Synthesizing Optimally Resilient Controllers*

Daniel Neider[1], Alexander Weinert[2] and Martin Zimmermann[3]

[1] Max Planck Institute for Software Systems, 67663 Kaiserslautern, Germany
neider@mpi-sws.org
[2] German Aerospace Center (DLR), Intelligent and Distributed Systems, Linder Höhe, 51147 Köln, Germany
alexander.weinert@dlr.de
[3] University of Liverpool, Liverpool L69 3BX, United Kingdom
martin.zimmermann@liverpool.ac.uk

Abstract. Recently, Dallal, Neider, and Tabuada studied a generalization of the classical game-theoretic model used in program synthesis, which additionally accounts for unmodeled intermittent disturbances. In this extended framework, one is interested in computing optimally resilient strategies, i.e., strategies that are resilient against as many disturbances as possible. Dallal, Neider, and Tabuada showed how to compute such strategies for safety specifications.

In this work, we compute optimally resilient strategies for a much wider range of winning conditions and show that they do not require more memory than winning strategies in the classical model. Our algorithms only have a polynomial overhead in comparison to the ones computing winning strategies. In particular, for parity conditions, optimally resilient strategies are positional and can be computed in quasipolynomial time.

1 Introduction

Reactive synthesis is an exciting and promising approach to solving a crucial problem, whose importance is ever-increasing due to ubiquitous deployment of embedded systems: obtaining correct and verified controllers for safety-critical systems. Instead of an engineer programming a controller by hand and then verifying it against a formal specification, synthesis automatically constructs a correct-by-construction controller from the given specification (or reports that no such controller exists).

Typically, reactive synthesis is modeled as a two-player zero-sum game on a finite graph that is played between the system, which seeks to satisfy the specification, and its environment, which seeks to violate it. Although this model is well understood, there are still multiple obstacles to overcome before synthesis can be realistically applied in practice. These obstacles include not only the high computational complexity of the problem, but also more fundamental ones. Among the most prohibitive issues in this regard is the need for a complete model of the interaction between the system and its environment, including an accurate model of the environment, the actions available to both players, as well as the effects of these actions.

This modeling task often places an insurmountable burden on engineers as the environments in which real-life controllers are intended to operate tend to be highly complex or not fully known at design time. Also, when a controller is deployed in the real world, a common source of errors is a mismatch between the controller's intended result of an action and the actual result. Such situations arise, e.g., in the presence of disturbances, when the effect of an action is not precisely known, or when the intended control action of the controller cannot be executed, e.g., when an actuator malfunctions. By a slight abuse of notation from control theory, such errors are subsumed under the generic term *disturbance* (cf. [12]).

To obtain controllers that can handle disturbances, one has to yield control over their occurrence to the environment. However, due to the antagonistic setting of the two-player zero-sum game, this would allow the environment to violate the specification by causing disturbances at will. Overcoming this requires the engineer to develop a realistic disturbance model, which is a highly complex task, as such disturbances are assumed to be rare events. Also, incorporating such a model into the game leads to a severe blowup in the size of the game, which can lead to intractability due to the high computational complexity of synthesis.

* Supported by the project "TriCS" (ZI 1516/1-1) of the German Research Foundation (DFG) and the Saarbrücken Graduate School of Computer Science

To overcome these fundamental difficulties, Dallal, Neider, and Tabuada [12] proposed a conceptually simple, yet powerful extension of infinite games termed “games with unmodeled intermittent disturbances”. Such games are played similarly to classical infinite games: two players, called Player 0 and Player 1, move a token through a finite graph, whose vertices are partitioned into vertices under the control of Player 0 and Player 1, respectively; the winner is declared based on a condition on the resulting play. In contrast to classical games, however, the graph is augmented with additional *disturbance edges* that originate in vertices of Player 0 and may lead to any other vertex. Moreover, the mechanics of how Player 0 moves is modified: whenever she moves the token, her move might be overridden, and the token instead moves along a disturbance edge. This change in outcome implicitly models the occurrence of a disturbance—the intended result of the controller and the actual result differ—but it is not considered to be antagonistic. Instead, the occurrence of a disturbance is treated as a rare event without any assumptions on frequency, distribution, etc. This approach very naturally models the kind of disturbances typically occurring in control engineering [12].

As a non-technical example, consider a scenario with three siblings, Alice, Bob, and Charlie, and their father, Donald. He repeatedly asks Alice to fetch water from a well using a jug made of clay. Alice has three ways to fulfill that task: she may get the water herself or she may delegate it to either Bob or Charlie. In a simple model, the outcome of these strategies is identical: Donald’s request for water is fulfilled. This is, however, unrealistic, as this model ignores the various ways that the execution of the strategies may go wrong. By modeling the situation as a game with disturbances, we obtain a more realistic model.

If Alice gets the jug herself, no disturbance can occur: she controls the outcome completely. If she delegates the task to Bob, the older of her brothers, Donald may get angry with her for not fulfilling her duties herself, which should not happen infinitely often. Finally, if she delegates the task to her younger brother Charlie, he might drop and break the jug, which would be disastrous for Alice.

These strategies can withstand different numbers of disturbances: the first strategy does not offer any possibility for disturbances, while infinitely many (a single) disturbance cause Alice to lose when using the second (the third) strategy. This model captures the intuition about Donald’s and Charlie’s behavior: both events occur non-antagonistically and their frequency is unknown.

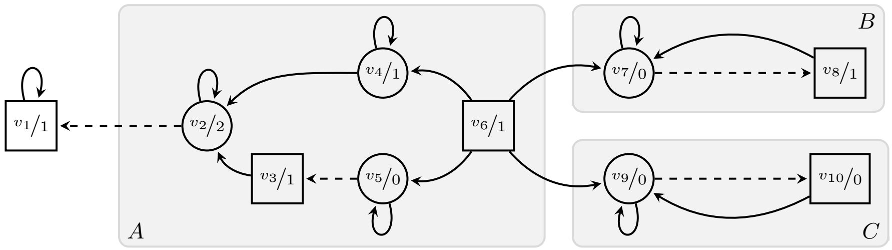


Fig. 1. A (max-) parity game with disturbances. Disturbance edges are drawn as dashed arrows. Vertices are labeled with both a name and a color. Vertices under control of Player 0 are drawn as circles, while vertices under control of Player 1 are drawn as rectangles.

This non-antagonistic nature of disturbances is different from existing approaches in the literature and causes many interesting phenomena that do not occur in the classical theory of infinite graph-based games. In Figure 1, we show an example of a parity game with disturbances that already exhibits some of these phenomena. In that parity game, vertices are labeled with non-negative integers, so-called colors, and Player 0 wins if the highest color seen infinitely often is even. For the sake of readability and conciseness, the parity game in Figure 1 does not model the example given in natural language above, but is rather constructed to showcase properties of games with disturbances.

Consider, for instance, vertex v_2. In the classical setting without disturbances, Player 0 wins every play reaching v_2 by simply looping in this vertex forever (since the highest color seen infinitely often is even). However, this is no longer true in the presence of disturbances: a disturbance in v_2 causes a play to proceed to vertex v_1, from which Player 0 can no longer win. In vertex v_7, Player 0 is in a similar, yet less severe situation: she wins every play with finitely many disturbances but loses if infinitely many disturbances occur. Finally, vertex v_9 falls into a third category: from this vertex, Player 0 wins every play even if infinitely many disturbances occur. In fact, disturbances partition the set of vertices from which Player 0 can guarantee to win into three disjoint regions (indicated as shaded boxes in Figure 1): (A) vertices from which she can win if at most a fixed finite number of disturbances occur, (B) vertices from which she can win if any finite number of disturbances occurs but not if infinitely many occur, and (C) vertices from which she can win even if infinitely many disturbances occur.

The observation above gives rise to a question that is both theoretically interesting and practically important: if Player 0 can tolerate different numbers of disturbances from different vertices, how should she play to be *resilient*[4] to as many disturbances as possible, i.e., to tolerate as many disturbances as possible but still win? Put slightly differently, disturbances induce an order on the space of winning strategies ("a winning strategy is better if it is more resilient"), and the natural problem is to compute optimally resilient winning strategies, yielding optimally resilient controllers. Note that this is in contrast to the classical theory of infinite games, where the space of winning strategies is unstructured.

Dallal, Neider, and Tabuada [12] have solved the problem of computing optimally resilient winning strategies for safety games. Their approach exploits the existence of maximally permissive winning strategies in safety games [2], which allows Player 0 to avoid "harmful" disturbance edges during a play. In games with more expressive winning conditions, however, this is no longer possible, as witnessed by vertex v_4 in the example of Figure 1: although Player 0 can avoid a disturbance edge by looping in v_4 forever, she needs to move to v_2 eventually in order to see an even color (otherwise she loses), thereby risking to lose if a disturbance occurs. In fact, the problem of constructing optimally resilient winning strategies for games other than safety games has been left open by Dallal, Neider, and Tabuada. In this work, we solve this problem for a large class of infinite games, including parity games.

Our Contributions. In Section 2, we introduce the concept of *resilience*, which captures for each vertex how many disturbances need to occur for Player 0 to lose. This generalizes the notion of determinacy and allows us to derive optimally resilient winning strategies.

Our main result is an algorithm for computing the resilience of vertices and optimally resilient winning strategies, which we present in Section 3. This algorithm requires the game to have a prefix-independent winning condition, to be determined, and all its subgames to be (classically) solvable. The latter two conditions are necessary, as resilience generalizes determinacy and computing optimally resilient strategies generalizes solving games. We discuss these assumptions in Section 4.

The algorithm uses solvers for the underlying game without disturbances as a subroutine, which it invokes a linear number of times on various subgames. For many winning conditions, the time complexity of our algorithm thus falls into the same complexity class as solving the original game without disturbances, e.g., we obtain a quasipolynomial algorithm for parity games with disturbances, which matches the currently best known upper bound for classical parity games. Stated differently, if the three assumptions above are satisfied by a winning condition, then computing the resilience and optimally resilient strategies is not harder than determining winning regions and winning strategies (ignoring a polynomial overhead).

Our algorithm requires the winning condition of the game to be prefix-independent. We also show how to overcome this restriction by generalizing the classical notion of game reductions to the setting of games with disturbances. As a consequence, via reductions, our algorithm can be applied to prefix-dependent winning conditions. We discuss details in Section 4.

Altogether, we have generalized the original result of Dallal, Neider, and Tabuada from safety games to all games which are algorithmically solvable, in particular all ω-regular games.

Finally, in Section 5, we discuss further phenomena that arise in the presence of disturbances. Amongst others, we illustrate how the additional goal of avoiding disturbances whenever possible affects the memory requirements of strategies. Similarly, we exhibit a tradeoff between resilience and the (semantic) quality of strategies in quantitative games. Moreover, we raise the question of how benevolent disturbances can be leveraged to recover from losing a play. However, an in-depth investigation of these phenomena is outside the scope of this paper and left for future work.

[4] We have deliberately chosen the term *resilience* so as to avoid confusion with the already highly ambiguous notions of *robustness* and *fault tolerance*.

2 Preliminaries

For notational convenience, we employ some ordinal notation à la von Neumann: the non-negative integers are defined inductively as $0 = \emptyset$ and $n+1 = n \cup \{n\}$. Now, the first limit ordinal is $\omega = \{0, 1, 2, \ldots\}$, the set of the non-negative integers. The next two successor ordinals are $\omega+1 = \omega \cup \{\omega\}$ and $\omega+2 = \omega+1 \cup \{\omega+1\}$. These ordinals are ordered by set inclusion, i.e., we have $0 < 1 < 2 < \cdots < \omega < \omega + 1 < \omega + 2$. For convenience of notation, we also denote the cardinality of ω by ω.

2.1 Infinite Games with Disturbances.

An *arena (with unmodeled disturbances)* $\mathcal{A} = (V, V_0, V_1, E, D)$ consists of a finite directed graph (V, E), a partition $\{V_0, V_1\}$ of V into the set of vertices V_0 of Player 0 (denoted by circles) and the set of vertices of Player 1 (denoted by squares), and a set $D \subseteq V_0 \times V$ of *disturbance edges* (denoted by dashed arrows). Note that only vertices of Player 0 have outgoing disturbance edges. We require that every vertex $v \in V$ has a successor v' with $(v, v') \in E$ to avoid finite plays.

A *play* in $\mathcal{A}$ is an infinite sequence $\rho = (v_0, b_0)(v_1, b_1)(v_2, b_2) \cdots \in (V \times \{0, 1\})^\omega$ such that $b_0 = 0$ and for all $j > 0$: $b_j = 0$ implies $(v_{j-1}, v_j) \in E$, and $b_j = 1$ implies $(v_{j-1}, v_j) \in D$. Hence, the additional bits b_j for $j > 0$ denote whether a standard or a disturbance edge has been taken to move from v_{j-1} to v_j, while b_0 is always zero. We say ρ starts in v_0. A play prefix $(v_0, b_0) \cdots (v_j, b_j)$ is defined similarly and ends in v_j. The number of disturbances in a play $\rho = (v_0, b_0)(v_1, b_1)(v_2, b_2) \cdots$ is $\#_d(\rho) = |\{j \in \omega \mid b_j = 1\}|$, which is either some $k \in \omega$ (if there are finitely many disturbances, namely k) or it is equal to ω (if there are infinitely many). A play ρ is *disturbance-free*, if $\#_d(\rho) = 0$.

