

Galois Energy Games

To Solve All Kinds of Quantitative Reachability Problems

Caroline Lemke ✉ 

Department of Computing Science, Carl von Ossietzky Universität Oldenburg, Germany

Benjamin Bisping ✉ 

Institute of Software Engineering and Theoretical Computer Science, Technische Universität Berlin, Germany

Abstract

We provide a generic decision procedure for energy games with energy-bounded attacker and reachability objective, moving beyond vector-valued energies and vector-addition updates. All we demand is that energies form well-founded bounded join-semilattices, and that energy updates have an upward-closed domain and can be “undone” through a Galois-connected function. We instantiate these *Galois energy games* to common energy games, declining energy games, multi-weighted reachability games, coverability on vector addition systems with states, and shortest path problems, supported by an Isabelle-formalization and two implementations. For the instantiations, our simple algorithm is polynomial w.r.t. game graph size and exponential w.r.t. dimension.

2012 ACM Subject Classification Theory of computation

Keywords and phrases Energy games, Galois connection, Reachability, Game theory, Decidability

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2025.29

Related Version *Previous Version:* <https://arxiv.org/abs/2505.14691> [21]

AFP Entry: https://www.isa-afp.org/entries/Galois_Energy_Games.html [20]

Supplementary Material *Collection:* <https://github.com/crmrtz/galois-energy-games> [19]

Acknowledgements We want to thank Gabriel Vogel for the WebGPU prototype implementation and Uli Fahrenberg, Lukas Panneke and the anonymous reviewers for their valuable feedback.

1 Introduction

Most problems of how to reach a goal at minimal cost can be expressed as *energy games* [10, 9]. In energy games, players try to achieve objectives, but fail if they run out of resources that they can gain or lose on the way. But strangely, existing literature does not quite explain how to solve the following kind of multi-weighted energy game:

► **Example 1.** The energy game of Figure 1a models the situation of a researcher, who hosts a coauthor at his office and wants to energize her with ten shots of espresso. For this, he leaves his Office for the CoffeeMaker and brews some Shots (bounded by the number of Cups he carries). Upon returning to the Office, the shots can be transformed into energization E. The return usually takes two units of time T. But there is a quicker way, passing by the office of the DepartmentHead. Unfortunately, the department head will either cost our host some time by chatting or help herself to one of the filled coffee cups. (Which of the two happens is not under his control.) Depending on the number of cups, he has to take several rounds until energization E reaches level 10 and the game can be won by moving to Energized.

Assuming that we start with no energization and shots ($E = \text{Shots} = 0$), but with a budget of cups and time, how much of the two does the host need to reach his objective? There are several answers, summarized by the Pareto front of cups and time $\{(1, 20), (2, 10), (3, 6), (4, 4), (5, 2), (10, 1)\}$, which is drawn in Figure 1b.



© Caroline Lemke and Benjamin Bisping;

licensed under Creative Commons License CC-BY 4.0

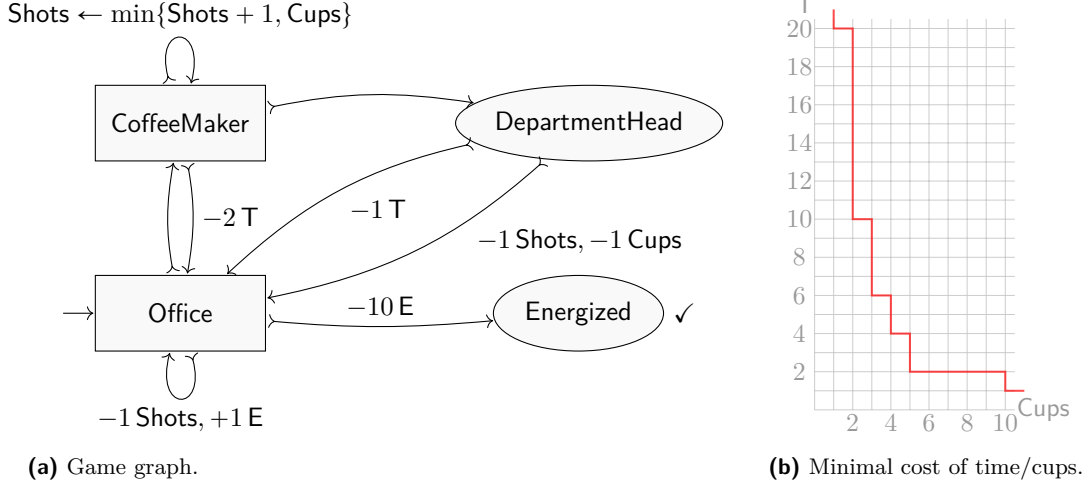
36th International Conference on Concurrency Theory (CONCUR 2025).

