_a are fixed to ∞, i.e. we decide whether there is some Nash equilibrium in general, we refer to this problem as the Decision variant of Nash equilibria problem without bounds.

We might want to consider only equilibria where each player reaches their goal, i. e. where each cost is finite. We would still want to be able to limit the individual costs.

Problem 3 (Decision variant of Nash equilibria with finite costs problem). Given a PCGS and a bound vector $(b_1 \dots b_K) \in \mathbb{N}_\infty^K$, decide whether there is a Nash equilibrium F satisfying bounds $(b_1 \dots b_K)$, in which each cost is finite.

3 Temptation and Punishment

When looking for Nash equilibria, we are looking neither for best strategies, nor for any competition. The players are not opponents, but collaborators. A Nash equilibrium is a stable mutual cooperation, where no player is tempted to defect. The cooperation need not be the most effective one. In games with multiple equilibria, we can often find multiple or even infinite number of stable cooperations that are strictly worse than other stable cooperations (such as the equilibrium (f_L, f_3) in the example from Fig. 1 with cost $(7, 3)$).

The *temptation* is the best possible cost a player can achieve by defecting a cooperation from a particular transition. Without temptation, all cooperations would be Nash equilibria because players would have no incentive to defect. That is also the approach we use for finding equilibria. We assume that players can agree on any outcome that is not jeopardized by a better temptation for one of the players.

Reactive games such as CGS allow strategies to detect a defection and adjust their behaviour. However, the players always provide their whole strategy in advance. Therefore the defecting adversary is able to adjust to the cooperating players' *punishment*. The punishment are the moves of the coalition that they had decided to use after they detect a betrayal. One of the properties of Nash equilibria strategies is that the players also advertise the punishment they will use. The defecting player can fully take advantage of that and choose a way of defection that guarantees the best outcome, given the future punishment.

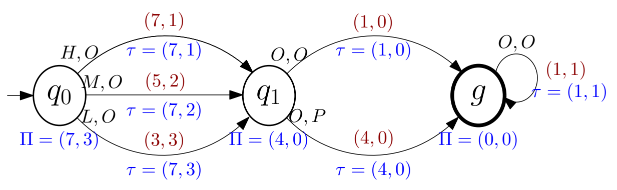
A strategy can never detect and punish a defection in advance. The reason for this is that a strategy is determined only by a history of moves. As long as the defecting strategy plays according to the deal, the other strategies must be using the same moves.

On the other hand, the punishing coalition's strategies do not care about their own cost anymore. After the alliance is broken by the traitor, all they are trying to do is make things miserable for the traitor. Such behaviour might seem counter-intuitive at first, but it makes sense when we reconsider the very purpose of such punishment. It exists to minimize temptation, and never has to occur when the players keep the deal which they have no incentive of breaking.

Let us introduce punishment values Π and temptation values τ. First of all, we state our requirements for these values and show an easy example. Then we provide an algorithm for computing punishment and temptation values based on relations between them and finally we formally prove that the computed values correspond to our requirements.

The *punishment value* $\Pi_a(q)$ for player a and state q is the cost of the worst outcome for player a which the coalition $\Omega \setminus \{a\}$ can enforce, starting in q. In other words, coalition $\Omega \setminus \{a\}$ has a strategy profile F_{-a} to guarantee that the cost for player a from q will be at least $\Pi_a(q)$.

Temptation value $\tau_a(q, J_{-a})$ for player a, state q and move J_{-a} of the coalition $\Omega \setminus \{a\}$ is the cost of the best outcome for player a starting from q, provided that $\Omega \setminus \{a\}$ use J_{-a} as their first move and they are commited to their strategy. In other words, if players $\Omega \setminus \{a\}$ use profile F_{-a} from q, starting with J_{-a}, player a has a strategy to guarantee that her cost from q will be at most $\tau_a(q, J_{-a})$. For the sake of simplicity, we use notation $\tau_a(q, J) = \tau_a(q, J_{-a})$.



	$\Pi(q_1) = (4, 0)$
$\Pi(q_0) = (7, 3)$	$\tau_1(q_1, O) = 1$
$\tau_1(q_0, O) = 7$	$\tau_1(q_1, P) = 4$
$\tau_2(q_0, H) = 1$	$\tau_2(q_1, O) = 0$
$\tau_2(q_0, M) = 2$	
$\tau_2(q_0, L) = 3$	$\Pi(g) = (0, 0)$
	$\tau_{1,2}(g, O) = 1$

Fig. 2. PCGS from Fig. 1 with punishment and temptation values

Our algorithm is based on the following relations between punishment and temptation values.

$$\begin{aligned} \tau_a(q, J_{-a}) &= \min_{j \in \Delta_a(q)} \gamma_a(q, j \rhd_a J_{-a}) + \Pi(\delta(q, j \rhd_a J_{-a})) \\ \text{if } g_a \notin \phi(q): \qquad \Pi_a(q) &= \max_{J \in \Delta_{-a}(q)} \tau_a(q, J) \\ \text{if } g_a \in \phi(q): \qquad \Pi_a(q) &= 0 \end{aligned}$$

It starts with an initial assignment of punishment values to ∞ for non-goal states and to 0 for goal states of a player. Then, in every iteration, it updates all the values according to these equations until reaching a fixed point. We can prove that no more than $|Q|$ iterations are needed and therefore the algorithm operates in polynomial time.

In the following lemmas – the proofs of which may be found in the Appendix – consider Π, τ to be values computed by the algorithm, i.e. values satisfying the above equations. Here we show that they accord with their meaning provided in the first paragraphs.

Lemma 3 (Punishment Lemma). *For each state q of a PCGS T, there is a strategy profile F_{-a} from q for coalition $\Omega \setminus \{a\}$, such that for each strategy f from q of player a, $\Gamma_a(\lambda(f \rhd_a F_{-a})) \geq \Pi_a(q)$.*

Lemma 4 (Temptation Lemma). *For each state q and move $J_{-a} \in \Delta_{-a}(q)$ and for each strategy profile F_{-a} from q for coalition $\Omega \setminus \{a\}$ starting with the move J_{-a}, there is a strategy f from state q of player a, such that $\Gamma_a(\lambda(f \rhd_a F_{-a})) \leq \tau_a(q, J_{-a})$*

4 Equilibrium Automaton

Theorem 1. *For a PCGS T, the set of all outcomes of Nash equilibria satisfying bounds $B \in \mathbb{N}_\infty^K$ is an $\omega-$regular language.*

We proceed with constructing a Büchi automaton accepting exactly the set of all equilibria outcomes. The idea of the construction is that we enhance states with local bounds for each player. Bounds $X = (x_1 \dots x_K)$ in state (q, X) mean that the cost of any infinite computation Λ from q represented by a run from (q, X) must satisfy X, i.e. $\Gamma_a(\Lambda) \leq x_a$. We are also allowed to say that we no longer care about the cost for player a from this state and let $x_a = \bot$.