A *game (with unmodeled disturbances)*, denoted by $\mathcal{G} = (\mathcal{A}, \mathrm{Win})$, consists of an arena $\mathcal{A} = (V, V_0, V_1, E, D)$ and a winning condition $\mathrm{Win} \subseteq V^\omega$. A play $\rho = (v_0, b_0)(v_1, b_1)(v_2, b_2) \cdots$ is winning for Player 0, if $v_0 v_1 v_2 \cdots \in \mathrm{Win}$, otherwise it is winning for Player 1. Hence, winning is oblivious to occurrences of disturbances. A winning condition Win is *prefix-independent* if for all $\rho \in V^\omega$ and all $w \in V^*$ we have $\rho \in \mathrm{Win}$ if and only if $w\rho \in \mathrm{Win}$. If Win is not prefix-independent, then it is called prefix-dependent.

In examples, we often use the parity condition, the canonical ω-regular winning condition. Let $\Omega \colon V \to \omega$ be a coloring of a set V of vertices. The (max-) parity condition

$$\mathrm{Parity}(\Omega) = \{v_0 v_1 v_2 \cdots \in V^\omega \mid \limsup \Omega(v_0)\Omega(v_1)\Omega(v_2) \cdots \text{ is even}\}$$

requires the maximal color occurring infinitely often during a play to be even. A game $(\mathcal{A}, \mathrm{Win})$ is a parity game, if $\mathrm{Win} = \mathrm{Parity}(\Omega)$ for some coloring Ω of the vertices of $\mathcal{A}$. In figures, we label a vertex v with color c by ${}^{v}/_{c}$.

In our proofs we make use of the safety condition

$$\mathrm{Safety}(U) = \{v_0 v_1 v_2 \cdots \in V^\omega \mid v_j \notin U \text{ for every } j \in \omega\}$$

for a given set $U \subseteq V$ of unsafe vertices. It requires Player 0 to only visit safe vertices, i.e., Player 1 wins a play if it visits at least one unsafe vertex. Note that due to notational convenience, we specify a safety condition by giving the *unsafe* vertices instead of the *safe* ones, i.e., $V \setminus U$, which is more common.

A strategy for Player $i \in \{0, 1\}$ is a function $\sigma \colon V^* V_i \to V$ such that $(v_j, \sigma(v_0 \cdots v_j)) \in E$ holds for every $v_0 \cdots v_j \in V^* V_i$. A play $(v_0, b_0)(v_1, b_1)(v_2, b_2) \cdots$ is consistent with σ, if $v_{j+1} = \sigma(v_0 \cdots v_j)$ for every j with $v_j \in V_i$ and $b_{j+1} = 0$, i.e., if the next vertex is the one prescribed by the strategy unless a disturbance edge is used.

Remark 1. A strategy σ does not have access to the bits indicating whether a disturbance occurred or not. However, this is not a restriction for Player 0: let $(v_0, b_0)(v_1, b_1)(v_2, b_2) \cdots$ be a play with $b_j = 1$ for some $j > 0$. We say that this disturbance is consequential (w.r.t. σ), if $v_j \neq \sigma(v_0 \cdots v_{j-1})$, i.e., if the disturbance transition (v_{j-1}, v_j) traversed by the play did not lead to the vertex the strategy prescribed. Such consequential disturbances can be detected by comparing the actual vertex v_j to σ's output $\sigma(v_0 \cdots v_{j-1})$. Hence, the bits b_j denoting consequential disturbances (w.r.t. σ) can be reconstructed by observing the sequence of vertices and by having access to the strategy σ.

On the other hand, inconsequential disturbances can just be ignored. In particular, the number of consequential disturbances is always at most the number of disturbances during each play.

2.2 Positional and Finite-state Strategies.

Fix a game $(\mathcal{A}, \mathrm{Win})$ with $\mathcal{A} = (V, V_0, V_1, E, D)$. A strategy σ for Player i is positional, if $\sigma(v_0 \cdots v_j) = \sigma(v_j)$ for all $v_0 \cdots v_j \in V^* V_i$, i.e., the output of σ only depends on the last vertex.

A memory structure for $\mathcal{A}$ is a triple $\mathcal{M} = (M, \mathrm{Init}, \mathrm{Upd})$ where M is a finite set of memory states, $\mathrm{Init}\colon V \to M$ is the initialization function, and $\mathrm{Upd}\colon M \times V \to M$ is the memory update function.

The update function can be extended to finite play prefixes: $\mathrm{Upd}^+(v) = \mathrm{Init}(v)$ and $\mathrm{Upd}^+(wv) = \mathrm{Upd}(\mathrm{Upd}^+(w), v)$ for $w \in V^+$ and $v \in V$. A next-move function $\mathrm{Nxt}\colon V_i \times M \to V$ for Player i has to satisfy $(v, \mathrm{Nxt}(v, m)) \in E$ for all $v \in V_i$ and all $m \in M$. It induces a strategy σ for Player i with memory $\mathcal{M}$ via $\sigma(v_0 \cdots v_j) = \mathrm{Nxt}(v_j, \mathrm{Upd}^+(v_0 \cdots v_j))$.

We say that a strategy σ is implementable by a memory structure $\mathcal{M}$, if there is a next-move function Nxt such that $\mathcal{M}$ and Nxt induce σ. If σ is implementable by some memory structure, then we call σ finite-state.

2.3 Infinite Games without Disturbances.

We can characterize the classical notion of infinite games, i.e., those without disturbances, (see, e.g., [18]) as a special case of games with disturbances. Let $\mathcal{G}$ be a game with vertex set V. A strategy σ for Player i in $\mathcal{G}$ is a winning strategy for her from $v \in V$, if every disturbance-free play that starts in v and that is consistent with σ is winning for Player i.

The winning region $\mathcal{W}_i(\mathcal{G})$ of Player i in $\mathcal{G}$ contains those vertices $v \in V$ from which Player i has a winning strategy. Thus, the winning regions of $\mathcal{G}$ are independent of the disturbance edges, i.e., we obtain the classical notion of infinite games. We say that Player i wins $\mathcal{G}$ from v, if $v \in \mathcal{W}_i(\mathcal{G})$. Solving a game amounts to determining its winning regions. Note that every game has disjoint winning regions. In contrast, a game is determined, if every vertex is in either winning region.

2.4 Resilient Strategies.

Let $\mathcal{G}$ be a game with vertex set V and let $\alpha \in \omega + 2$. A strategy σ for Player 0 in $\mathcal{G}$ is *α-resilient from $v \in V$* if every play ρ that starts in v, that is consistent with σ, and with $\#_d(\rho) < \alpha$, is winning for Player 0. Thus, a k-resilient strategy with $k \in \omega$ is winning even under at most $k - 1$ disturbances, an ω-resilient strategy is winning even under any finite number of disturbances, and an $(\omega + 1)$-resilient strategy is winning even under infinitely many disturbances.

Remark 2. Let v be a vertex.

1. Let $\alpha, \alpha' \in \omega + 2$ with $\alpha > \alpha'$. If a strategy is α-resilient from v, then it is also α'-resilient from v.
2. Every strategy is 0-resilient from v.
3. A strategy is 1-resilient from v if and only if it is winning for Player 0 from v.

We define the resilience of a vertex v of $\mathcal{G}$ as

$$r_{\mathcal{G}}(v) = \sup\{\alpha \in \omega + 2 \mid \text{Player 0 has an } \alpha\text{-resilient strategy for } \mathcal{G} \text{ from } v\}.$$

Note that the definition is not antagonistic, i.e., it is not defined via strategies of Player 1. Nevertheless, due to the remarks above, resilient strategies generalize winning strategies.

Lemma 1. *Let $\mathcal{G}$ be a game and v a vertex of $\mathcal{G}$.*

1. *$r_{\mathcal{G}}(v) > 0$ if and only if $v \in \mathcal{W}_0(\mathcal{G})$.*
2. *If $\mathcal{G}$ is determined, then $r_{\mathcal{G}}(v) = 0$ if and only if $v \in \mathcal{W}_1(\mathcal{G})$.*

Proof. 1.) The resilience of v is greater than zero if and only if Player 0 has a 1-resilient strategy from v due to Item 1 of Remark 2. The latter condition is equivalent to Player 0 having a winning strategy for $\mathcal{G}$ from v, i.e., to $v \in \mathcal{W}_0(\mathcal{G})$, due to Item 3 of Remark 2.

2.) Due to Items 1 and 3 of Remark 2, the resilience of v is zero if and only if Player 0 has no winning strategy for $\mathcal{G}$ from v, i.e., $v \notin \mathcal{W}_0(\mathcal{G})$. Due to determinacy, this is equivalent to $v \in \mathcal{W}_1(\mathcal{G})$. □

Note that determinacy is a necessary condition for Item 2. In an undetermined game, the vertices that are in neither winning region have resilience zero, due to Item 1, but are in particular not in $\mathcal{W}_1(\mathcal{G})$.

A strategy σ is *optimally resilient*, if it is $r_\mathcal{G}(v)$-resilient from every vertex v. Every such strategy is a *uniform* winning strategy for Player 0, i.e., a strategy that is winning from every vertex in her winning region. Hence, positional optimally resilient strategies can only exist in games which have uniform positional winning strategies for Player 0.

Our goal is to determine the mapping $r_\mathcal{G}$ and to compute an optimally resilient strategy.

3 Computing Optimally Resilient Strategies

To compute optimally resilient strategies, we first characterize the vertices of finite resilience in Subsection 3.1. All other vertices either have resilience ω or $\omega + 1$. To distinguish between these possibilities, we show how to determine the vertices with resilience $\omega + 1$ in Subsection 3.2. In Subsection 3.3, we show how to compute optimally resilient strategies using the results of the first two subsections. We only consider prefix-independent winning conditions in Subsections 3.1 and 3.3. In Section 4, we show how to overcome this restriction.

3.1 Characterizing Vertices of Finite Resilience

Our goal in this subsection is to characterize vertices with finite resilience in a game with prefix-independent winning condition, i.e., those vertices from which Player 0 can win even under $k - 1$ disturbances, but not under k disturbances, for some $k \in \omega$.

To illustrate our approach, consider the parity game in Figure 1 (on Page 2), which is determined and has a prefix-independent winning condition. The winning region of Player 1 only contains the vertex v_1. Thus, by Lemma 1, v_1 is the only vertex with resilience zero, every other vertex has a larger resilience.

Now, consider the vertex v_2, which has a disturbance edge leading into the winning region of Player 1. Due to this edge, v_2 has resilience at most one. This implies, as argued above, that v_2 has resilience precisely one. The unique disturbance-free play starting in v_1 is consistent with every strategy for Player 0 and violates the winning condition. Due to prefix-independence, prepending the disturbance edge does not change the winner and consistency with every strategy for Player 0. Hence, this play witnesses that v_2 has resilience at most one, while v_2 being in Player 0's winning region yields the matching lower bound. However, v_2 is the only vertex to which this reasoning applies. Now, consider v_3: from here, Player 1 can force a play to visit v_2 using a standard edge. Thus, v_3 has resilience one as well. Again, this is the only vertex to which this reasoning is applicable.

In particular, from v_4, Player 0 can avoid reaching the vertices for which we have already determined the resilience by using the self loop. However, this comes at a steep price for her: doing so results in a losing play, as the color of v_4 is odd. Thus, if she wants to have a chance at winning, she has to take a risk by moving to v_2, from which she has a 1-resilient strategy, i.e., one that is winning if no more disturbances occur. For this reason, v_4 has resilience one as well. The same reasoning applies to v_6: Player 1 can force the play to v_4 and from there Player 0 has to take a risk by moving to v_2.

The vertices v_3, v_4, and v_6 share the property that Player 1 can either enforce a play violating the winning condition or reach a vertex with already determined finite resilience. These three vertices are the only ones currently satisfying this property. They all have resilience one since Player 1 can enforce to reach a vertex of resilience one, but he cannot enforce reaching a vertex of resilience zero. Now, we can also determine the resilience of v_5: the disturbance edge from v_5 to v_3 witnesses it being two.

Afterwards, these two arguments no longer apply to new vertices: no disturbance edge leads from a vertex $v \in \{v_7, \dots, v_{10}\}$ to some vertex whose resilience is already determined and Player 0 has a winning strategy from each such v that additionally avoids vertices whose resilience is already determined. Thus, our reasoning cannot determine their resilience. This is consistent with our goal, as all four vertices have non-finite resilience: v_7 and v_8 have resilience ω and v_9 and v_{10} have resilience $\omega + 1$. Our reasoning here cannot distinguish these two values. We solve this problem in Subsection 3.2.

We now formalize the reasoning sketched above: starting from the vertices in Player 1's winning region having resilience zero, we use a so-called disturbance update and a risk update to determine all vertices of finite resilience. A disturbance update computes the resilience of vertices having a disturbance edge to a vertex whose resilience is already known (such as vertices v_2 and v_5 in the example of Figure 1).

A risk update, on the other hand, determines the resilience of vertices from which either Player 1 can force a visit to a vertex with known resilience (such as vertices v_3 and v_6) or Player 0 needs to move to such a vertex in order to avoid losing (e.g., vertex v_4). To simplify our proofs, we describe both as monotone operators updating partial rankings mapping vertices to ω, which might update already defined values. We show that applying these updates in alternation eventually yields a stable ranking that indeed characterizes the vertices of finite resilience.