Editors: Patricia Bouyer and Jaco van de Pol; Article No. 29; pp. 29:1–29:19

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 1** Espresso energy game of Example 1.

The strategy of the host depends on the resources: If there is only one cup, then he should never pass by the department head, where it might be lost. So, he will cycle through $\text{Office} \rightarrow \text{CoffeeMaker} \xrightarrow{\text{Shots} \leftarrow \min\{\text{Shots} + 1, \text{Cups}\}} \text{CoffeeMaker} \xrightarrow{-2 T} \text{Office} \xrightarrow{-1 \text{ Shots}, +1 E} \text{Office} \dots$ until the espressos have been served after at least $10 \cdot 2 T$. If the host brings 10 or more cups, he can fill them all and safely take the quicker route. There might be a delay at the head's office or it might take a second round, but $1 T$ will suffice. For the scenarios in-between, things become more involved ... would it not be nice to have an algorithm to find the minimal budgets there?

Energy games are often studied in light of their closeness to mean-payoff games and (games on) vector addition systems with states [2, 9, 10]. They can model how different resources may interact with each other, for example, in a trade-off between one resource gained (energization) and another lost (espresso shots). Energy updates are usually understood as vector addition. But vector addition is too restricted to capture Example 1, where the **Cups** component bounds the **Shots** component!

A recent article by Bisping on process equivalence games [3] introduces *declining energy games*, which allow resources to constrain one-another. But these games prohibit increasing updates! Another recent article by Brihaye and Goeminne [8] formally studies decision procedures and complexity bounds for *multi-weighted reachability games*, which roughly are the intersection of declining and standard energy games, that is, even more restricted.

With this work, we show that we can have all this cake and eat it: Positive as well as negative updates (even multiplicative ones), components bounding each other, and a formalized solution. All we need is that energy updates can be undone through *Galois connections*, a weak form of inversion on monotonic functions. Other than that, we demand little of the precise objects used as energies, which makes our results quite versatile.

Contributions.

- Section 2 introduces *Galois energy games* as a generalization of existing energy game models with an *inductive characterization of winning budgets*.
- Section 3 provides a *simple fixed-point algorithm to compute winning budgets* of Galois energy games and proves its correctness – formalized in Isabelle/HOL.

- We examine how other energy game models and quantitative reachability problems can be *understood as instances of our result* in Section 4 and discuss two prototype implementations.
- We give precise *parametric upper bounds for the complexity* of the algorithm and its instantiations in Section 5: polynomial in game graph size, exponential in dimensionality.
- In Section 6, we discuss how other scenarios and winning conditions could be addressed.

2 Energy Games

Energy games are zero-sum two-player games with perfect information. They are played on directed graphs labeled by energy update functions. Energies represent resources such as time or espresso shots, often modeled as vectors. In our quest for genericity, we model energies as arbitrary partially ordered sets, where each non-empty subset has at least one minimal element and suprema of finite subsets exist, i.e. well-founded bounded join-semilattices. The energy updates describe how each move across an edge affects the current energy, altering the resources as players navigate the game.

2.1 Generic Energy Games

We introduce energy games as generically as possible. For this, we first fix a set of energies \mathcal{E} and assume (\mathcal{E}, \leq) to be a well-founded bounded join-semilattice. Further, we assume (\mathcal{E}, \leq) to be computable, more precisely we assume \leq to be a computable relation and mapping two energies to their supremum to be a computable function. Then, we can define energy games played on directed graphs labeled by updates.

- **Definition 2** (Energy game). An energy game $\mathcal{G} = (G, G_a, \succrightarrow)$ over (\mathcal{E}, \leq) consists of
- a set of positions G , partitioned into attacker positions $G_a \subseteq G$ and defender positions $G_d := G \setminus G_a$, and
 - a partial function $\succrightarrow : (G \times G) \rightarrow (\mathcal{E} \rightarrow \mathcal{E})$ mapping edges to partial energy functions.
- For $g, g' \in G$ we write $g \succrightarrow g'$ instead of $\succrightarrow(g, g') = u$ and call u an update in \mathcal{G} . The game with initial position $g_0 \in G$ and initial energy $e_0 \in \mathcal{E}$ is denoted as $\mathcal{G}[g_0, e_0]$.