Whenever there is a transition from q to r with cost C, there should be a transition from (q, X) to (r, Y'), where $Y' = (y'_1 \dots y'_K)$ and $y'_a = x_a - C_a$. However, whenever q is a goal state for player a, then instead the local bound y'_a for a is set to $\bot$, because the cost of this run for a has already been determined. This alone would allow us to observe the global bounds B.

On the other hand, we also have to account for the temptations of players to defect a potential equlibrium. Whenever we want to agree on a move J with temptation $\tau_a(q, J)$, then the cost of the rest of the outcome for player a must be lower than or equal to this temptation. Otherwise, player a would defect the cooperation in this transition. Therefore we also update local bounds in r to $y''_a = \tau_a(q, J) - C_a$.

We are interested in the lower of the two bounds y', y''. Thus finally,

$$y_a = \min(x_a, \tau_a(q, J)) - \gamma_a(q, J)$$

Note that such transition function guarantees that on any run of an equilibrium automaton, the local bound for any player a is nonincreasing.

Additionally, whenever y_a is lower than 0, we omit that transition. The transition function for the Equilibrium automaton will be deterministic, but not total.

A choice of accepting states reflects the local bounds. If local bound x_a is $\perp$, the cost is finite and respects the bounds. However, a computation Λ which never reaches a state with goal g_a has cost $\Gamma_a(\Lambda) = \infty$. Such computation can only be an equilibrium if local bounds for all states are ∞. Otherwise, there is a temptation for player a to defect. Therefore we allow ∞ as a local bound for accepting states.

Definition 8 (Equilibrium automaton $\mathcal{T}$). *For a PCGS T and $B \in \mathbb{N}^K_\infty$, an* Equilibrium automaton for T and bounds B *is a Büchi automaton* $(\Sigma, \mathcal{Q}, \delta', \{(q_0, B)\}, \mathcal{F})$ *with the following components:*

- Alphabet $\Sigma = \bigcup_{q \in Q} \Delta(q)$, *the set of all move vectors.*
- *The* state set $\mathcal{Q} = Q \times P_1 \times \ldots \times P_K$, *where P_a denotes the set $\{\perp, \infty\} \cup \{0, 1 \ldots p_a\}$. If $B_a \neq \infty$, $p_a = B_a$, otherwise p_a equals to the highest punishment value Π_a for player a lower than ∞.*
- *The* partial transition function $\delta' : \mathcal{Q} \times \Sigma \to \mathcal{Q}$, *defined as follows. For each state q and local bounds $X = (x_1 \ldots x_K)$, let $Y = (y_1 \ldots y_K)$, s. t.*

$$y_a = \begin{cases} \perp & \text{if } x_a = \perp \text{ or } g_a \in \phi(q), \\ \min(x_a, \tau_a(q, J)) - \gamma_a(q, J) & \text{otherwise.} \end{cases}$$

$$\text{Then } \delta'((q, X), J) = \begin{cases} (\delta(q, J), Y) & \text{if } y_a = \perp \text{ or } y_a \geq 0 \text{ for all } a, \\ \text{undefined} & \text{otherwise.} \end{cases}$$

- *The* initial state set $\{(q_0, B)\}$, *the initial state of T augmented with bounds B.*
- *The* accepting state set $\mathcal{F} = Q \times \{\perp, \infty\}^K$.

Lemma 5 (Correspondence). *Let T be a PCGS, L_T be the language of all outcomes $\lambda(F)$, such that F is a Nash equilibrium satisfying bounds $B \in \mathbb{N}^K_\infty$, and let $\mathcal{T}$ be an Equilibrium automaton for T and bounds B. Then $\mathcal{L}(\mathcal{T}) = L_T$.*

Proof. The $\subseteq$ direction is given by Lemma 7, the $\supseteq$ is given by Lemma 8.

Theorem 1 is a corollary of the previous Lemma.

Example 2. In Fig. 3 we provide a second example which is not turn-based and includes cycles. On the left side there is PCGS T with temptation and punishment values according to the previous section. The cost of each transition is $(1, 1)$, except for the transition $1, 1$ from q_k which is $(0, 1)$. On the right side, there is an Equilibrium automaton for T and bounds (∞, ∞).

Lemma 6. *Let $\mathcal{T}$ be an Equilibrium automaton for a PCGS T and bounds $B \in \mathbb{N}^K_\infty$.*

Then, if $\mathcal{T}$ has an accepting run ρ over word Λ, then Λ is a computation on T starting in q_0 and the state component of the extended state always corresponds to the state of the computation, e.g. for $\rho(i) = (q_i, x^i_1 \ldots x^i_K)$, $q_i = \hat{\delta}(q_0, \Lambda_i)$.

Furthermore, ρ satisfies local bounds in each state, e.g. for each player a, either $x^i_a = \perp$, or $\Gamma_a(\Lambda^i) \leq x^i_a$.