Throughout this section, we fix a game $\mathcal{G} = (\mathcal{A}, \mathrm{Win})$ with $\mathcal{A} = (V, V_0, V_1, E, D)$ and prefix-independent $\mathrm{Win} \subseteq V^\omega$ satisfying the following condition: the game $(\mathcal{A}, \mathrm{Win} \cap \mathrm{Safety}(U))$ is determined for every $U \subseteq V$. We discuss this requirement in Section 4.

A ranking for $\mathcal{G}$ is a partial mapping $r\colon V \dashrightarrow \omega$. The domain of r is denoted by $\mathrm{dom}(r)$, its image by $\mathrm{im}(r)$. Let r and r' be two rankings. We say that r' refines r if $\mathrm{dom}(r') \supseteq \mathrm{dom}(r)$ and if $r'(v) \leq r(v)$ for all $v \in \mathrm{dom}(r)$. A ranking r is sound, if we have $r(v) = 0$ if and only if $v \in \mathcal{W}_1(\mathcal{G})$ (cf. Lemma 1).

Let r be a ranking for $\mathcal{G}$. We define the ranking r' as

$$r'(v) = \min\bigl(\{r(v)\} \cup \{r(v') + 1 \mid v' \in \mathrm{dom}(r) \text{ and } (v, v') \in D\}\bigr),$$

where $\{r(v)\} = \emptyset$ if $v \notin \mathrm{dom}(r)$, and $\min \emptyset$ is undefined (causing $r'(v)$ to be undefined). We call r' the *disturbance update* of r.

Lemma 2. *The disturbance update r' of a sound ranking r is sound and refines r.*

Proof. As the minimization defining $r'(v)$ ranges over a superset of $\{r(v)\}$, we have $r'(v) \leq r(v)$ for every $v \in \mathrm{dom}(r)$. This immediately implies refinement. From this inequality, we also obtain $r'(v) = 0$ for every $v \in \mathcal{W}_1(\mathcal{G})$, due to soundness of r. Finally, consider some $v \in \mathcal{W}_0(\mathcal{G})$. Then, $r(v) > 0$ by soundness of r. Thus, $r'(v) > 0$ as well, as both $r(v)$ and each $r(v') + 1$ are greater than zero. Altogether, r' is sound as well. □

Again, let r be a ranking for $\mathcal{G}$. For every $k \in \mathrm{im}(r)$ let

$$A_k = \mathcal{W}_1(\mathcal{A}, \mathrm{Win} \cap \mathrm{Safety}(\{v \in \mathrm{dom}(r) \mid r(v) \leq k\}))$$

be the winning region of Player 1 in the game where he either wins by reaching a vertex v with $r(v) \leq k$ or by violating the winning condition of $\mathcal{G}$. Now, define $r'(v) = \min\{k \mid v \in A_k\}$, where $\min \emptyset$ is again undefined. We call r' the *risk update* of r.

Lemma 3. *The risk update r' of a sound ranking r is sound and refines r.*

Proof. We show $r'(v) \leq r(v)$ for every $v \in \mathrm{dom}(r)$, which implies both refinement and $r'(v) = 0$ for every $v \in \mathcal{W}_1(\mathcal{G})$, as argued in the proof of Lemma 2.

Thus, let $v \in \mathrm{dom}(r)$. Trivially, $v \in \{v' \in \mathrm{dom}(r) \mid r(v') \leq r(v)\}$. Thus, Player 1 wins the game $(\mathcal{A}, \mathrm{Win} \cap \mathrm{Safety}(\{v' \in \mathrm{dom}(r) \mid r(v') \leq r(v)\}))$ from v by violating the safety condition right away. Hence, $v \in A_{r(v)}$ and thus $r'(v) \leq r(v)$.

To complete the proof of soundness of r', we just have to show $r'(v) > 0$ for every $v \in \mathcal{W}_0(\mathcal{G})$. Towards a contradiction, assume $r'(v) = 0$, i.e., $v \in A_0$. Thus, Player 1 has a strategy τ from v that ensures that either the winning condition is violated or that a vertex v' with $r(v') = 0$ is reached, i.e., $v' \in \mathcal{W}_1(\mathcal{G})$ by soundness of r. Hence, Player 1 has a winning strategy $\tau_{v'}$ for $\mathcal{G}$ from every such v'. This implies that he also has a winning strategy from v: play according to τ until a vertex v' with $r(v') = 0$ is reached. From there, mimic $\tau_{v'}$ when starting from v'. Every resulting disturbance-free play has a suffix that violates the winning condition Win. Thus, by prefix-independence, the whole play violates Win as well, i.e., it is winning for Player 1. Thus, $v \in \mathcal{W}_1(\mathcal{G})$, which yields the desired contradiction, as winning regions are always disjoint. □

Let r_0 be the unique sound ranking with domain $\mathcal{W}_1(\mathcal{G})$, i.e., r_0 maps exactly the vertices in Player 1's winning region to zero, all others are undefined. Starting with r_0, we inductively define a sequence of rankings $(r_j)_{j \in \omega}$ such that r_j for an odd (even) $j > 0$ is the disturbance (risk) update of r_{j-1}, i.e., we alternate between disturbance and risk updates.

Due to refinement, the r_j eventually stabilize, i.e., there is some j_0 such that $r_j = r_{j_0}$ for all $j \geq j_0$. Define $r^* = r_{j_0}$. Due to r_0 being sound and by Lemma 2 and Lemma 3, each r_j, and r^* in particular, is sound. If $v \in \mathrm{dom}(r^*)$, let j_v be the minimal j with $v \in \mathrm{dom}(r_j)$; otherwise, j_v is undefined.

Lemma 4. *If $v \in \mathrm{dom}(r^*)$, then $r_{j_v}(v) = r_j(v)$ for all $j \geq j_v$.*

Proof. We show the following stronger result for every $v \in \mathrm{dom}(r^*)$:

- If j_v is odd, then $r_j(v) = \frac{j_v+1}{2}$ for every $j \geq j_v$.
- If j_v is even, then $r_j(v) = \frac{j_v}{2}$ for every $j \geq j_v$.

The disturbance update increases the maximal rank by at most one and the risk update does not increase the maximal rank at all. Furthermore, due to refinement, the rank of v is set and then it cannot increase. Hence, we obtain $r_j(v) \leq \frac{j_v+1}{2}$ and $r_j(v) \leq \frac{j_v}{2}$ for odd and even j_v, respectively. In the remainder of the proof, we show a matching lower bound.

We say that a vertex v is updated to $k \in \omega$ in r_j if $r_j(v) = k$ and either $v \notin \mathrm{dom}(r_{j-1})$ or both $v \in \mathrm{dom}(r_{j-1})$ and $r_{j-1}(v) \neq k$ (here, r_{-1} is the unique ranking with empty domain). Note that as part of the proof, we have to show that the second case never occurs.

Now, we show the following by induction over j, which implies the matching lower bound.

- If j is odd, then no v is updated in r_j to some $k < \frac{j+1}{2}$.
- If j is even, then no v is updated in r_j to some $k < \frac{j}{2}$.

For $j = 0$, we have $\frac{j}{2} = 0$, and clearly, no vertex is assigned a negative rank by r_0. For $j = 1$ and $j' = 2$, we obtain $\frac{j+1}{2} = \frac{j'}{2} = 1$. As r_0, r_1, and r_2 are sound, neither r_1 nor r_2 update some v to zero.

Now, let $j > 2$ and first consider the case where j is odd. Towards a contradiction, assume that $v \in V$ is updated in r_j to some value less than $\frac{j+1}{2}$. Since j is odd, r_j is the disturbance update of r_{j-1}. Further, as v is updated in r_j, there exists some disturbance edge $(v, v') \in D$ such that $r_j(v) = r_{j-1}(v') + 1$. Thus, $r_{j-1}(v') < r_j(v) < \frac{j+1}{2}$, i.e., $r_{j-1}(v') \leq \frac{j+1}{2} - 2 = \frac{j-3}{2}$. First, we show $r_{j-3}(v') = r_{j-2}(v') = r_{j-1}(v')$, i.e., the rank of v' is stable during the last two updates.

First assume towards a contradiction $r_{j-2}(v') \neq r_{j-1}(v')$. Then, v' is updated in r_{j-1} to some rank of at most $\frac{j-3}{2}$, which is in turn smaller than $\frac{j-1}{2}$, violating the induction hypothesis for $j-1$. Hence, $r_{j-2}(v') = r_{j-1}(v')$. The same reasoning yields a contradiction to the assumption $r_{j-3}(v') \neq r_{j-2}(v')$. Thus, we indeed obtain $r_{j-3}(v') = r_{j-2}(v') = r_{j-1}(v')$.

Since r_{j-2} is the disturbance update of r_{j-3}, we obtain $r_{j-2}(v) \leq r_{j-3}(v') + 1 = r_{j-1}(v') + 1 = r_j(v)$. Due to refinement, we obtain $r_{j-2}(v) \geq r_j(v)$, i.e., altogether $r_{j-2}(v) = r_{j-1}(v) = r_j(v)$. The latter equality contradicts our initial assumption, namely v being updated in r_j to $r_j(v)$.

Now, consider the case where j is even. Again, assume towards a contradiction that $v \in V$ is updated in r_j to some value less than $\frac{j}{2}$. Since j is even, r_j is the risk update of r_{j-1}. Further, as v is updated in r_j, Player 1 wins the game $(\mathcal{A}, \mathrm{Win} \cap \mathrm{Safety}(U))$ from v, where $U = \{v' \in \mathrm{dom}(r_{j-1}) \mid r_{j-1}(v') \leq r_j(v)\}$. Hence, he has a strategy τ such that every play starting in v and consistent with τ either violates Win or eventually visits some vertex v' with $r_{j-1}(v') \leq r_j(v)$. We claim $r_{j-2}(v') = r_{j-1}(v')$ for all $v' \in U$.

Towards a contradiction, assume $r_{j-2}(v') \neq r_{j-1}(v')$ for some $v' \in U$. Note that we have $r_{j-1}(v') \leq r_j(v) < \frac{j}{2}$. Thus, v' is updated in r_{j-1} to some value strictly less than $\frac{j}{2}$, which contradicts the induction hypothesis for $j-1$. Hence, we indeed obtain $r_{j-2}(v') = r_{j-1}(v')$ for all $v' \in U$.

Thus, there are two types of vertices v' in U: those for which $r_{j-3}(v')$ is defined, which implies $r_{j-3}(v') = r_{j-1}(v')$ due to the induction hypothesis and refinement, and those where $r_{j-3}(v')$ is undefined, which implies $r_{j-2}(v') = r_{j-1}(v')$ due to the claim above.

We claim that Player 1 wins $(\mathcal{A}, \mathrm{Win} \cap \mathrm{Safety}(\{v'' \in \mathrm{dom}(r_{j-3}) \mid r_{j-3}(v'') \leq r_j(v)\}))$ from v, which implies $r_{j-2}(v) = r_j(v)$. This contradicts v being updated in r_j, our initial assumption.

To this end, we construct a strategy τ' from v that either violates Win or reaches a vertex v'' with $r_{j-3}(v'') \leq r_j(v)$ as follows. From v, τ' mimics τ until a vertex v' in U is reached (if it is at all). If v' is of the first type, then we have $r_{j-3}(v') = r_{j-1}(v') \leq r_j(v)$. If v' is of the second type, then v' is updated in r_{j-2} to some rank $r_{j-2}(v') = r_{j-1}(v') \leq r_j(v)$. As r_{j-2} is the risk update of r_{j-3}, Player 1 has a strategy $\tau_{v'}$ from v' that either violates Win or reaches a vertex v'' with $r_{j-3}(v'') \leq r_{j-2}(v') \leq r_j(v)$. Thus, starting in v', τ' mimics $\tau_{v'}$ from v' until such a vertex is reached (if it is reached at all). Thus, every play that starts in v and is consistent with τ' either violates Win (as it has a suffix that does) or reaches a vertex v'' with $r_{j-3}(v'') \leq r_j(v)$, which proves our claim. □

Lemma 4 implies that an algorithm computing the r_j does not need to implement the definition of the two updates as presented above, but can be optimized by taking into account that a rank is never updated once set. However, for the proofs below, the definition presented above is more expedient, as it gives stronger preconditions to rely on, e.g., Lemma 2 and 3 only hold for the definition presented above.

Also, from the proof of Lemma 4, we obtain an upper bound on the maximal rank of r^*. This in turn implies that the r_j stabilize quickly, as $r_j = r_{j+1} = r_{j+2}$ implies $r_j = r^*$.

Corollary 1. *We have* $\mathrm{im}(r^*) = \{0, 1, \dots, n\}$ *for some* $n < |V|$ *and* $r^* = r_{2|V|}$.

The main result of this section shows that r^* characterizes the resilience of vertices of finite resilience.

Lemma 5. *Let* r^* *be defined for* $\mathcal{G}$ *as above, and let* $v \in V$.

1. *If* $v \in \mathrm{dom}(r^*)$, *then* $r_{\mathcal{G}}(v) = r^*(v)$.
2. *If* $v \notin \mathrm{dom}(r^*)$, *then* $r_{\mathcal{G}}(v) \in \{\omega, \omega + 1\}$.