► **Example 3.** Formally, Definition 2 does not permit the two edges between `DepartmentHead` and `Office` in Example 1. However, this can be fixed by adding a node `Chat` such that `DepartmentHead` $\succrightarrow^{-1\text{T}}$ `Chat` \succrightarrow `Office`. In the example, the researcher getting coffee is the attacker, choosing at rectangular positions, while the environment is the defender at elliptical positions. The resources (cups, time, shots, energization) can be represented as four-dimensional natural vectors, making Example 1 an energy game over \mathbb{N}^4 with the component-wise order.

In an energy game, the players move one token from one position to the next according to the edges, resulting in a play. At attacker positions, the attacker chooses (according to some attacker strategy) and vice versa.

- **Definition 4** (Play, strategy). Let $\mathcal{G} = (G, G_a, \succrightarrow)$ be an energy game over (\mathcal{E}, \leq) . A play in \mathcal{G} is a possibly infinite walk $\rho = g_0 g_1 \dots \in G^* \cup G^\omega$ in the underlying directed graph.

Let $G_p \in \{G_a, G_d\}$. A strategy $s : G^* G_p \rightarrow G$ is a partial function mapping finite plays to a next position such that $s(g_0 \dots g_n)$ is some successor of g_n if $g_n \in G_p$ is not a deadlock. If $G_p = G_a$ (resp. $G_p = G_d$) we call s an attacker (resp. defender) strategy. A play $\rho = g_0 g_1 \dots$ is consistent with s if $g_i \in G_p$ implies $g_{i+1} = s(g_0 \dots g_i)$ for each $i \in \mathbb{N}$ with $i + 1 < |\rho|$.

During a play, an initial energy is repeatedly updated according to the edges in the play. This is defined as the energy level, which keeps track of the resources. The attacker wins a play if and only if it ends in a deadlocked defender position before running out of energy, i.e. without the energy level becoming undefined.

► **Definition 5** (Energy level, winning a play). *Let $\mathcal{G} = (G, G_a, \succrightarrow)$ be an energy game over (\mathcal{E}, \leq) . The energy level of a play $\rho = g_0 g_1 \dots$ w.r.t. initial energy $e_0 \in \mathcal{E}$ is inductively defined as $EL(\rho, e_0, 0) := e_0$ and $EL(\rho, e_0, i + 1) := u(EL(\rho, e_0, i))$ where $g_i \succrightarrow g_{i+1}$ and $i + 1 < |\rho|$.*

If the energy level becomes undefined or the play is infinite, the defender wins the play. If the final energy level $EL(\rho, e_0, |\rho| - 1)$ of a finite play is defined and its last position g_l is a deadlock, then the defender wins the play if $g_l \in G_d$ and the attacker wins if $g_l \in G_a$.

In other settings, infinite plays (where the energy level is always defined) are won by the energy-bound player, i.e. our attacker [10, 15]. We will revisit this distinction in Section 6 as safety winning conditions. Now we define what it means to win a game w.r.t. a starting position and an initial energy.

► **Definition 6** (Winning strategy, winning a game). *Let $\mathcal{G} = (G, G_a, \succrightarrow)$ be an energy game over (\mathcal{E}, \leq) . For $g_0 \in G$ and $e_0 \in \mathcal{E}$, an attacker winning strategy for $\mathcal{G}[g_0, e_0]$ is an attacker strategy ensuring the attacker to eventually win all consistent plays.*

The attacker wins the game $\mathcal{G}[g_0, e_0]$ if and only if such an attacker winning strategy exists. The attacker winning budget of a position $g \in G$ is the set of energies sufficient for the attacker to enforce a win, i.e. $\text{Win}_a(g) = \{e \in \mathcal{E} \mid \exists s. s \text{ is an attacker winning strategy for } \mathcal{G}[g, e]\}$.

Defender winning strategies and budgets can be defined symmetrically to Definition 6. However, it is sufficient to focus on the attacker's objective, since energy games are determined w.r.t. a starting position and initial energy.