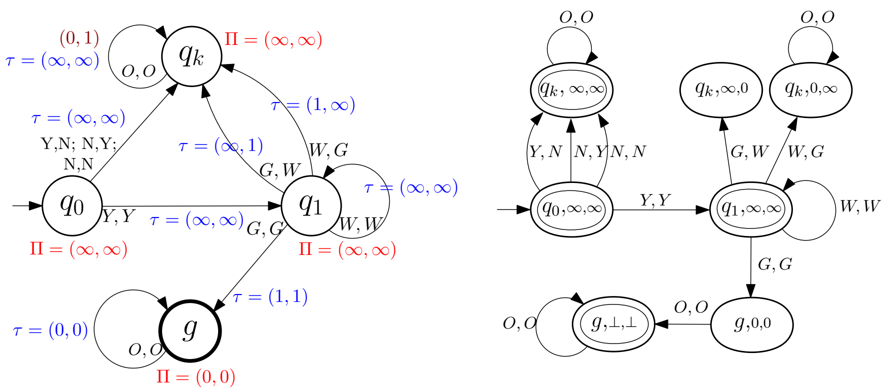


Fig. 3. Construction of an Equilibrium automaton for bounds (∞,∞)

Lemma 7. *Let $\mathcal{T}$ be an Equilibrium automaton for a PCGS T and bounds $B \in \mathbb{N}_{\infty}^{K}$.*

If $\mathcal{T}$ accepts Λ, there exists a strategy profile F from q_0, such that Λ is an outcome $\lambda(F)$ and F is a Nash equilibrium satisfying B.

Proof. Firstly, we construct strategy profile $F = (f_1 \dots f_K)$ s. t. $\Lambda = \lambda(F)$. These strategies follow Λ but as soon they detect a defection, they employ a punishing strategy according to the Punishment Lemma 3. Then we show that F is a Nash equilibrium by contradiction. Assuming that player a can reduce her cost by changing to f'_a, we find the last state q_{i-1} before the defection and refer to the Punishment Lemma for the next state to show that the new cost for a from q_{i-1} is at least x_a^{i-1}. However, since the original cost for a is at most x_a^{i-1} thanks to Lemma 6, we reach a contradiction with the improvement of the cost.

Finally F satisfies bounds B, as B are local bounds for state q_0 and Lemma 6 gives an upper bound for the cost of $\lambda(F)$. □

Lemma 8. *For each strategy profile F that is a Nash equilibrium on a PCGS T satisfying bounds B, their outcome is in the language of $\mathcal{T}$, the Equilibrium automaton for T and B. That is, $\lambda(F) \in \mathcal{L}(\mathcal{T})$.*

Proof. If Λ is not accepted by $\mathcal{T}$, we find the last index k s. t. condition $x_a^i = \perp$ or $\Gamma_a(\Lambda^i) \leq x_a^i$ is satisfied for each $i \leq k$. If such index does not exist because the run is infinite but the condition is always satisfied, the cost for some player is ∞ which implies that the condition is not satisfied in some state (a contradiction). If such index does not exist because the condition is never satisfied, the equilibrium does not meet the bounds.

Otherwise, we show that $\Lambda[k]$ is a move with a low temptation value for player a and according to the Temptation Lemma 4, we can find a strategy f defecting in this move, resulting in cost lower than the original. Thus, F is not Nash equilibrium. □

The Equilibrium automaton provides a straightforward solution to Problems 1 and 2. We can easily modify the automaton to solve Problem 3 by limiting the set of accepting states to those where all local bounds are $\perp$, as this corresponds to runs where each player reaches their goal. This also allow us to compute all Pareto optimal equilibria: the set of bounds $(b_1, \dots, b_k) \in \mathbb{N}^k$ satisfied by a

Nash equilibrium is clearly upwards closed. Having just presented the solution to Problem 3 we can apply the result of Valk and Jantzen [8] for computing the finite (due to Dickson's lemma) minimal such bounds:

Theorem 2. *The set of Pareto optimal bounds* $(b_1, \ldots, b_k) \in \mathbb{N}^k$ *satisfied by a Nash equilibrium can be computed.*

5 Complexity of the Decision Variant

Consider the decision variant of the Nash equilibria problem. With the construction of the Equilibrium automaton $\mathcal{T}$ for PCGS T and bounds B, the problem is reduced to deciding an existence of an accepting run in the Büchi automaton. Although the size of the automaton is possibly exponential, we present the following result:

Theorem 3. *Decision variant of the Nash equilibria problem is NP-complete.*

First we focus on showing that the problem is solvable in NP. The idea is that instead of constructing the Equilibrium automaton, we nondeterministically guess an accepting lasso in the automaton. We then verify the lasso in time linear to its length using the transition rules. The following two lemmas show that a lasso of polynomial length is sufficient for this.

Definition 9 (Relation $\succsim$). *Two states* $X = (q_X, x_1 \ldots x_K), Y = (q_Y, y_1 \ldots y_K)$ *are in relation* $X \succsim Y$ *iff* $q_X = q_Y$ *and for each player* a*, either* $x_a = y_a = \perp$*, or* $x_a \geq y_a > \perp$*.*

Lemma 9. *Let* X, Y *be two states of an equilibrium automaton* $\mathcal{T}$ *such that* $X \succsim Y$*. Then, for every computation* Λ *from* q*, whenever there is a run* ρ_Y *from state* Y *over* Λ*, then there also exists a run* ρ_X *from* X *over* Λ*. Furthermore,* $\rho_X(i) \succsim \rho_Y(i)$ *for all* i*.*

Proof. As all local bounds are lower in Y, whenever there is a transition from Y, the conditions of the transition function also hold in X. Furthermore, the end states are in $\succsim$. □

Lemma 10. *Given a PCGS with* K *players and* $|Q|$ *states, if there is an accepting run in* $\mathcal{T}$*, there is also an accepting run which is a lasso of length at most* $(K+2)|Q|$*.*

Proof. Let ρ be the shortest accepting lasso. We first show that the length of the cycle is at most $|Q|$. As the local bounds are nonincreasing, the values of the local bounds must be the same on all states on the cycle (either ∞ or $\perp$). Therefore they differ only in the base state. For cycle longer than $|Q|$ we find a repeating state and create a shorter accepting cycle, reaching a contradiction.