Proof. 1.) We show $r_{\mathcal{G}}(v) \le r^*(v)$ and $r_{\mathcal{G}}(v) \ge r^*(v)$.

"$\boldsymbol{r_{\mathcal{G}}(v) \le r^*(v)}$": An α-resilient strategy from v is also α'-resilient from v for every $\alpha' \le \alpha$. Thus, to prove

$$r_{\mathcal{G}}(v) = \sup\{\alpha \in \omega + 2 \mid \text{Player 0 has an } \alpha\text{-resilient strategy for } \mathcal{G} \text{ from } v\} \le r^*(v)$$

we just have to show that Player 0 has no $(r^*(v) + 1)$-resilient strategy from v. By definition, for every strategy σ for Player 0, we have to show that there is a play ρ starting in v and consistent with σ that has at most $r^*(v)$ disturbances and is winning for Player 1. So, fix an arbitrary strategy σ.

We define a play with the desired properties by constructing longer and longer finite prefixes before finally appending an infinite suffix. During the construction, we ensure that each such prefix ends in $\mathrm{dom}(r^*)$ in order to be able to proceed with our construction.

The first prefix just contains the starting vertex $(v, 0)$, i.e., the prefix does indeed end in $\mathrm{dom}(r^*)$. Now, assume we have produced a prefix $w(v', b')$ ending in some vertex $v' \in \mathrm{dom}(r^*)$, which implies that $j_{v'}$ is defined. We consider three cases:

If $j_{v'} = 0$, then $v' \in \mathcal{W}_1(\mathcal{G})$ by definition of r_0, i.e., Player 1 has a winning strategy τ from v. Thus, we extend $w(v', b')$ by the unique disturbance-free play that starts in v' and is consistent with σ and τ, without its first vertex. In that case, the construction of the infinite play is complete.

Second, if $j_{v'} > 0$ is odd, then v' received its rank $r^*(v')$ during a disturbance update. Hence, there is some v'' such that $(v', v'') \in D$ with $r^*(v') - 1 = r^*(v'')$. In this case, we extend $w(v', b')$ by such a vertex v'' to obtain the new prefix $w(v', b')(v'', 1)$, which satisfies the invariant, as v'' is in $\mathrm{dom}(r^*)$. Further, we have $j_{v''} < j_{v'}$ as the rank of v'' had to be defined in order to be considered during the disturbance update assigning a rank to v'.

Finally, if $j_{v'} > 0$ is even, then v' received its rank $r^*(v')$ during a risk update. We claim that Player 1 has a strategy $\tau_{v'}$ that guarantees one of the following outcomes from v': either the resulting play violates Win or it encounters a vertex v'' that satisfies $r^*(v'') \le r^*(v')$ and $j_{v''} < j_{v'}$ (which implies $v'' \ne v'$).

In that case, consider the unique disturbance-free play ρ' that starts in v' and is consistent with σ and the strategy $\tau_{v'}$ as above. If ρ' violates Win, then we extend $w(v', b')$ by ρ' without its first vertex. In that case, the construction of the infinite play is complete.

If ρ' does not violate Win, then we extend $w(v', b')$ by the prefix of ρ' without its first vertex and up to (and including) the first occurrence of a vertex v'' in ρ' satisfying the properties described above. Note that this again satisfies the invariant.

It remains to argue our claim: v' was assigned its rank $r^*(v') = r_{j_{v'}}(v')$ because it is in Player 1's winning region in the game with winning condition $\mathrm{Win} \cap \mathrm{Safety}(U)$, for

$$U = \{v'' \in \mathrm{dom}(r_{j_{v'}-1}) \mid r_{j_{v'}-1}(v'') \le r_{j_{v'}}(v')\}.$$

Hence, from v', Player 1 has a strategy to either violate the winning condition or to reach U. Thus, $r_{j_{v'}-1}(v'') = r^*(v'')$ for every $v'' \in \mathrm{dom}(r_{j_{v'}-1})$ yields $r^*(v'') \le r^*(v')$. Finally, we have $j_{v''} < j_{v'}$, as the rank of v' is assigned due to vertices in U already having ranks.

Note that only in two cases, we extend the prefix to an infinite play. In the other two cases, we just extend the prefix to a longer finite one. Thus, we first show that this construction always results in an infinite play. To this end, let $w_0(v_0, b_0)$ and $w_1(v_1, b_1)$ be two of the prefixes constructed above such that

$w_1(v_1, b_1)$ is an extension of $w_0(v_0, b_0)$. A simple induction proves $j_{v_1} < j_{v_0}$. Hence, as the value can only decrease finitely often, at some point an infinite suffix is added. Thus, we indeed construct an infinite play.

Finally, we have to show that the resulting play has the desired properties: by construction, the play starts in v and is consistent with σ. Furthermore, by construction, it has a disturbance-free suffix that violates Win. Thus, by prefix-independence, the whole play also violates Win. It remains to show that it has at most $r^*(v)$ disturbances. To this end, let $w_0(v_0, b_0)$ and $w_1(v_1, b_1)$ be two of the prefixes such that $w_1(v_1, b_1)$ is obtained by extending $w_0(v_0, b_0)$ once. If the extension consists of taking the disturbance edge $(v_0, v_1) \in D$, then we have $r^*(v_1) = r^*(v_0) + 1$. The only other possibility is the extension consisting of a finite play prefix that is consistent with the strategy τ_{v_0}. Then, by construction, we obtain $r^*(v_1) \leq r^*(v_0)$. So, there are at most $r^*(v)$ many disturbances in the play, as the current rank decreases with every disturbance edge and does not increase with the other type of extension, but is always non-negative.

“$\boldsymbol{r_\mathcal{G}(v) \geq r^*(v)}$”: Here, we construct a strategy σ_f for Player 0 that is $r^*(v)$-resilient from every $v \in \mathrm{dom}(r^*)$, i.e., from v, σ_f has to be winning even under $r^*(v) - 1$ disturbances. As every strategy is 0-resilient, we only have to consider those v with $r^*(v) > 0$.

The proof is based on the fact that r^* is both stable under the disturbance and under the risk update, i.e., the disturbance update and the risk update of r^* are r^*, which yields the following properties. Let $(v, v') \in D$ be a disturbance edge such that $r^*(v) > 0$. Then, we have $r^*(v') \geq r^*(v) - 1$. Also, for every $v \in \mathrm{dom}(r^*)$ with $r^*(v) > 0$, Player 0 has a winning strategy σ_v from v for the game $\mathcal{G}_v = (\mathcal{A}, \mathrm{Win} \cap \mathrm{Safety}(\{v' \in \mathrm{dom}(r^*) \mid r^*(v') < r^*(v)\}))$ (note the strict inequality). Here, we apply determinacy of $\mathcal{G}_v$, as the risk update is formulated in terms of Player 1's winning region.

Now, we define σ_f to always mimic a strategy σ_{v_cur} for some $v_\mathrm{cur} \in \mathrm{dom}(r^*)$, which is initialized by the starting vertex. The strategy σ_{v_cur} is mimicked until a consequential (w.r.t. σ_{v_cur}) disturbance edge is taken, say by reaching v'. In that case, the strategy σ_f discards the history of the play constructed so far, updates v_cur to v', and begins mimicking $\sigma_{v'}$. This is repeated ad infinitum.

Now, consider a play that starts in $\mathrm{dom}(r^*)$, is consistent with σ_f, and has less than $r^*(v)$ disturbances. The part up to the first consequential disturbance edge (if it exists at all) is consistent with σ_v. Now, let (v_0, v_0') be the corresponding disturbance edge. Then, we have $r^*(v_0) \geq r^*(v)$, as σ_v being a winning strategy for the safety condition never visits vertices with a rank smaller than $r^*(v)$. Thus, we conclude $r^*(v_0') \geq r^*(v_0) - 1 \geq r^*(v) - 1$. Similarly, the part between the first and the second consequential disturbance edge (if it exists at all) is consistent with $\sigma_{v_0'}$. Again, if (v_1, v_1') is the corresponding disturbance edge, then we have $r^*(v_1') \geq r^*(v_1) - 1 \geq r^*(v) - 2$. Continuing this reasoning shows that less than $r^*(v)$ (consequential) disturbance edges lead to a vertex v' with $r^*(v') > 0$, as the rank is decreased by at most one for every disturbance edge. The suffix starting in this vertex is disturbance-free and consistent with $\sigma_{v'}$. Hence, the suffix satisfies Win, i.e., by prefix-independence, the whole play satisfies Win as well. Thus, σ_f is indeed $r^*(v)$-resilient from every $v \in \mathrm{dom}(r^*)$.

2.) Let $X = V \setminus \mathrm{dom}(r^*)$. The disturbance update of r^* being r^* implies that every disturbance edge starting in X leads back to X. Similarly, the risk update of r^* being r^* implies $X = \mathcal{W}_0(\mathcal{G}_X)$ for $\mathcal{G}_X = (\mathcal{A}, \mathrm{Win} \cap \mathrm{Safety}(V \setminus X))$. Thus, from every $v \in X$, Player 0 has a strategy σ_v such that every disturbance-free play that starts in v and is consistent with σ_v satisfies the winning condition Win and never leaves X. Using these properties, we construct a strategy σ_ω that is ω-resilient from each $v \in X$. Thus, $r_\mathcal{G}(v) \in \{\omega, \omega + 1\}$.

The definition of the strategy σ_ω here is similar to the one above yielding the lower bound on the resilience. Again, σ_ω always mimics a strategy σ_{v_cur} for some $v_\mathrm{cur} \in X$, which is initialized by the starting vertex. The strategy σ_{v_cur} is mimicked until a consequential (w.r.t. σ_{v_cur}) disturbance edge is taken, say by reaching the vertex v'. In that case, the strategy σ_ω discards the history of the play constructed so far, updates v_cur to v', and begins mimicking $\sigma_{v'}$. This is repeated ad infinitum.

Due to the properties of the disturbance edges and the strategies σ_v, such a play never leaves X, even if disturbances occur. Furthermore, if only finitely many disturbances occur, then the resulting play has a disturbance-free suffix that starts in some $v' \in X$ and is consistent with $\sigma_{v'}$. As $\sigma_{v'}$ is winning from v' in $\mathcal{G}_X$, this suffix satisfies Win. Hence, by prefix-independence of Win, the whole play also satisfies Win. Thus, σ_ω is indeed an ω-resilient strategy from every $v \in X$. □

Combining Corollary 1 and Lemma 5, we obtain an upper bound on the resilience of vertices with finite resilience.

Corollary 2. *We have $r_\mathcal{G}(V) \cap \omega = \{0, 1, \ldots, n\}$ for some $n < |V|$.*

3.2 Characterizing Vertices of Resilience $\omega + 1$

Our goal in this subsection is to determine the vertices of resilience $\omega + 1$, i.e., those from which Player 0 can win even under an infinite number of disturbances. Intuitively, in this setting, we give Player 1 control over the disturbance edges, as he cannot execute more than infinitely many disturbances during a play. In the following, we prove this intuition to be correct. To this end, we transform the arena of the game so that at a vertex of Player 0, first Player 1 gets to chose whether he wants to take one of the disturbance edges and, if not, gives control to Player 0, who is then able to use a standard edge.

Given a game $\mathcal{G} = (\mathcal{A}, \mathrm{Win})$ with $\mathcal{A} = (V, V_0, V_1, E, D)$, we define the *rigged game* $\mathcal{G}_{\mathrm{rig}} = (\mathcal{A}', \mathrm{Win}')$ with $\mathcal{A}' = (V', V_0', V_1', E', D')$ such that $V' = V_0' \cup V_1'$ with $V_0' = \{\overline{v} \mid v \in V_0\}$ and $V_1' = V$, and $D' = \emptyset$. The set E' of edges is the union of the following sets:

- D: Player 1 uses a disturbance edge.
- $\{(v, \overline{v}) \mid v \in V_0\}$: Player 1 does not use a disturbance edge and yields control to Player 0.
- $\{(\overline{v}, v') \mid (v, v') \in E \text{ and } v \in V_0\}$: Player 0 has control and picks a standard edge.
- $\{(v, v') \mid (v, v') \in E \text{ and } v \in V_1\}$: Player 1 takes a standard edge.

Further, $\mathrm{Win}' = \{\rho \in (V')^\omega \mid h(\rho) \in \mathrm{Win}\}$ where h is the homomorphism induced by $h(v) = v$ and $h(\overline{v}) = \varepsilon$ for every $v \in V$.

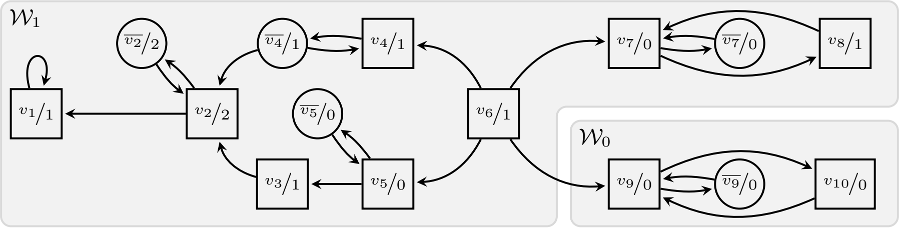


Fig. 2. The rigged game obtained for the game of Figure 1.

Figure 2 illustrates the construction of a rigged game for the example game of Figure 1 on Page 2 (note that the rigged game is also a parity game in this example). And indeed, the winning region of Player 0 corresponds to the vertices of resilience $\omega + 1$ in the game of Figure 1.