► **Lemma 7.** *Let \mathcal{G} be an energy game over (\mathcal{E}, \leq) . For all positions $g \in G$ and energies $e \in \mathcal{E}$ either the defender or the attacker wins $\mathcal{G}[g, e]$. Further, there exists an energy-positional winning strategy, i.e. a winning strategy that depends only on the current position and energy level.*

Lemma 7 follows from the construction of a parity game, which is positionally determined. For more details we refer to the appendix.

► **Remark 8.** The winning conditions in Definition 5 and 6 encode quantitative reachability problems: In reachability games [9], the attacker aims to reach a fixed target set in a (weighted) game graph. By treating defender deadlock positions as the target set, we formulate (quantitative) reachability winning conditions.

When starting with a multi-weighted reachability game [8], i.e. a game $(G, G_a, \succrightarrow)$ weighted by vectors of naturals with a target set $F \subseteq G$, this can easily be transformed to an energy game: Remove outgoing edges of positions in $G_d \cap F$ and add a loop to every deadlock in $G_d \setminus F$. Further, add a deadlocked defender position with an ingoing edge labeled by the zero-vector from each position in $G_a \cap F$. By this construction, a defender position is a deadlock if and only if it corresponds to a position in the target set F in the original game. In the reachability game, one is interested in the minimal ensured cost for a position g , i.e. a minimal upper bound that the attacker can enforce while winning for the sum of edge weights of plays starting in g . By switching from addition to subtraction, this corresponds to asking for the minimal attacker winning budget of g in the constructed energy game.

2.2 Inductive Characterization of Attacker Winning Budgets

We want to decide energy games by calculating the attacker winning budgets. For this, we study attacker winning budgets and observe that they are upward-closed in monotonic energy games, i.e. where all updates are monotonic and have an upward-closed domain.

► **Definition 9** (Monotonic energy game). *Let \mathcal{G} be an energy game over (\mathcal{E}, \leq) . Then, \mathcal{G} is a monotonic energy game if, for all updates u in \mathcal{G} and all $e, e' \in \mathcal{E}$ with $e \leq e'$, the following implication holds: If $u(e)$ is defined, then so is $u(e')$ and $u(e) \leq u(e')$.*

► **Lemma 10.** *Let \mathcal{G} be a monotonic energy game over (\mathcal{E}, \leq) . Then, the attacker winning budgets are upward-closed, i.e. $\text{Win}_a(g) = \uparrow \text{Win}_a(g) := \{e \in \mathcal{E} \mid \exists e' \in \text{Win}_a(g). e' \leq e\}$ for all positions $g \in G$.*

Lemma 10 follows directly from the definition of winning budgets and monotonic games since the same strategy serves as a witness when starting with more energy.

In monotonic energy games, each attacker winning budget of a position g can be characterized by a finite Pareto front, i.e. by the minimal attacker winning budget $\text{Win}_a^{\min}(g) := \text{Min Win}_a(g)$. (For a set M we distinguish between the set of all minimal elements, i.e. $\text{Min } M := \{m \in M \mid \forall m' \in M. m' \not\leq m\}$, and the minimum. We write $\min M$ to denote the minimum if it exists.) Since the energies are well-founded, antichains in (\mathcal{E}, \leq) are finite and thereby so are the sets $\text{Win}_a^{\min}(g)$ for all positions g .

Attacker winning budgets have more structure. In particular, they can be characterized inductively.

► **Lemma 11.** *Let \mathcal{G} be an energy game over (\mathcal{E}, \leq) . The attacker winning budgets can be characterized inductively by the rules:*

$$\frac{g \in G_a \quad g \xrightarrow{u} g' \quad u(e) \in \text{Win}_a(g')}{e \in \text{Win}_a(g)} \quad \frac{g \in G_d \quad \forall g'. g \xrightarrow{u} g' \longrightarrow u(e) \in \text{Win}_a(g')}{e \in \text{Win}_a(g)}$$

Proof. An attacker winning strategy can be constructed from attacker winning strategies at successors w.r.t. the accordingly updated energies. This implies soundness.