Now we show that the length of the nonrepeating path is at most $(K+1)|Q|$. If it is longer, some base state must repeat more than $K+1$ times. A local bound for player a can change to $\perp$ only once. Therefore between at least one pair of those repetitions, no local bound changes to $\perp$. The previous lemma gives us a run ρ' from the first of those states. Now take the first state X of the cycle on ρ. Since all local bounds are either $\perp$ or ∞, ρ' leads to this exact state and continues on the same accepting cycle. Joining ρ' and the path to X skips the steps between its one repetition, which contradicts that ρ is the shortest. □

Lemma 11. *Decision variant of the Nash equilibria problem is NP-hard.*

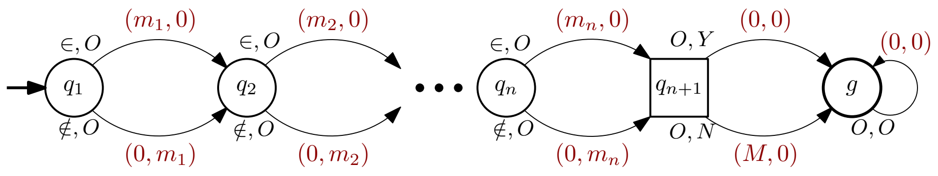


Fig. 4. Reduction from the subset sum problem

Proof. We show a reduction from the subset sum problem [5], i.e. for every input instance of the subset sum problem, we construct a game structure and bounds, such that there is a Nash equilibrium meeting these bounds iff the input instance of the subset sum problem has a solution.

Let $S = \{m_1, m_2, \ldots, m_n\}$ be a set of positive integers and m be the target sum. The input instance (S, m) has a solution iff there exists a set $S' \subseteq S$, such that sum of the numbers of S' is exactly m. Let $M = \sum_{s \in S} s$, the sum of all numbers. We construct a two-player turn-based game G according to the Fig. 4. The initial state is q_1, g is a goal state for both players and numbers above the transitions represent the prices. In circle (resp. square) states, player 1 (resp. player 2) chooses the next move. We set the bounds $(b_1, b_2) = (m, M - m)$.

For the first direction, suppose there is a solution to the subset sum problem S'. Now consider the following strategies for the players.

Player 1 In $q_i (1 \leq i \leq n)$, choose $\in$ if $m_i \in S'$, otherwise choose $\notin$.
Player 2 In q_{n+1}, choose Y if the accumulated costs so far are $(m, M - m)$, otherwise choose N.

Outcome of these strategies has costs $(m, M - m) = (b_1, b_2)$. If player 1 changes his strategy such that the costs are different in q_{n+1}, his cost increases by $M > m$ in the last transition and thus she can not improve her cost. Player 2 can not influence her cost at all. The strategies are Nash equilibrium.

Now for the second direction, suppose there exists a Nash equilibrium meeting the bounds (b_1, b_2). As the sum of the costs for both players is at least $m_1 + \ldots + m_n = M = m + (M - m) = b_1 + b_2$, the cost is exactly (b_1, b_2). We consider the outcome and construct the set S' as the set $\{m_i \mid \text{player 1 chooses the} \in \text{in } q_i\}$. Since the cost for player 1 is m, the sum of S' must be exactly m and it is a solution to the subset sum problem. □

The Theorem 3 is a corollary of the previous lemma and Lemma 10. We now show that the problem of deciding the existence of an equilibrium point is NP-hard even if we have no bounds on the possible equilibria.

Theorem 4. *Decision variant of the Nash equilibria problem without bounds is NP-complete.*

Proof. Given a two-player PCGS T and bounds (b_1, b_2) we construct a two-player PCGS T' according to the Fig. 5. The new initial state is q'_0 and the state g is a goal state of both players. We need to prove that there is an equilibrium in T' iff there is an equilibrium in T satisfying the bounds (b_1, b_2).

None of the added edges can be part of an equilibrium outcome as can be seen in the Table of choices in Fig. 5. The horizontal arrows indicate improvement for player 1, the vertical for player 2. Every equilibrium in T satisfying bounds (b_1, b_2) is preserved in T', but the equilibria not satisfying the bounds are suppressed, because both players could change their first move and get a better cost. □

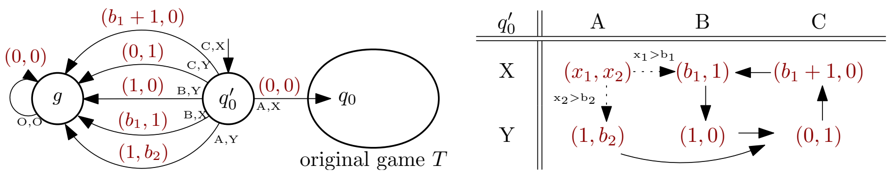


Fig. 5. Reduction from the problem with bounds (b_1, b_2) to the problem without bounds

Lemma 11 shows NP-hardness even for turn-based games. However, [3] shows that for a special kind of games which roughly correspond to turn-based games where cost for each transition and player is 1, a Nash equilibrium always exists. Without elaborating on this further, we can confirm their result for any priced turn-based game with strategies that use history. Hence, the decision problem for this case is trivial and the answer is always positive.

6 Conclusion

We introduced the Nash Equilibrium problem with bounds for priced concurrent games structures and provided a construction of a Büchi automaton accepting the set of all equilibria outcomes, characterizing the class of all Nash equilibria outcomes as an ω-regular set. The Equilibrium automaton can also be used to solve a variety of similar problems involving Nash equilibria, such as deciding existence of equilibria satisfying properties expressed in Linear temporal logic.

Furthermore, we characterized the complexity of the decision variant of the problem as NP-complete. The problem remains NP-complete even if we consider either turn-based PCGS, or we omit bounds. If we do both, the problem becomes trivial as the equilibrium always exists.

References

1. Alur, R., Henzinger, T.A., Kupferman, O.: Alternating-time temporal logic. J. ACM 49, 672–713 (September 2002), `http://doi.acm.org/10.1145/585265.585270`
2. Bouyer, P., Brenguier, R., Markey, N.: Nash equilibria for reachability objectives in multi-player timed games. In: Gastin, P., Laroussinie, F. (eds.) CONCUR 2010 - Concurrency Theory, Lecture Notes in Computer Science, vol. 6269, pp. 192–206. Springer Berlin / Heidelberg (2010), `http://dx.doi.org/10.1007/978-3-642-15375-4_14`, 10.1007/978-3-642-15375-4_14
3. Brihaye, T., Bruyère, V., De Pril, J.: Equilibria in quantitative reachability games. In: Ablayev, F., Mayr, E. (eds.) Computer Science – Theory and Applications, Lecture Notes in Computer Science, vol. 6072, pp. 72–83. Springer Berlin / Heidelberg (2010), `http://dx.doi.org/10.1007/978-3-642-13182-0_7`, 10.1007/978-3-642-13182-0_7
4. Felegyhazi, M., Hubaux, J., Buttyan, L.: Nash equilibria of packet forwarding strategies in wireless ad hoc networks. IEEE Transactions on Mobile Computing pp. 463–476 (2006)
5. Gormen, T., Leiserson, C., Rivest, R., Stein, C.: Introduction to algorithms. MIT Press and McGraw-Hill Book Company 7, 1162–1171 (1976)
6. Nash, J.: Equilibrium points in n-person games. Proceedings of the National Academy of Sciences of the United States of America 36(1), 48–49 (1950)
7. Ummels, M., Wojtczak, D.: The complexity of nash equilibria in limit-average games. In: Katoen, J.P., König, B. (eds.) CONCUR. Lecture Notes in Computer Science, vol. 6901, pp. 482–496. Springer (2011)
8. Valk, R., Jantzen, M.: The residue of vector sets with applications to decidability problems in petri nets. In: Application and Theory of Petri Nets. pp. 234–258 (1984)