The following lemma formalizes the observation that $\mathcal{W}_0(\mathcal{G}_{\mathrm{rig}})$ characterizes the vertices of resilience $\omega + 1$ in $\mathcal{G}$. Note that we have no assumptions on $\mathcal{G}$ here.

Lemma 6. *Let v be a vertex of the game $\mathcal{G}$. Then, $v \in \mathcal{W}_0(\mathcal{G}_{\mathrm{rig}})$ if and only if $r_{\mathcal{G}}(v) = \omega + 1$.*

Proof. The proof consists of constructing mappings between play prefixes and plays in both games, which are then used to transfer strategies between the games. This is conceptually straightforward, but technical due to the presence of the bits indicating whether a disturbance occurred or not. These have to be reconstructed to obtain proper mappings.

"⇒": Let Player 0 win $\mathcal{G}_{\mathrm{rig}}$ from v, say with winning strategy σ'. We inductively translate play prefixes w in $\mathcal{G}$ into play prefixes $t'(w)$ in $\mathcal{G}_{\mathrm{rig}}$ that satisfy the following invariant: $t'((v_0, b_0) \cdots (v_j, b_j))$ starts in v_0 and ends in v_j.

For the induction start, we define $t'(v_0, b_0) = (v_0, 0)$; to define

$$t'((v_0, b_0) \cdots (v_j, b_j)(v_{j+1}, b_{j+1})),$$

we consider several cases:

- If $b_{j+1} = 1$, then $(v_j, v_{j+1}) \in D$, i.e., the play traverses the disturbance edge (v_j, v_{j+1}). This move is mimicked by defining

$$t'((v_0, b_0) \cdots (v_j, b_j)(v_{j+1}, b_{j+1})) = t'((v_0, b_0) \cdots (v_j, b_j)) \cdot (v_{j+1}, 0).$$

- If $b_{j+1} = 0$, i.e., $(v_j, v_{j+1}) \in E$, and $v_j \in V_0$, then the play did not traverse a disturbance edge and instead allowed Player 0 to pick a standard edge (v_j, v_{j+1}) to traverse. This move is mimicked by defining

  $$t'((v_0, b_0) \cdots (v_j, b_j)(v_{j+1}, b_{j+1})) = t'((v_0, b_0) \cdots (v_j, b_j)) \cdot (\overline{v_j}, 0) \cdot (v_{j+1}, 0).$$

- If $b_{j+1} = 0$, i.e., $(v_j, v_{j+1}) \in E$, and $v_j \in V_1$, then the play traversed the standard edge (v_j, v_{j+1}). This move is mimicked by defining

  $$t'((v_0, b_0) \cdots (v_j, b_j)(v_{j+1}, b_{j+1})) = t'((v_0, b_0) \cdots (v_j, b_j)) \cdot (v_{j+1}, 0).$$

Note that our invariant is satisfied in any case. Also, we lift t' to infinite plays by taking limits as usual.

Let d be the homomorphism induced by mapping $(v, b) \in V' \times \{0, 1\}$ to $v \in V'$, i.e., d removes the bits indicating the occurrence of disturbances. Using the translation t', we define a strategy σ for Player 0 in $\mathcal{G}$ via

$$\sigma(v_0 \cdots v_j) = \sigma'(d(t'((v_0, b_0) \cdots (v_j, b_j))) \cdot \overline{v_j}),$$

where $b_0 = 0$ and where for every $j' > 0$, $b_{j'} = 1$ if and only if $v_{j'} \neq \sigma(v_0 \cdots v_{j'-1})$, i.e., we reconstruct the consequential disturbances. A straightforward induction shows that for every play $\rho = (v_0, b_0)(v_1, b_1)(v_2, b_2) \cdots$ in $\mathcal{G}$ that is consistent with σ, the play $t'(\rho)$ is consistent with σ'. Hence, $t'(\rho) \in \mathrm{Win}'$ for every ρ starting in v. Furthermore, we have $h(t'(\rho)) = v_0 v_1 v_2 \cdots \in \mathrm{Win}$, as $t'(\rho) \in \mathrm{Win}'$. Thus, $\rho = (v_0, b_0)(v_1, b_1)(v_2, b_2) \cdots$ is winning for Player 0. As we have no restriction on the number of disturbances in ρ, σ is $(\omega + 1)$-resilient from v. Thus, $r_{\mathcal{G}}(v) = \omega + 1$.

"$\Leftarrow$": Now, let $r_{\mathcal{G}}(v) = \omega + 1$, i.e., Player 0 has an $(\omega + 1)$-resilient strategy σ from v in $\mathcal{G}$. This time, we inductively define a translation t of play prefixes in $\mathcal{G}_{\mathrm{rig}}$ into play prefixes in $\mathcal{G}$. Here, it suffices to consider those prefixes that start and end in V_1'. For these, we satisfy the following invariant: if w starts in v_0 and ends in v_j, then $t(w)$ starts in v_0 and ends in v_j as well. Note that $\mathcal{G}_{\mathrm{rig}}$ has no disturbance edges. Hence, the bits indicating whether such an edge has been traversed are always zero in plays of $\mathcal{G}_{\mathrm{rig}}$. Thus, we define $t(v_0, 0) = (v_0, 0)$ and consider several cases for the inductive step:

- First, assume we have a prefix of the form $(v_0, 0) \cdots (v_j, 0)(v_{j+1}, 0)$ for some $v_j \in V_0$, i.e., Player 1's move simulates the disturbance edge $(v_j, v_{j+1}) \in D$. Then, we define

  $$t((v_0, 0) \cdots (v_j, 0)(v_{j+1}, 0)) = t((v_0, 0) \cdots (v_j, 0)) \cdot (v_{j+1}, 1) \ .$$

- Next, assume we have a prefix of the form $(v_0, 0) \cdots (v_j, 0)(v_{j+1}, 0)$ for some $v_j \in V_1$, i.e., Player 1's move simulates the standard edge $(v_j, v_{j+1}) \in E$. Then, we define

  $$t((v_0, 0) \cdots (v_j, 0)(v_{j+1}, 0)) = t((v_0, 0) \cdots (v_j, 0)) \cdot (v_{j+1}, 0) \ .$$

- Finally, the last case is a prefix of the form $(v_0, 0) \cdots (v_j, 0)(\overline{v_j}, 0)(v_{j+1}, 0)$ for some $v_j \in V_0$, i.e., Player 0's move simulates the standard edge $(v_j, v_{j+1}) \in E$. Then, we define

  $$t((v_0, 0) \cdots (v_j, 0)(\overline{v_j}, 0)(v_{j+1}, 0)) = t((v_0, 0) \cdots (v_j, 0)) \cdot (v_{j+1}, 0) \ .$$

The invariant is satisfied in any case. Also, we can again lift t to infinite plays via limits.

Now, let d be the homomorphism induced by mapping $(v, b) \in V \times \{0, 1\}$ to $v \in V$, i.e., d again deletes the bits indicating the occurrence of disturbances. Then, we define a strategy σ' for Player 0 in $\mathcal{G}_{\mathrm{rig}}$ via

$$\sigma'(v_0 \cdots v_j \overline{v_j}) = \sigma(d(t((v_0, 0) \cdots (v_j, 0)))).$$

A straightforward induction shows that for every play ρ that is consistent with σ', the play $t(\rho)$ is consistent with σ. Hence, if ρ starts in v, then $t(\rho)$ satisfies the winning condition, as σ is $(\omega + 1)$-resilient from v. Let $t(\rho) = (v_0, b_0)(v_1, b_1)(v_2, b_2) \cdots$. Then, $v_0 v_1 v_2 \cdots \in \mathrm{Win}$. Now, $h(\rho) = v_0 v_1 v_2 \cdots$ implies $\rho \in \mathrm{Win}'$. Thus, σ' is a winning strategy for Player 0 from v. □

With an adaption of the rigged game, one can also directly characterize the vertices with resilience ω. However, since our algorithm and the rigged game already provide an indirect characterization, we do not present this construction here.

Furthermore, the proof of Lemma 6 also yields the preservation of positional and finite-state strategies. To this end, consider the first implication proved above. If σ is positional (finite-state), then σ' is positional (finite-state) as well. Thus, applying both implications yields the following corollary.

Corollary 3. *Let $\mathcal{G}$ and $\mathcal{G}_{\mathrm{rig}}$ be defined as above and v a vertex of $\mathcal{G}$.*

1. *Assume Player 0 has a positional winning strategy for $\mathcal{G}_{\mathrm{rig}}$ from v. Then, Player 0 has an $(\omega + 1)$-resilient positional strategy for $\mathcal{G}$ from v.*
2. *Assume Player 0 has a finite-state winning strategy for $\mathcal{G}_{\mathrm{rig}}$ from v. Then, Player 0 has an $(\omega + 1)$-resilient finite-state strategy (of the same size) for $\mathcal{G}$ from v.*

3.3 Computing Optimally Resilient Strategies

This subsection is concerned with computing the resilience and optimally resilient strategies. Here, we focus on positional and finite-state strategies, which are sufficient for the majority of winning conditions in the literature. Nevertheless, it is easy to see that our framework is also applicable to infinite-state strategies.

In the proof of Lemma 5, we construct strategies σ_f and σ_ω such that σ_f is $r_{\mathcal{G}}(v)$-resilient from every v with $r_{\mathcal{G}}(v) \in \omega$ and such that σ_ω is ω-resilient from every v with $r_{\mathcal{G}}(v) \geq \omega$. Both strategies are obtained by combining winning strategies for some game $(\mathcal{A}, \mathrm{Win} \cap \mathrm{Safety}(U))$. However, even if these winning strategies are positional, the strategies σ_f and σ_ω are in general not positional. Nonetheless, we show in the proof of Theorem 1 that such positional winning strategies and a positional one for $\mathcal{G}_{\mathrm{rig}}$ can be combined into a single positional optimally resilient strategy.

Recall the requirements from Subsection 3.1 for a game $(\mathcal{A}, \mathrm{Win})$: Win is prefix-independent and the game $\mathcal{G}_U$ is determined for every $U \subseteq V$, where we write $\mathcal{G}_U$ for the game $(\mathcal{A}, \mathrm{Win} \cap \mathrm{Safety}(U))$ for some $U \subseteq V$. To prove the results of this subsection, we need to impose some additional effectiveness requirements: we require that each game $\mathcal{G}_U$ and the rigged game $\mathcal{G}_{\mathrm{rig}}$ can be effectively solved. Also, we first assume that Player 0 has positional winning strategies for each of these games, which have to be effectively computable as well. We discuss the severity of these requirements in Section 4.

Theorem 1. *Let $\mathcal{G}$ satisfy all the above requirements. Then, the resilience of $\mathcal{G}$'s vertices and a positional optimally resilient strategy can be effectively computed.*

To prove this result, we refine the following standard technique that combines positional winning strategies for games with prefix-independent winning conditions.

Assume we have a positional strategy σ_v for every vertex v in some set $W \subseteq V$ such that σ_v is winning from v. Furthermore, let R_v be the set of vertices visited by plays that start in v and are consistent with σ_v. Also, let $m(v) = \min_{\prec}\{v' \in V \mid v \in R_{v'}\}$ for some strict total ordering $\prec$ of W. Then, the positional strategy σ defined by $\sigma(v) = \sigma_{m(v)}(v)$ is winning from each $v \in W$, as along every play that starts in some $v \in W$ and is consistent with σ, the value of the function m cannot increase. Thus, after it has stabilized, the remaining suffix is consistent with some strategy $\sigma_{v'}$. Hence, the suffix is winning for Player 0 and prefix-independence implies that the whole play is winning for her as well.

Here, we have to adapt this reasoning to respect the resilience of the vertices and to handle disturbance edges. Also, we have to pay attention to vertices of resilience $\omega + 1$, as plays starting in such vertices have to be winning under infinitely many disturbances.

Proof (of Theorem 1). The effective computability of the resilience follows from the effectiveness requirements on $\mathcal{G}$: to compute the ranking r^*, it suffices to compute the disturbance and risk updates. The former are trivially effective while the effectiveness of the latter ones follows from our assumption. Lemma 5 shows that r^* correctly determines the resilience of all vertices with finite resilience. Finally, by solving the rigged game, we also determine the resilience of the remaining vertices (Lemma 6). Again, this game can be solved due to our assumption.

Thus, it remains to show how to compute a positional optimally resilient strategy. To this end, we compute a positional strategy σ_v for every v satisfying the following:

- For every $v \in V$ with $r_{\mathcal{G}}(v) \in \omega \setminus \{0\}$, the strategy σ_v is winning for Player 0 from v for the game $(\mathcal{A}, \mathrm{Win} \cap \mathrm{Safety}(\{v' \in V \mid r_{\mathcal{G}}(v') < r_{\mathcal{G}}(v)\}))$. We have shown the existence of such a strategy in the proof of Item 1 of Lemma 5.
- For every $v \in V$ with $r_{\mathcal{G}}(v) = \omega$, the strategy σ_v is winning for Player 0 from v for the game $(\mathcal{A}, \mathrm{Win} \cap \mathrm{Safety}(\{v' \in V \mid r_{\mathcal{G}}(v') \in \omega\}))$. We have shown the existence of such a strategy in the proof of Item 2 of Lemma 5.

- For every $v \in V$ with $r_{\mathcal{G}}(v) = \omega + 1$, the strategy σ_v is $(\omega + 1)$-resilient from v. The existence of such a strategy follows from Item 1 of Corollary 3, as we assume Player 0 to win $\mathcal{G}_{\mathrm{rig}}$ with positional strategies.
- For every $v \in V$ with $r_{\mathcal{G}}(v) = 0$, we fix an arbitrary positional strategy σ_v for Player 0.

Furthermore, we fix a strict linear order $\prec$ on V such that $v \prec v'$ implies $r_{\mathcal{G}}(v) \leq r_{\mathcal{G}}(v')$, i.e., we order the vertices by ascending resilience. For $v \in V$ with $r_{\mathcal{G}}(v) \neq \omega + 1$, let R_v be the set of vertices reachable via disturbance-free plays that start in v and are consistent with σ_v. On the other hand, for $v \in V$ with $r_{\mathcal{G}}(v) = \omega + 1$, let R_v be the set of vertices reachable via plays with arbitrarily many disturbances that start in v and are consistent with σ_v.

We claim $R_v \subseteq \{v' \in V \mid r_{\mathcal{G}}(v') \geq r_{\mathcal{G}}(v)\}$ for every $v \in V$ $(*)$. For v with $r_{\mathcal{G}}(v) \neq \omega + 1$ this follows immediately from the choice of σ_v. Thus, let v with $r_{\mathcal{G}}(v) = \omega + 1$. Assume σ_v reaches a vertex v' of resilience $r_{\mathcal{G}}(v') \neq \omega + 1$. Then, there exists a play ρ' starting in v' that is consistent with σ_v, has less than $\omega + 1$ many disturbances and is losing for Player 0. Thus, the play obtained by first taking the play prefix to v' and then appending ρ' without its first vertex yields a play starting in v, consistent with σ_v, but losing for Player 0. This play witnesses that σ_v is not $(\omega + 1)$-resilient from v, which contradicts our assumption and thus concludes the proof of the claim for the case $r_{\mathcal{G}}(v) = \omega + 1$.

Let $m \colon V \to V$ be given as $m(v) = \min_{\prec}\{v' \in V \mid v \in R_{v'}\}$ and define the positional strategy σ as $\sigma(v) = \sigma_{m(v)}(v)$. By our assumptions, σ can be effectively computed. It remains to show that it is optimally resilient.

To this end, we apply the following two properties of edges (v, v') that may appear during a play that is consistent with σ, i.e., we either have $v \in V_0$ and $\sigma(v) = v'$ (which implies $(v, v') \in E$), or $v \in V_1$ and $(v, v') \in E$, or $v \in V_0$ and $(v, v') \in D$:

1. If $(v, v') \in E$, then we have $r_{\mathcal{G}}(v) \leq r_{\mathcal{G}}(v')$ and $m(v) \geq m(v')$. The first property follows from minimality of $m(v)$ and $(*)$ while the second follows from the definition of R_v.
2. If $(v, v') \in D$, then we distinguish several subcases, which all follow immediately from the definition of resilience:
   - If $r_{\mathcal{G}}(v) \in \omega$, then $r_{\mathcal{G}}(v') \geq r_{\mathcal{G}}(v) - 1$.
   - If $r_{\mathcal{G}}(v) = \omega$, then $r_{\mathcal{G}}(v') = \omega$, and
   - If $r_{\mathcal{G}}(v) = \omega + 1$, then $r_{\mathcal{G}}(v') = \omega + 1$ and $m(v) \geq m(v')$ (here, the second property follows from the definition of R_v for v with $r_{\mathcal{G}}(v) = \omega + 1$, which takes disturbance edges into account).

Now, consider a play $\rho = (v_0, b_0)(v_1, b_1)(v_2, b_2) \cdots$ that is consistent with σ. If $r_{\mathcal{G}}(v_0) = 0$ then we have nothing to show, as every strategy is 0-resilient from v.

Now, assume $r_{\mathcal{G}}(v_0) \in \omega \setminus \{0\}$. We have to show that if ρ has less than $r_{\mathcal{G}}(v_0)$ disturbances, then it is winning for Player 0. An inductive application of the above properties shows that in that case the last disturbance edge leads to a vertex of non-zero resilience. Furthermore, as the values $m(v_j)$ are only decreasing afterwards, they have to stabilize at some later point. Hence, there is some suffix of ρ that starts in some v' with non-zero resilience and that is consistent with the strategy $\sigma_{v'}$. Thus, the suffix is winning for Player 0 by the choice of $\sigma_{v'}$ and prefix-independence implies that ρ is winning for her as well.

Next, assume $r_{\mathcal{G}}(v_0) = \omega$. We have to show that if ρ has a finite number of disturbances, then it is winning for Player 0. Again, an inductive application of the above properties shows that in that case the last disturbance edge leads to a vertex of resilience ω or $\omega + 1$. Afterwards, the values $m(v_j)$ stabilize again. Hence, there is some suffix of ρ that starts in some v' with non-zero resilience and that is consistent with the strategy $\sigma_{v'}$. Thus, the suffix is winning for Player 0 by the choice of $\sigma_{v'}$ and prefix-independence implies that ρ is winning for her as well.

Finally, assume $r_{\mathcal{G}}(v_0) = \omega + 1$. Then, the above properties imply that ρ only visits vertices with resilience $\omega + 1$ and that the values $m(v_j)$ eventually stabilize. Hence, there is a suffix of ρ that is consistent with some $(\omega + 1)$-resilient strategy $\sigma_{v'}$, where v' is the first vertex of the suffix. Hence, the suffix is winning for Player 0, no matter how many disturbances occur. This again implies that ρ is winning for her as well. □

The algorithm determining the vertices' resilience and a positional optimally resilient strategy first computes r^* and the winner of the rigged game. This yields the resilience of $\mathcal{G}$'s vertices. Furthermore, the strategy is obtained by combining winning strategies for the games $\mathcal{G}_U$ and for the rigged game as explained above.

Next, we analyze the complexity of the algorithm sketched above in some more detail. The inductive definition of the r_j can be turned into an algorithm computing r^* (using the results of Lemma 4 to optimize the naive implementation), which has to solve $\mathcal{O}(|V|)$ many games (and compute winning strategies for some of them) with winning condition $\mathrm{Win} \cap \mathrm{Safety}(U)$. Furthermore, the rigged game, which is of size $\mathcal{O}(|V|)$, has to be solved and winning strategies have to be determined. Thus, the overall complexity is in general dominated by the complexity of solving these tasks.

We explicitly state one complexity result for the important case of parity games, using the fact that each of these games is then a parity game as well. Also, we use a quasipolynomial time algorithm for solving parity games [8,15,20,22] to solve the games $\mathcal{G}_U$ and $\mathcal{G}_{\mathrm{rig}}$.

Theorem 2. *Optimally resilient strategies in parity games are positional and can be computed in quasipolynomial time.*

Using similar arguments, one can also analyze games where positional strategies do not suffice. As above, assume $\mathcal{G}$ satisfies the same assumptions on determinacy and effectiveness, but only require that Player 0 has finite-state winning strategies for each game with winning condition $(\mathcal{A}, \mathrm{Win} \cap \mathrm{Safety}(U))$ and for the rigged game $\mathcal{G}_{\mathrm{rig}}$. Then, one can show that she has a finite-state optimally resilient strategy. In fact, by reusing memory states, one can construct an optimally resilient strategy that it is not larger than any constituent strategy.

4 Discussion

In this section, we discuss the assumptions required to be able to compute positional (finite-state) optimally resilient strategies with the algorithm presented in Section 3. Here, we only consider the case of positional strategies. The case of finite-state strategies is analogous.

To this end, fix a game $\mathcal{G} = (\mathcal{A}, \mathrm{Win})$ with vertex set V and recall that $\mathcal{G}_{\mathrm{rig}}$ is the corresponding rigged game and that we defined $\mathcal{G}_U = (\mathcal{A}, \mathrm{Win} \cap \mathrm{Safety}(U))$ for $U \subseteq V$. Now, the assumptions on $\mathcal{G}$ that need to be satisfied for Theorem 1 to hold are as follows:

1. The game $\mathcal{G}_U$ is determined for every $U \subseteq V$.
2. Player 0 has a positional winning strategy from every vertex in her winning regions in the $\mathcal{G}_U$ and in the game $\mathcal{G}_{\mathrm{rig}}$.
3. Each $\mathcal{G}_U$ and the game $\mathcal{G}_{\mathrm{rig}}$ can be effectively solved and positional winning strategies can be effectively computed for each such game.
4. Win is prefix-independent.

First, consider the determinacy assumption. For $W \subseteq V$ let $\mathcal{A} \setminus W$ denote the arena obtained from $\mathcal{A}$ by removing all vertices from W, as well as all edges from or to vertices in W. It is easy to show that $\mathcal{A} \setminus W$ has no terminal vertices, if W is the winning region of Player 1 in a safety game played in $\mathcal{A}$. Now, it is straightforward to show

$$\mathcal{W}_0(\mathcal{G}_U) = \mathcal{W}_0(\mathcal{A} \setminus W, \mathrm{Win} \cap (V \setminus W)^\omega)$$

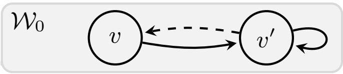
and

$$\mathcal{W}_1(\mathcal{G}_U) = W \cup \mathcal{W}_1(\mathcal{A} \setminus W, \mathrm{Win} \cap (V \setminus W)^\omega)$$

where $W = \mathcal{W}_1(\mathcal{A}, \mathrm{Safety}(U))$. Thus, one can remove the winning region of Player 1 in the safety game and then consider the subgame of $\mathcal{G}$ played in Player 0's winning region of the safety game. Thus, all subgames of $\mathcal{G}$ being determined suffices for the determinacy requirement being satisfied. The winning conditions one typically studies, e.g., parity and in fact all Borel ones [24], satisfy this property.

The next requirement concerns the existence of positional winning strategies for the games $\mathcal{G}_U$ and $\mathcal{G}_{\mathrm{rig}}$. For the $\mathcal{G}_U$, this requirement is satisfied if Player 0 has positional winning strategies for all subgames of $\mathcal{G}$, as argued above. As every positional optimally resilient strategy is also a winning strategy in a certain subgame, this condition is necessary. Now, consider $\mathcal{G}_{\mathrm{rig}}$, whose winning condition can be written as $h^{-1}(\mathrm{Win})$ for the homomorphism h from Subsection 3.2. The winning conditions one typically studies, e.g., the Borel ones, are closed w.r.t. such supersequences. If $\mathcal{G}$ is from a class of winning conditions that allows for positional winning strategies for Player 0, then this class typically also contains $\mathcal{G}_{\mathrm{rig}}$. Also, the assumption on the effective solvability and computability of positional strategies is obviously necessary, as we solve a more general problem when determining optimally resilient strategies.

Finally, let us consider prefix-independence. If the winning condition is not prefix-independent, then the algorithm presented in Section 3 does not compute the resilience of vertices correctly. In fact, recall that a winning condition Win is prefix-independent if, for all plays ρ and all play prefixes w, we have $\rho \in \mathrm{Win}$ if and only if $w\rho \in \mathrm{Win}$. We show that neither implication of this equivalence suffices on its own for the algorithm from Section 3 to compute the correct resilience of vertices.

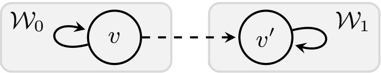


$$\mathrm{Win}_k = \{v_0v_1v_2\cdots \in V^\omega \mid |\{j \mid v_j = v\}| \leq k\}$$

Fig. 3. Counterexample for requiring only the implication from right to left from the definition of prefix-independence for the computation of resilience.

First, consider the family $\mathcal{G}_k = (\mathcal{A}, \mathrm{Win}_k)$ of games shown in Figure 3. In $\mathcal{G}_k$, it is the goal of Player 0 to avoid more than k visits to v. Hence, for all plays ρ and all play prefixes w we have that $w\rho \in \mathrm{Win}$ implies $\rho \in \mathrm{Win}$.

In each of the $\mathcal{G}_k$, a visit to v only occurs via a disturbance or if the initial vertex is v. Hence, we have $r_{\mathcal{G}_k}(v) = k$ and $r_{\mathcal{G}_k}(v') = k+1$. If we apply the algorithm from Section 3, however, the initial ranking function r_0 has an empty domain, since we have $\mathcal{W}_1(\mathcal{G}_k) = \emptyset$. Thus, the computation of the r_j immediately stabilizes, yielding r^* with empty domain. Hence, that algorithm does, in general, not compute the correct resilience if only the implication from right to left from the definition of prefix-independence is satisfied.



$$\mathrm{Win} = V^\omega \setminus \{(v')^\omega\}$$

Fig. 4. Counterexample for requiring only the implication from left to right from the definition of prefix-independence for the computation of resilience.

Conversely, consider the game $\mathcal{G}$ shown in Figure 4. The winning condition of this game satisfies that, for all play prefixes w and all plays ρ, we have that $\rho \in \mathrm{Win}$ implies $w\rho \in \mathrm{Win}$. If we apply the algorithm from Section 3, however, the initial ranking r_0 has the domain $\{v'\}$ with $r_0(v') = 0$, due to $\mathcal{W}_1(\mathcal{G}) = \{v'\}$. The disturbance update of r_0 then yields the ranking r_1 with $r_1(v) = 1$ due to the single disturbance edge of $\mathcal{G}$ and with $r_1(v') = 0$. At this point, the rankings stabilize and we obtain $r^* = r_1$.