7 Appendix

7.1 Büchi automaton

Definition 10 (Büchi automaton). *A Büchi automaton $\mathcal{A}$ is a five tuple $(\Sigma, Q, \delta, Q^0, F)$ s. t.*

- Σ *is the finite* alphabet*.*
- Q *is the finite set of* states*.*
- $\delta \subseteq Q \times \Sigma \times Q$ *is the* transition relation*.*
- $Q^0 \subseteq Q$ *is the set of* initial states*.*
- $F \subseteq Q$ *is the set of* final states*.*

Let Λ be a (finite or infinite) word from $\Sigma^* \cup \Sigma^\omega$. A *run* over Λ from q is a mapping $\rho : \{0, 1 \dots |\Lambda|\} \to Q$ such that:

- The first state is q, that is, $\rho(0) = q$.
- Moving from the ith state $\rho(i)$ to the $i+1$st state $\rho(i+1)$ upon reading the ith input letter $\Lambda[i]$ is consistent with the transition relation. That is, for $0 \leq i < |\Lambda|$, $(\rho(i), \Lambda[i], \rho(i+1)) \in \delta$.

We say that ρ is a run of $\mathcal{A}$ over Λ if ρ is a run over Λ from q and q is an initial state. That is, $\rho(0) \in Q^0$.

Let $\inf(\rho)$ be the set of states that appear infinitely often in the run ρ. A run ρ of $\mathcal{A}$ over an infinite word Λ is *accepting* iff some accepting state appears in ρ infinitely often. That is, $\inf(\rho) \cap F \neq \emptyset$.

An automaton $\mathcal{A}$ *accepts* a word Λ iff there exists an accepting run of $\mathcal{A}$ over Λ.

The *language* of $\mathcal{A}$, $\mathcal{L}(\mathcal{A}) \subseteq \Sigma^\omega$ consists of all words accepted by $\mathcal{A}$.

7.2 Algorithm computing punishment and temptation values

Algorithm 1: Computation of punishment and temptation values

```
Input: PCGS $(K, Q, q_0, \Phi, \phi, \Delta, \delta, \gamma)$
Output: Punishment values $\Pi$ and temptation values $\tau$
begin
    foreach $q \in Q$ and $a \in \Omega$ do
        if $g_a \in \phi(q)$ then $\Pi_a(q) := 0$ else $\Pi_a(q) := \infty$;
    end
    repeat $|Q|$ times  or  until no values change
        foreach $q \in Q$ and $a \in \Omega$ do
            foreach $J_{-a} \in \Delta_{-a}(q)$ do
                $\tau_a(q, J_{-a}) := \min_{j \in \Delta_a(q)} \gamma_a(q, j \rhd_a J_{-a}) + \Pi_a(\delta(q, j \rhd_a J_{-a}))$;
            end
            if $g_a \notin \phi(q)$ then $\Pi_a(q) := \max_{J \in \Delta_{-a}(q)} \tau_a(q, J)$;
        end
    end
end
return $\Pi, \tau$
```

Lemma 12. *During the execution of Algorithm 1, there is no update in* $(|Q|+1)$*-th iteration.*

Proof. We show that in $|Q|$-th iteration, there is no update of Π and, therefore, the changes can propagate to τ in the last run. The idea of the proof is the following. The update of Π in some state can induce another updates in connected states. However, the update path, i.e. the sequence of induced updates, cannot go through the same state twice. The complete proof can be found in Appendix. $\square$

7.3 Proof of Lemma 12

Proof. We will prove that in $|Q|$-th iteration, there is no update of Π and, therefore, the changes can propagate to τ in the next, last run. We say that update of $\Pi(q)$ is induced by update of $\Pi(q')$, if $q' = \delta(q, J)$ and $\Pi(q) = \gamma(q, J) + \Pi(q')$ after the update. Let us start with some observations:

1. Both Π and τ values decrease in time.
2. If we consider all the punishment values to be initially ∞ and setting them to 0 in goal states as the first update, we can say that every next update is induced by some previous update of punishment value in a succesor. If there are more possiblities, we do not care which one we take.
3. If an update of $\Pi(q)$ is induced by a change of $\Pi(q')$, then $\Pi(q) \geq \Pi(q')$.

To get a contradiction, suppose there is a change of punishment value in state q_1 in $|Q|$th iteration. Then, with the second observation, we can reconstruct the update path $(q_1, v_1), (q_2, v_2), \ldots, (q_m, 0)$, such that the update in q_i to value v_i was induced by the update in q_{i+1} to value v_{i+1} and q_m is a goal state. The length of this path is at least $|Q| + 1$, because we start with $(q_m, 0)$ and every iteration lengthened the path by at least one. Hence at least one state repeats, let say $q_k = q_l, k < l$. From the third observation we know that $v_i \geq v_{i+1}$, so $v_k \geq v_l$. The first observation says that the values decrease in time, so $v_k \leq v_l$. The only possible case $v_k = v_l$ means that there was no update and that is a contradiction. $\square$

7.4 Proof of Punishment Lemma 3

Proof. Let strategies in F_{-a} choose in each state the move selected by the maximum function when updating the punishment value in the state. To get a contradiction, suppose that there is a strategy f for player a, such that $\Gamma_a(\Lambda) < \Pi_a(q)$, where $\Lambda = \lambda(f \rhd_a F_{-a})$ is the corresponding outcome from the state q. Denote by $q_i = \hat{\delta}(q, \Lambda_i)$ the i-th state visited in computation Λ.

Consider the longest finite prefix Λ_k of Λ, s. t. $\Gamma_a(\Lambda^l) < \Pi_a(q_l)$ for all $l \leq k$. Note that there is always such a state, because the computation goes through a goal state, where this condition does not hold. Let $J = \Lambda[k]$ be the vector of moves of the players from q_k according to the strategy profile $f \rhd_a F_{-a}$. Since J_{-a} is the move vector selected by the maximum function, it follows from the algorithm that

$$\Pi_a(q_k) = \min_{j \in \Delta_a(q_k)} \gamma_a(q_k, j \rhd_a J_{-a}) + \Pi_a(\delta(q_k, j \rhd_a J_{-a})) \leq \gamma_a(q_k, J) + \Pi_a(q_{k+1})$$

From this inequality we get

$$\Gamma_a(\Lambda^{k+1}) = \Gamma_a(\Lambda^k) - \gamma_a(q_k, J) < \Pi_a(q_k) - \gamma_a(q_k, J) \leq \Pi_a(q_{k+1})$$

and that is a contradiction with the selection of Λ_k. $\square$

7.5 Proof of Temptation Lemma 4

Proof. The proof is similar to the proof of Lemma 3. To get a contradiction, suppose that there is a strategy profile F_{-a}, such that for any strategy f for a, $\Gamma_a(\Lambda) > \tau_a(q, J_{-a})$, where $\Lambda = \lambda(f \rhd_a F_{-a})$.

Let construct the strategy f as follows. Denote by $J^k_{-a} = F_{-a}(\Lambda_k)$ the move of $\Omega \setminus \{a\}$ in k-th step. Let the move j^k_a of player a in k-th step is the one selected by the minimum function when updating the temptation value $\tau_a(q_k, J^k_{-a})$.

Consider the longest prefix Λ_k of Λ, such that $\Gamma_a(\Lambda^l) > \tau_a(q_l, J^l_{-a})$, for all $l \leq k$. Since j_a is the move selected by the minimum function and Π is the maximum of temptation values,

$$\tau_a(q_k, J^k_{-a}) = \gamma_a(q_k, j^k_a \rhd_a J^k_{-a}) + \Pi_a(q_{k+1}) \geq \gamma_a(q_k, j^k_a \rhd_a J^k_{-a}) + \tau_a(q_{k+1}, J^{k+1}_{-a})$$

From this inequality we get

$$\Gamma_a(\Lambda^{k+1}) = \Gamma_a(\Lambda^k) - \gamma_a(q_k, j^k_a \rhd_a J^k_{-a})$$

$$\Gamma_a(\Lambda^k) - \gamma_a(q_k, j^k_a \rhd_a J^k_{-a}) > \tau_a(q_k, J^k_{-a}) - \gamma_a(q_k, j^k_a \rhd_a J^k_{-a}) \geq \tau_a(q_{k+1}, J^{k+1}_{-a})$$

and that is a contradiction with the selection of Λ_k.

□

7.6 Proof of Lemma 6

Proof. We show the first statement by induction over i.

For base case, $i = 0$, $\Lambda_i = \epsilon$. From the definition, $q_0 = \hat{\delta}(q_0, \epsilon)$. Now since $(\rho(0), \Lambda[1], \rho(1)) \in \delta'$, also $q_1 = \delta(q_0, \Lambda[1])$ and therefore $\Lambda[1] \in \Delta(q_0) = \Delta(\hat{\delta}(q_0, \epsilon))$.

Now let assume that the statement holds for i. We show that it holds for $i + 1$. Since $(\rho(i), \Lambda[i + 1], \rho(i + 1)) \in \delta'$, also $q_{i+1} = \delta(q_i, \Lambda[i + 1])$. Therefore $\Lambda[i + 1] \in \Delta(q_i) = \Delta(\hat{\delta}(q_0, \Lambda_i))$. Also, from the definition of $\hat{\delta}$, $\hat{\delta}(q_0, \Lambda_{i+1}) = \delta(\hat{\delta}(q_0, \Lambda_i), \Lambda[i + 1]) = \delta(q_i, \Lambda[i + 1]) = q_{i+1}$.

The induction shows that the statement holds for all $i \geq 0$. Now we prove the second statement.

Since ρ is accepting, an accepting state must repeat infinitely often. Let that state be $(r, y_1, \ldots, y_K)$. We show that the consistency holds for any player a. Local bound y_a must either be ∞ or $\bot$.

If y_a is ∞, then x^i_a is ∞ for all i, because local bounds are nonincreasing. Indeed, assume that for some j, $x^j_a < \infty$. Then for each $i \geq j$, $x^i_a \leq x^j_a < \infty$. However, then for no $i \geq j$, $\rho(i) = (r, y_1, \ldots, y_K)$, which is a contradiction.

If y_a is $\bot$, let j be the the the first index such that $x^{j+1}_a = \bot$. The rules of transition relation δ' clearly show that for each $i \geq j + 1$, $x^{j+1}_a = \bot$.

Additionally, we also can make sure that j is the smallest index such that $g_a \in \phi(q_j)$. If there was some smaller index $i < j$, then $x^{i+1}_a = \bot$ according to δ', which contradicts that j is the smallest index s. t. $x^{j+1}_a = \bot$.

We show by backwards induction that for each $i \leq j$, $\Gamma_a(\Lambda^i) \leq x^i_a$.

For the base case, $i = j$, $\Gamma_a(\Lambda^j) = 0$ since $g_a \in \phi(q_j)$. As x^j_a can not be $\bot$, $0 \leq x^j_a$.