While we indeed have $r_{\mathcal{G}}(v') = 0 = r^*(v')$, we furthermore have $r_{\mathcal{G}}(v) = \omega + 1 \neq r^*(v)$, as every play starting in vertex v satisfies the winning condition. Hence, this example showcases that the implication from left to right from the definition of prefix-independence also does not suffice for the algorithm from Section 3 to correctly compute the resilience. Thus, we indeed require full prefix-independence of the winning condition as a precondition for the correctness of that algorithm.

In the following subsection, we show that one can still leverage our algorithm from Section 3 in order to compute the resilience of a wide range of games with prefix-dependent winning conditions. To this end, we extend the framework of game reductions to games with disturbances, in such a way that the existence of α-resilient strategies is preserved. Using this framework shows that Player 0 has a finite-state optimally resilient strategy in every game with ω-regular winning condition.

4.1 Prefix-dependent Winning Conditions

We begin by introducing some notation regarding game reductions. An arena $\mathcal{A} = (V, V_0, V_1, E, D)$ and a memory structure $\mathcal{M} = (M, \mathrm{Init}, \mathrm{Upd})$ for $\mathcal{A}$ induce the expanded arena $\mathcal{A} \times \mathcal{M} = (V \times M, V_0 \times M, V_1 \times M, E', D')$ where E' is defined via $((v, m), (v', m')) \in E'$ if and only if $(v, v') \in E$ and $\mathrm{Upd}(m, v') = m'$.

The disturbance edges D' are defined analogously, i.e., $((v, m), (v', m')) \in D'$ if and only if $(v, v') \in D$ and $\mathrm{Upd}(m, v') = m'$. Every play $(v_0, b_0)(v_1, b_1)(v_2, b_2)\cdots$ in $\mathcal{A}$ has a unique extended play

$$\mathrm{ext}(\rho) = ((v_0, m_0), b_0)((v_1, m_1), b_1)((v_2, m_2), b_2)\cdots$$

in $\mathcal{A} \times \mathcal{M}$ defined by $m_0 = \mathrm{Init}(v_0)$ and $m_{j+1} = \mathrm{Upd}(m_j, v_{j+1})$, i.e., $m_j = \mathrm{Upd}^+(v_0 \cdots v_j)$. Play prefixes are translated analogously.

Remark 3. Let ρ be a play in $\mathcal{G}$. Then, $\#_d(\rho) = \#_d(\mathrm{ext}(\rho))$.

A game $\mathcal{G} = (\mathcal{A}, \mathrm{Win})$ is reducible to $\mathcal{G}' = (\mathcal{A}', \mathrm{Win}')$ via $\mathcal{M}$, written $\mathcal{G} \leq_{\mathcal{M}} \mathcal{G}'$, if $\mathcal{A}' = \mathcal{A} \times \mathcal{M}$ and every play ρ in $\mathcal{G}$ is won by the same player that wins $\mathrm{ext}(\rho)$ in $\mathcal{G}'$.

Lemma 7. *Let $\mathcal{G} \leq_{\mathcal{M}} \mathcal{G}'$. Then, $r_{\mathcal{G}}(v) = r_{\mathcal{G}'}(v, \mathrm{Init}(v))$ for all vertices v of $\mathcal{G}$.*

Proof. We show that Player 0 has an α-resilient strategy σ' for $\mathcal{G}'$ from $(v, \mathrm{Init}(v))$ if and only if she has an α-resilient strategy σ for $\mathcal{G}$ from v, which implies our claim. The translation of the strategies is the same as in the disturbance-free setting (see, e.g., [21]), but here we have to argue about resilience instead of just winning.

"$\Leftarrow$": Given a strategy σ for $\mathcal{G}$, we define σ' for $\mathcal{G}'$ via

$$\sigma'((v_0, m_0)\cdots(v_j, m_j)) = \sigma(v_0 \cdots v_j) \enspace .$$

Consider a play $\rho' = ((v_0, m_0), b_0)((v_1, m_1), b_1)((v_2, m_2), b_2)\cdots$ consistent with σ'. If $m_0 = \mathrm{Init}(v_0)$, then $\rho' = \mathrm{ext}(\rho)$ for $\rho = (v_0, b_0)(v_1, b_1)(v_2, b_2)\cdots$, which is consistent with σ. Hence, ρ' and ρ have the same winner and the same number of disturbances. Hence, if σ is α-resilient from a vertex v, then σ' is α-resilient from $(v, \mathrm{Init}(v))$.

"$\Rightarrow$": Given a strategy σ' for $\mathcal{G}'$, we define σ for $\mathcal{G}$ via $\sigma(v_0 \cdots v_j) = v$, if $\sigma'((v_0, m_0)\cdots(v_j, m_j)) = (v, m)$ for some $m \in M$, where $m_{j'} = \mathrm{Upd}^+(v_0 \cdots v_{j'})$.

A straightforward induction shows that a play in $\mathcal{G}$ is consistent with σ if and only if its extended play in $\mathcal{G}'$ is consistent with σ'. Thus, these plays have the same winner and the same number of disturbances. Thus, again, if σ' is α-resilient from a vertex $(v, \mathrm{Init}(v))$ then σ is α-resilient from v. □

As usual for game reductions, we obtain a finite-state strategy for $\mathcal{G}$ when starting with a positional strategy in $\mathcal{G}'$. To this end, consider the proof of the second implication above. If σ is positional, then the strategy σ' is implemented by $\mathcal{M}$ and the next-move function Nxt given by $\mathrm{Nxt}(v, m) = v'$, if $\sigma(v, m) = (v', m')$ for some $m' \in M$.

A similar construction works in case σ' is finite-state, say implemented by $\mathcal{M}'$. Then, σ is implemented by the product of $\mathcal{M}$ and $\mathcal{M}'$, which is defined as expected (we refer to, e.g., [21] for a formal definition). Altogether, we obtain the following result.

Corollary 4. *Let $\mathcal{G} \leq_{\mathcal{M}} \mathcal{G}'$.*

1. *If Player 0 has an α-resilient positional strategy from $(v, \mathrm{Init}(v))$ in $\mathcal{G}'$, then she has an α-resilient finite-state strategy from v in $\mathcal{G}$, which is implemented by $\mathcal{M}$.*
2. *If Player 0 has an α-resilient finite-state strategy from $(v, \mathrm{Init}(v))$ in $\mathcal{G}'$, say implemented by $\mathcal{M}'$, then she has an α-resilient finite-state strategy from v in $\mathcal{G}$, which is implemented by the product of $\mathcal{M}$ and $\mathcal{M}'$.*

Now, we can formulate the main theorem of this subsection, which shows that prefix-dependence is not a restriction, as long as the game is reducible to a prefix-independent one. Note that this is in particular true for every ω-regular winning condition (see, e.g., [18]): every such condition is recognized by a deterministic parity automaton, which can be turned into a memory structure which allows to reduce the original game to a parity game.

Theorem 3. *Let $\mathcal{G} \leq_{\mathcal{M}} \mathcal{G}'$ so that $\mathcal{G}'$ has a prefix-independent winning condition, can be effectively computed from $\mathcal{G}$, and satisfies the assumptions from Section 3.3 (with finite-state strategies).*

Then, the resilience of $\mathcal{G}$'s vertices and a finite-state optimally resilient strategy can be effectively computed.

Proof. This is a direct consequence of Lemma 7 and Theorem 1. To obtain an optimally resilient strategy, we apply Corollary 4 for finite-state strategies. □

Recall the family of games shown in Figure 3 in which Player 0 aims to prevent more than k visits to the vertex v_1 for some parameter $k \in \omega$. Such a game can be reduced to a parity game using a memory structure implementing a counter up to $k+1$. Such a memory structure has $k+1$ memory states, and a straightforward pumping argument shows that there is no smaller memory structure.

Thus, we obtain an optimally resilient strategy for Player 0 that is implemented by a memory structure with $k+1$ states. While this strategy is indeed optimally resilient, it is not of minimal size: in fact, the unique strategy for Player 0 in $\mathcal{G}_k$ is positional and optimally resilient. Thus, the approach of computing optimally resilient strategies for games with prefix-dependent winning conditions via reductions to prefix-independent winning conditions is not optimal in that sense, as it may yield unnecessarily large optimally resilient strategies. In current research, we study how to synthesize minimal optimally resilient strategies for games with prefix-dependent winning conditions.

Moreover, in the case of prefix-dependent winning conditions, the question arises whether or not optimally resilient strategies may be necessarily larger than winning ones. It is easy to construct a game in which Player 0 has a positional winning strategy, but an optimally resilient one requires an infinite amount of memory. One example is a game with a dedicated vertex v with a self-loop, such that using the self-loop ad infinitum is winning for Player 0. Furthermore, there is a disturbance edge leading from v into a disturbance-free subgame in which Player 0 needs an infinite amount of memory to win.

However, this example is not very useful, as Player 0 needs infinite memory to win the game from some vertex of her winning region. A more interesting question for further research is whether a result similar to Theorem 1 holds true for prefix-dependent games with positional winning strategies, e.g., weak parity games [9] or bounded parity games [11]. However, for both of these conditions, monotonicity arguments allow to transform finite-state optimally resilient strategies into positional ones (similar to the construction in [16, Section 5]). However, these arguments rely on monotonicity properties of the parity condition and are therefore unlikely to be generalizable. On the other hand, we are not aware of an example of a class of winning conditions that always allow for positional winning strategies for Player 0, but require memory to implement optimally resilient strategies. In future work, we investigate whether the blowup introduced by the reduction can be avoided.

5 Outlook

We have developed a fine-grained view on the quality of strategies: instead of evaluating whether or not a strategy is winning, we compute its resilience against intermittent disturbances. While this measure of quality allows constructing "better" strategies than the distinction between winning and losing strategies, there remain aspects of optimality that are not captured in our notion of resilience. In this section we discuss these aspects and give examples of games in which there are crucial differences between optimally resilient strategies. In further research, we aim to synthesize optimal strategies with respect to these criteria.

As a first example, consider the parity game shown in Figure 5. Vertices v_0 and v_3 have resilience 1 and $\omega+1$, respectively, while vertices v_1, v_2, and v_2' have resilience 0. Player 0's only choice consists of moving to v_2 or to v_2' from v_1. Let σ and σ' be strategies for Player 0 that always move to v_2 and v_2' from v_1, respectively. Both strategies are optimally resilient. Hence, the algorithm from Section 3 may yield either one, depending on the underlying parity game solver used. Intuitively, however, σ' is preferable for Player 0, as a play prefix ending in v_2' may proceed to her winning region if a single disturbance occurs. All plays encountering v_2 at some point, however, are losing for her. Hence, another interesting avenue for further research is to study **how to recover from losing**, i.e., how to construct strategies that leverage disturbances in order to leave Player 1's winning region. For safety games, this has been addressed by Dallal, Neider, and Tabuada [12].

The previous example shows that Player 0 can still make "meaningful" choices even if the play has moved outside her winning region. The game $\mathcal{G}$ shown in Figure 6 demonstrates that she can do so as well when remaining in vertices of resilience ω. Every vertex in $\mathcal{G}$ has resilience ω, since every play with finitely many disturbances eventually remains in vertices of color 0. Moreover, the only choice to be made by Player 0 is whether to move to vertex v_1 or to vertex v_1' from vertex v_0. Let σ and σ' be positional strategies that implement the former and the latter choice, respectively.

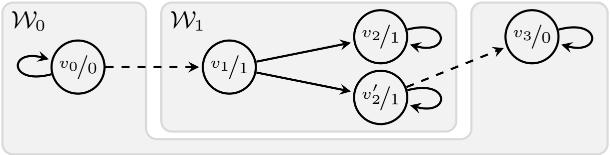


Fig. 5. Intuitively, moving from v_1 to v_2' is preferable for Player 0, as it allows her to possibly "recover" from a first disturbance with the "help" of a second one.

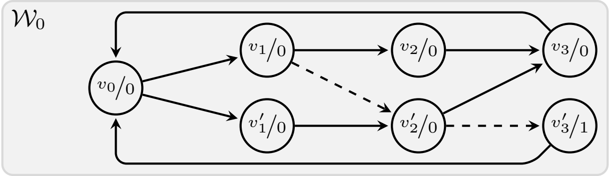


Fig. 6. Moving from v_0 to v_1 allows Player 0 to minimize visits to odd colors, while moving to v_1' allows her to minimize the occurrence of disturbances.

First consider a scenario in which visiting an odd color models the occurrence of some undesirable event, e.g., that a request has not been answered. In this case, Player 0 should aim to prevent visits to v_3' in $\mathcal{G}$, the only vertex of odd color. Hence, the strategy σ should be more desirable for her, as it requires two disturbances in direct succession in order to visit v_3'. When playing consistently with σ', however, a single disturbance suffices to visit v_3'.

On the other hand, consider a setting in which Player 0's goal is to avoid the occurrence of disturbances. In that case, σ' is preferable over σ, as it allows for fewer situations in which disturbances may occur, since no disturbances are possible from vertices v_2 and v_3.