Let us assume that the inequality holds for $i+1$, $0 \le i < j$. We show that it also holds for i. According to the transitions rules, $x_a^{i+1} = \min(x_a^i, \tau_a(q_i, \Lambda[i+1])) - \gamma_a(q_i, \Lambda[i+1])$. Therefore from the hypothesis:

$$\begin{aligned} \Gamma_a(\Lambda^{i+1}) &\le \min(x_a^i, \tau_1(q_i, \Lambda[i+1])) - \gamma_a(q_i, \Lambda[i+1]) \\ \Gamma_a(\Lambda^i) &\le \min(x_a^i, \tau_a(q_i, \Lambda[i+1])), \quad \text{since } g_a \notin \phi(q_i) \\ \Gamma_a(\Lambda^i) &\le x_a^i \end{aligned}$$

□

7.7 Proof of Lemma 7

Proof. First, we construct strategy profile $F = (f_1 \dots f_K)$ from q_0 with outcome Λ. The strategies are constructed in the following way. As long as the history matches Λ, both strategies are moving according to Λ:

$$f_a(\Lambda_i) = j_a^{i+1}, \text{where } \Lambda[i+1] = (j_1^{i+1} \dots j_K^{i+1})$$

By reversal of this rule for strategies all f_a, we get:

$$\Lambda[i+1] = (f_1(\Lambda_i) \dots f_K(\Lambda_i))$$

Therefore $\Lambda = \lambda(F)$.

Whenever Λ' is not a prefix of Λ, it means that a player betrayed the coalition in the past and the remaining players will employ their punishment strategy. Let l be the smallest index s. t. $\Lambda'[l] \ne \Lambda[l]$. Then let p_a be a strategy for a from $\hat{\delta}(q_0, \Lambda'_l)$ according to the Punisment Lemma 3. We set f_a to act exactly like p_a from this state:

$$f_a(\Lambda') = p_a(\Lambda'^l)$$

We now show that F is a Nash equilibrium by contradiction. Let us assume the contrary, that there exists a strategy f'_a from q_0 for player a which achieves lower cost. Let $\Lambda' = \lambda(f'_a \rhd_a F_{-a})$. Then:

$$\Gamma_a(\Lambda') < \Gamma_a(\Lambda)$$

Let $r_i = \hat{\delta}(q_0, \Lambda_i)$ for any i. Since the cost of the outcomes is different, no state $r_i, i < l$ that occurs before the distinguishing move $\Lambda'[l]$ is a goal state for player a.

According to the punishment lemma, the cost of outcome Λ'^l for player a from r_l is greater than or equal than its punishment value $\Pi_a(r_i)$:

$$\begin{aligned} \Gamma_a(\Lambda'^l) &\ge \Pi_a(r_l) \\ \Gamma_a(\Lambda') &\ge \Pi_a(r_l) + \sum_{i=1}^{l} \gamma_a(r_{i-1}, \Lambda'[i]) \\ \Gamma_a(\Lambda') &\ge \Pi_a(r_l) + \gamma_a(r_{l-1}, \Lambda'[l]) + \sum_{i=1}^{l-1} \gamma_1(r_{i-1}, \Lambda'[i]) \end{aligned}$$

Since temptation value τ_a is calculated as a minimum:

$$\Pi_a(r_l) + \gamma_a(r_{l-1}, \Lambda'[l]) \geq \tau_a(r_{l-1}, \Lambda'[l])$$
$$\Gamma_a(\Lambda') \geq \tau_a(r_{l-1}, \Lambda'[l]) + \sum_{i=1}^{l-1} \gamma_a(r_{i-1}, \Lambda'[i]).$$

Now, let us go back to the original computation Λ accepted by $\mathcal{T}$. There is run ρ over Λ. Let $\rho(i) = (q_i, x_1^i \dots x_K^i)$. As the prefixes Λ_{l-1} and Λ'_{l-1} are identical, $q_i = r_i = \hat{\delta}(q_0, \Lambda_i)$ for all $i \leq l-1$.

Since q_{l-1} is not a goal state for player 1, we can extract from the transition rules of $\mathcal{T}$ the following:

$$\min(x_1^{l-1}, \tau_a(q_{l-1}, \Lambda[l])) - \gamma_a(q_{l-1}, \Lambda[l]) = x_a^l$$
$$\tau_a(q_{l-1}, \Lambda[l]) - \gamma_a(q_{l-1}, \Lambda[l]) \geq x_1^l$$

Note that since $g_a \notin q_i$ for all $i < l$, we know that $x_a^l \neq \bot$. From Lemma 6, we get that $x_a^l \geq \Gamma_a(\Lambda^l)$.

$$\tau_a(q_{l-1}, \Lambda[l]) - \gamma_a(q_{l-1}, \Lambda[l]) \geq \Gamma_a(\Lambda^l)$$
$$\tau_a(q_{l-1}, \Lambda[l]) \geq \Gamma_a(\Lambda^{l-1})$$

Now as $\Lambda_{l-1} = \Lambda'_{l-1}$, then also $f_b(\Lambda_{l-1}) = f_b(\Lambda'_{l-1})$ for all players b. Therefore if $\Lambda[l] = J^l$, then $\Lambda'[l] = j_a'^l \rhd_a J_{-a}^l$ and:

$$\tau_a(q_{l-1}, j_a'^l \rhd_a J_{-a}^l) = \tau_a(q_{l-1}, J^l)$$
$$\tau_a(r_{l-1}, \Lambda'[l]) = \tau_a(q_{l-1}, \Lambda[l])$$
$$\tau_a(r_{l-1}, \Lambda'[l]) \geq \Gamma_a(\Lambda^{l-1})$$
$$\tau_a(r_{l-1}, \Lambda'[l]) + \sum_{i=1}^{l-1} \gamma_a(r_{i-1}, \Lambda[i]) \geq \Gamma_a(\Lambda^{l-1}) + \sum_{i=1}^{l-1} \gamma_a(r_{i-1}, \Lambda[i])$$
$$\tau_a(r_{l-1}, \Lambda'[l]) + \sum_{i=1}^{l-1} \gamma_a(r_{i-1}, \Lambda'[i]) \geq \Gamma_a(\Lambda)$$
$$\Gamma_a(\Lambda') \geq \Gamma_a(\Lambda),$$

which is a contradiction. □

7.8 Proof of Lemma 8

Proof. We will prove this lemma by contradiction. Let F be a Nash equilibrium on T satisfying bounds B, but $\Lambda = \lambda(F) \notin \mathcal{L}(\mathcal{T})$.

Since the transition function is deterministic, there can be at most one run of $\mathcal{T}$ over Λ. We take the longest possible prefix Λ_j s. t. we can find run ρ over Λ_j. Let $\rho(i) = (q_i, x_1^i \dots x_K^i)$. Then we find the last index k where all local bounds are met, that is the largest k s. t. for each player a and $j \leq k$, $x_a^j \geq \Gamma_a(\Lambda^j) \vee x_a^j = \bot$.

If such index does not exist because the local bounds are never met, then for $j = 0$, $\mathit{\Gamma}_a(\mathit{\Lambda}) > x^0_a = B_a \neq \bot$. Therefore equilibrium F does not meet bounds B.

If such index does not exist because the run is infinite and the bounds are always met, the run must be nonaccepting. Hence, since the bounds are nonincreasing, a bound for some player must be always greater than $\bot$, but also lower than ∞ after some point. However, that means that the run does not pass any goal state for that player and the cost for her is infinite. Thus we can see that there is a state where the bounds are not met, which is a contradiction.

Finally, for the case when such k does exist, we prove that F is not a Nash equilibrium and $k + 1$ is the index of a move in which a player would defect.

Case $k = i$**:** Since i is the length of the induced path, $\delta'((q_i, x^i_1 \dots x^i_K), \mathit{\Lambda}[k + 1])$ is undefined. Therefore there is player a, s. t. $x^k_a \neq \bot$, $g_a \neq \phi(q_k)$, and $0 > y^k_a$.

$$\begin{aligned} 0 &> \min(x^k_a, \tau_a(q_k, \mathit{\Lambda}[k + 1])) - \gamma_a(q_k, \mathit{\Lambda}[k + 1]) \\ \mathit{\Gamma}_a(\mathit{\Lambda}^k) = \gamma_a(q_k, \mathit{\Lambda}[k + 1]) &> \min(x^k_a, \tau_a(q_k, \mathit{\Lambda}[k + 1])) \end{aligned}$$

Case $k < i$**:** There must be a player a for which $\mathit{\Gamma}_a(\mathit{\Lambda}^{k+1}) > x^{k+1}_a \wedge x^{k+1}_a \neq \bot$. Otherwise, k would not be the largest index. That gives us $g_a \notin \phi(q^k)$ and $x^k_a \neq \bot$.

$$\begin{aligned} \mathit{\Gamma}_a(\mathit{\Lambda}^{k+1}) &> x^{k+1}_a \\ x^{k+1}_a &= \min(x^k_a, \tau_a(q_k, \mathit{\Lambda}[k + 1])) - \gamma_a(q_k, \mathit{\Lambda}[k + 1]) \\ \gamma_a(q_k, \mathit{\Lambda}[k + 1]) + \mathit{\Gamma}_a(\mathit{\Lambda}^{k+1}) &> \min(x^k_a, \tau_a(q_k, \mathit{\Lambda}[k + 1])) \\ \mathit{\Gamma}_a(\mathit{\Lambda}^k) &> \min(x^k_a, \tau_a(q_k, \mathit{\Lambda}[k + 1])) \end{aligned}$$

Finally, since $\mathit{\Gamma}_a(\mathit{\Lambda}^k) \leq x^k_a$, then for both cases:

$$\mathit{\Gamma}_a(\mathit{\Lambda}^k) > \tau_a(q_k, \mathit{\Lambda}[k + 1])$$

Now we let t_a be a strategy for player a from q_k for move $\mathit{\Lambda}[k + 1]$ according to the Temptation Lemma. We construct a strategy f'_a:

$$\begin{aligned} f'_a(\mathit{\Lambda}_k.\mathit{\Lambda}') &= t_a(\mathit{\Lambda}') \\ f'_a(\mathit{\Lambda}') &= f_a(\mathit{\Lambda}') \text{ otherwise} \end{aligned}$$

Now $\lambda(f'_a \rhd_a F_{-a})$ is $\mathit{\Lambda}_k.\mathit{\Lambda}'$, because a follows $\mathit{\Lambda}$ for k steps. According to the Temptation Lemma, $\tau_a(q_k, \mathit{\Lambda}[k + 1]) \geq \mathit{\Gamma}_a(\mathit{\Lambda}')$, and $\mathit{\Gamma}_a(\mathit{\Lambda}^k) > \mathit{\Gamma}_a(\mathit{\Lambda}')$ Now since for all $j \leq k$, $g_a \neq \phi(q_j)$,

$$\begin{aligned} \mathit{\Gamma}_a(\mathit{\Lambda}) &> \mathit{\Gamma}_a(\mathit{\Lambda}_k.\mathit{\Lambda}') \\ \mathit{\Gamma}_a(\lambda(F)) &> \mathit{\Gamma}_a(\lambda(f'_a \rhd_a F_{-a})) \end{aligned}$$

Therefore for this last case, F is not a Nash equilibrium. We have exhausted all cases and reached a contradiction. □

7.9 Proof of Theorem 4

Proof. Given a two-player PCGS T and bounds (b_1, b_2) we construct a two-player PCGS T' according to the Figure 5. The new initial state is q'_0 and the state g is a goal state of both players. We need to prove that there is an equilibrium in T' iff there is an equilibrium in T satisfying the bounds (b_1, b_2).

Suppose there is a Nash equilibrium strategy profile F in T satisfying the bounds b_1, b_2. Then there is a strategy profile F' in T', in which both players start with the move 1 and then continue according to F. Formally, $F'(\epsilon) = 1, F'(\Lambda) = F(\Lambda^1)$ if $\Lambda[1] = (1, 1)$, $F'(\Lambda) = 1$ otherwise. Changing the first move is not profitable for either player: If player 1 change his move to 2 or 3, his cost will be at least $b_1 \geq \Gamma_1(\lambda(F'))$. If player 2 change his move to 2, his cost will be $b_2 \geq \Gamma_2(\lambda(F'))$. Hence, there is a Nash equilibrium in T'.

On the other hand, suppose there is a Nash equilibrium strategy profile F' in T'. If it is not the case, that both players are starting with move 1, at least one player can improve his cost according to the table in Figure 5. Hence, $\lambda(F')$ starts with $(1, 1)$. Denote by $c_i = \Gamma_i(\lambda(F'))$ the cost for player i when using F'. If $c_1 > b_1$ (resp. $c_2 > b_c$), the player 1(resp. 2) have better to change his first move to 2 to improve his cost to b_1(resp. b_2) and it would not be an equilibrium. Hence, F' satisfies the bouds (b_1, b_2) and there is a Nash equilibrium strategy profile F: $F(\Lambda) = F'((1, 1).\Lambda)$ in PCGS T, also satisfying the bounds.