Note that the goals of minimizing visits to vertices of odd color and minimizing the occurrence of disturbances are not contradictory: if both events are undesirable, it may be optimal for Player 0 to combine the strategies σ and σ'. In general, it is interesting to study how to **how to best brace for a finite number of disturbances**.

Recall that, due to Theorem 2, optimally resilient strategies for parity games do not require memory. In contrast, the game shown in Figure 7 demonstrates that additional memory can serve to further improve such strategies. Any strategy for Player 0 that does not stay in v_1 from some point onwards is optimally resilient. However, every visit to v_2 risks a disturbance occurring, which would lead the play into a losing sink for Player 0. Hence, it is in her best interest to remain in vertex v_1 for as often as possible, thus minimizing the possibility for disturbances to occur. This behavior does, however, require memory to implement, as Player 0 needs to count the visits to v_1 in order to not remain in that state ad infinitum. Even worse, for each optimally resilient strategy σ with finite memory there exists another optimally resilient strategy that uses more memory, but visits v_2 more rarely than σ, reducing the possibilities for disturbances to occur. Hence, it is interesting to study **how to balance avoiding disturbances with satisfying the winning condition.** This is particularly interesting if there is some cost assigned to disturbances.

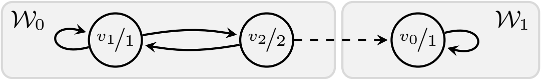


Fig. 7. Additional memory allows Player 0 to remain in v_1 longer and longer, thus decreasing the potential for disturbances.

In quantitative games, there is a further tradeoff between resilience and the semantic quality of strategies. As a simple example, consider the parity game in Figure 8 and assume, for the sake of argument, that Player 0 aims to maximize the maximal color seen infinitely often. Thus, when it comes to semantic quality of strategies, Player 0 prefers moving to v_2 over moving to v_0, when starting in v_1.

However, v_2 has resilience one while v_0 has resilience two. Hence, in this aspect, Player 0 prefers moving to v_0 over moving to v_2. In general, it is an interesting question to **determine the tradeoff between resilience and semantic quality and to compute strategies that optimize both aspects, if possible.**

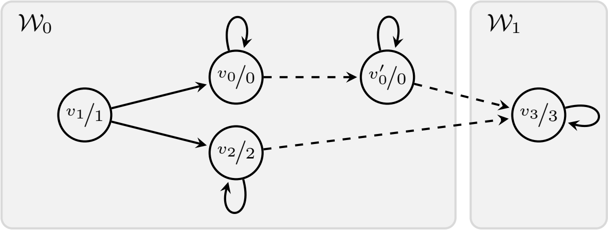


Fig. 8. A tradeoff between resilience and semantic quality (measured in the maximal color occurring infinitely often).

Finally, another important and interesting aspect, which falls outside the scope of this paper, is to provide general guidelines and best practices on how to model synthesis problems by games with disturbances. We will address these problems in future research.

6 Related Work

The notion of *unmodeled intermittent disturbances* in infinite games has recently been formulated by Dallal, Neider, and Tabuada [12]. In that work, the authors also present an algorithm for computing optimally resilient strategies for safety games with disturbances, which is an extension of the classical attractor computation [18]. Due to the relatively simple nature of such games, however, this algorithm cannot easily be extended to handle more expressive winning conditions, and the approach presented in this work relies on fundamentally different ideas.

Resilience is not a novel concept in the context of reactive systems synthesis. It appears, for instance, in the work by Topcu et al. [28] as well as Ehlers and Topcu [14]. A notion of resilience that is very similar to the one considered here has been proposed by Huang et al. [19], where the game graph is augmented with so-called "error edges". However, this setting differs from the one studied in this work in various aspects. Firstly, Huang et al. work in the framework of concurrent games and model errors as being under the control of Player 1. This contrasts to the setting considered here, in which the players play in alternation and disturbances are seen as rare events rather than antagonistic to Player 0. Secondly, Huang et al. restrict themselves to safety games, whereas we consider a much broader class of infinite games. Finally, Huang et al. compute resilient strategies with respect to a fixed parameter k, thus requiring to repeat the computation for various values of k to find optimally resilient strategies. In contrast, our approach computes an optimal strategy in a single run. Hence, they consider a more general model of interaction, but only a simple winning condition, while the notion of disturbances considered here is incomparable to theirs.

Related to resilience are various notions of *fault tolerance* [1,7,13,17] and *robustness* [3,4,5,6,23,26,27]. For instance, Brihaye et al. [7] consider quantitative games under failures, which are a generalization of sabotage games [29]. The main difference to our setting is that Brihaye et al. consider failures—embodied by a saboteur player—as antagonistic, whereas we consider disturbances as non-antagonistic events. Moreover, solving a parity game while maintaining a cost associated with the sabotage semantics below a given threshold is ExpTime-complete, whereas our approach computes optimally resilient controllers for parity conditions in quasipolynomial time.

Besides fault tolerance, robustness in the area of reactive controller synthesis has also attracted considerable interest in the recent years, typically in settings with specifications of the form $\varphi \Rightarrow \psi$ stating that the controller needs to fulfill the guarantee ψ if the environment satisfies the assumption φ. A prominent example of such work is that of Bloem et al. [3], in which the authors understand robustness as the property that "if assumptions are violated temporarily, the system is required to recover to normal operation with as few errors as possible" and consider the synthesis of robust controllers for the GR(1) fragment of Linear Temporal Logic [6]. Other examples include quantitative synthesis [4], where robustness is defined in terms of payoffs, and the synthesis of robust controllers for cyber-physical systems [23,26]. For a more in-depth discussion of related notions of resilience and robustness in reactive synthesis, we refer the interested reader to Dallal, Neider, and Tabuada's section on related work [12, Section I]. Moreover, a survey of a large body of work dealing with robustness in reactive synthesis has been presented by Bloem et al. [5].

Finally, note that for the special case of parity games, we can also characterize vertices of finite resilience (cf. Subsection 3.1) by a reduction to finding optimal strategies in energy parity games [10], which yields the same complexity as our algorithm (though such a reduction would not distinguish between vertices with resilience ω and vertices with resilience $\omega + 1$. Also, it is unclear if and how this reduction can be extended to other winning conditions and if custom-made solutions would be required for each new class of game. By contrast, our refinement-based approach works for any class of infinite games that satisfies the mild assumptions discussed in Section 4.

7 Conclusion

We presented an algorithm for computing optimally resilient strategies in games with disturbances that is applicable to any game that satisfies some mild (and necessary) assumptions. Thereby, we have vastly generalized the work of Dallal, Neider, and Tabuada, who only considered safety games. Furthermore, we showed that optimally resilient strategies are typically of the same size as classical winning strategies. Finally, we have illustrated numerous novel phenomena that appear in the setting with disturbances but not in the classical one. Studying these phenomena is a very promising direction of future work.

As part of future work, we are currently implementing our proposed method on top of the parity game solver Oink [30] and SCOTS [25], a tool for the synthesis of controllers in the context of dynamic and cyber-physical systems. Besides developing an end-to-end synthesis tool for controllers of dynamic and cyber-physical systems, a major part of this effort is to evaluate the impact of the polynomial overhead as compared to classical parity game solvers. Preliminary experiments with this prototype implementation suggest that the additional overhead does not impact the overall performance much.

References

1. Paul C. Attie, Anish Arora, and E. Allen Emerson. Synthesis of fault-tolerant concurrent programs. *ACM Trans. Program. Lang. Syst.*, 26(1):125–185, 2004.
2. Julien Bernet, David Janin, and Igor Walukiewicz. Permissive strategies: from parity games to safety games. *ITA*, 36(3):261–275, 2002.
3. Roderick Bloem, Krishnendu Chatterjee, Karin Greimel, Thomas A. Henzinger, Georg Hofferek, Barbara Jobstmann, Bettina Könighofer, and Robert Könighofer. Synthesizing robust systems. *Acta Inf.*, 51(3-4):193–220, 2014.
4. Roderick Bloem, Krishnendu Chatterjee, Thomas A. Henzinger, and Barbara Jobstmann. Better quality in synthesis through quantitative objectives. In Ahmed Bouajjani and Oded Maler, editors, *CAV 2009*, volume 5643 of *LNCS*, pages 140–156. Springer, 2009.
5. Roderick Bloem, Rüdiger Ehlers, Swen Jacobs, and Robert Könighofer. How to handle assumptions in synthesis. In Krishnendu Chatterjee, Rüdiger Ehlers, and Susmit Jha, editors, *SYNT 2014*, volume 157 of *EPTCS*, pages 34–50, 2014.
6. Roderick Bloem, Barbara Jobstmann, Nir Piterman, Amir Pnueli, and Yaniv Sa'ar. Synthesis of Reactive(1) designs. *J. Comput. Syst. Sci.*, 78(3):911–938, 2012.
7. Thomas Brihaye, Gilles Geeraerts, Axel Haddad, Benjamin Monmege, Guillermo A. Pérez, and Gabriel Renault. Quantitative games under failures. In *FSTTCS 2015*, volume 45 of *LIPIcs*, pages 293–306. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015.
8. Cristian S. Calude, Sanjay Jain, Bakhadyr Khoussainov, Wei Li, and Frank Stephan. Deciding parity games in quasipolynomial time. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *STOC 2017*, pages 252–263. ACM, 2017.
9. Krishnendu Chatterjee. Linear time algorithm for weak parity games. *arXiv*, 0805.1391, 2008.
10. Krishnendu Chatterjee and Laurent Doyen. Energy parity games. *Theor. Comput. Sci.*, 458:49–60, 2012.
11. Krishnendu Chatterjee, Thomas A. Henzinger, and Florian Horn. Finitary winning in ω-regular games. *ACM Trans. Comput. Log.*, 11(1), 2009.

12. Eric Dallal, Daniel Neider, and Paulo Tabuada. Synthesis of safety controllers robust to unmodeled intermittent disturbances. In *CDC 2016*, pages 7425–7430. IEEE, 2016.
13. Ali Ebnenasir, Sandeep S. Kulkarni, and Anish Arora. FTSyn: a framework for automatic synthesis of fault-tolerance. *STTT*, 10(5):455–471, 2008.
14. Rüdiger Ehlers and Ufuk Topcu. Resilience to intermittent assumption violations in reactive synthesis. In Martin Fränzle and John Lygeros, editors, *HSCC 2014*, pages 203–212. ACM, 2014.
15. John Fearnley, Sanjay Jain, Sven Schewe, Frank Stephan, and Dominik Wojtczak. An ordered approach to solving parity games in quasi polynomial time and quasi linear space. In Hakan Erdogmus and Klaus Havelund, editors, *SPIN 2017*, pages 112–121. ACM, 2017.
16. Nathanaël Fijalkow and Martin Zimmermann. Parity and streett games with costs. *LMCS*, 10(2), 2014.
17. Alain Girault and Éric Rutten. Automating the addition of fault tolerance with discrete controller synthesis. *Form. Meth. in Sys. Des.*, 35(2):190–225, 2009.
18. Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *LNCS*. Springer, 2002.
19. Chung-Hao Huang, Doron A. Peled, Sven Schewe, and Farn Wang. A game-theoretic foundation for the maximum software resilience against dense errors. *IEEE Trans. Software Eng.*, 42(7):605–622, 2016.
20. Marcin Jurdzinski and Ranko Lazic. Succinct progress measures for solving parity games. In *LICS 2017*, pages 1–9. IEEE Computer Society, 2017.
21. Lukasz Kaiser. *Logic and Games on Automatic Structures - Playing with Quantifiers and Decompositions*, volume 6810 of *LNCS*. Springer, 2011.
22. Karoliina Lehtinen. A modal μ perspective on solving parity games in quasi-polynomial time. In Anuj Dawar and Erich Grädel, editors, *LICS 2018*, pages 639–648. ACM, 2018.
23. Rupak Majumdar, Elaine Render, and Paulo Tabuada. A theory of robust omega-regular software synthesis. *ACM Trans. Embedded Comput. Syst.*, 13(3):48:1–48:27, 2013.
24. Donald A. Martin. Borel determinacy. *Annals of Mathematics*, 102:363–371, 1975.
25. Matthias Rungger and Majid Zamani. SCOTS: A tool for the synthesis of symbolic controllers. In *HSCC 2016*, pages 99–104. ACM, 2016.
26. Paulo Tabuada, Sina Yamac Caliskan, Matthias Rungger, and Rupak Majumdar. Towards robustness for cyber-physical systems. *IEEE Trans. Automat. Contr.*, 59(12):3151–3163, 2014.
27. Paulo Tabuada and Daniel Neider. Robust linear temporal logic. In Jean-Marc Talbot and Laurent Regnier, editors, *CSL 2016*, volume 62 of *LIPIcs*, pages 10:1–10:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.
28. Ufuk Topcu, Necmiye Ozay, Jun Liu, and Richard M. Murray. On synthesizing robust discrete controllers under modeling uncertainty. In Thao Dang and Ian M. Mitchell, editors, *HSCC 2012*, pages 85–94. ACM, 2012.
29. Johan van Benthem. An essay on sabotage and obstruction. In *Mechanizing Mathematical Reasoning, Essays in Honor of Jörg H. Siekmann on the Occasion of His 60th Birthday*, volume 2605 of *LNCS*, pages 268–276. Springer, 2005.
30. Tom van Dijk. Oink: An implementation and evaluation of modern parity game solvers. In *TACAS 2018*, volume 10805 of *LNCS*, pages 291–308. Springer, 2018.