To argue completeness we utilize Lemma 7. Assuming $e \in \text{Win}_a(g)$, there exists an energy-positional attacker winning strategy s for $\mathcal{G}[g, e]$. We then fix an order induced by s by setting $(g'', e'') \leq_s (g', e')$ if $(g'', e'') = (g', e')$ or there is an edge in the configuration graph from (g', e') to (g'', e'') consistent with s , i.e. if $g' \xrightarrow{u} g''$, $e'' = u(e')$ and $g' \in G_a \longrightarrow s(g', e') = g''$. Then, a derivation of $e \in \text{Win}_a(g)$ can be found by well-founded induction. ◀

2.3 Galois Energy Games

In order to convert the inductive characterization of Lemma 11 into a fixed-point algorithm, it seems that we need to *invert* energy updates in the game. However, it turns out that what is sufficient is a weaker form of invertibility given by *Galois connections*.

► **Definition 12** (Galois energy game). *A Galois energy game is a monotonic energy game \mathcal{G} over (\mathcal{E}, \leq) such that $u^{\mathcal{G}}(e') := \min\{e \in \mathcal{E} \mid e' \leq u(e)\}$ exists and is computable for all updates u in \mathcal{G} and $e' \in \mathcal{E}$.*

Galois connections can be characterized by one of the following four equivalent properties. We refer to Ern  et al. [14] for more detail.

► **Lemma 13.** Let $\mathcal{P} = (P, \leq_P)$ and $\mathcal{Q} = (Q, \leq_Q)$ be partially ordered sets with functions $f : P \rightarrow Q$ and $g : Q \rightarrow P$. Then, the following are equivalent:

1. $f(p) \leq_Q q \iff p \leq_P g(q)$ holds for all $p \in P$ and $q \in Q$.
2. f and g are monotonic, $g \circ f$ is increasing, and $f \circ g$ is decreasing.
3. f is monotonic and $g(q) = \max_P \{p \in P \mid f(p) \leq_Q q\}$ for each $q \in Q$.
4. g is monotonic and $f(p) = \min_Q \{q \in Q \mid p \leq_P g(q)\}$ for each $p \in P$.

If any of the (and thereby all) properties hold, then f and g form a Galois connection between P and Q . By Lemma 13 an energy game \mathcal{G} over (\mathcal{E}, \leq) is a Galois energy game if and only if each update u in \mathcal{G} has an upward-closed domain and there exists a computable function $u^\circ : \mathcal{E} \rightarrow \text{dom}(u)$ such that u° and u form a Galois connection between \mathcal{E} and $\text{dom}(u)$. Then, the minima used to define u° in Definition 12 always exist.

► **Example 14.** The energy game from Example 1 is a Galois energy game. In particular, $(\text{Shots} \leftarrow \min\{\text{Shots} + 1, \text{Cups}\})^\circ(e_C, e_T, e_S, e_E) = (\max\{e_C, e_S\}, e_T, \max\{e_S - 1, 0\}, e_E)$.

3 Deciding Galois Energy Games

For a fixed position g , the *known initial credit problem* asks whether energy e is sufficient for the attacker to always win when starting from g , i.e. whether $e \in \text{Win}_a(g)$. The *unknown initial credit problem* asks, whether there exists an energy sufficient for the attacker to win from g , i.e. whether $\text{Win}_a(g) \neq \emptyset$. Subsuming both these problems, we provide an algorithm calculating the *minimal attacker winning budgets of each position*.

The core idea of the algorithm is that of a shortest path algorithm: We start at deadlocks and move backwards through the graph using the inductive characterization of attacker winning budgets (Lemma 11). To calculate energies backwards, we use Galois connections, allowing us to prove:

► **Theorem 15** (Decidability of Galois energy games). *Let \mathcal{G} be a Galois energy game over (\mathcal{E}, \leq) with a finite set of positions. Then, the (un)known initial credit problem is decidable.*

3.1 The Algorithm

To prove Theorem 15, we first discuss Algorithm 1. This is a simplified and generalized version of an algorithm proposed by Bisping [3].

■ **Algorithm 1** Computing minimal attacker winning budgets of Galois energy game \mathcal{G} .

```

1 def compute_winning_budgets( $\mathcal{G} = (G, G_a, \succ)$ ):
2   win := [ $g \mapsto \{\}$  |  $g \in G$ ]
3   do
4     old_win := win
5     for  $g \in G$  :
6       if  $g \in G_a$  :
7         win[g] := Min{ $u^\circ(e') \mid g \succ g' \wedge e' \in \text{old\_win}[g']$ }
8       else:
9         win[g] := Min{sup{ $u^\circ(e_{g'}) \mid g \succ g' \mid \forall g'. g \succ g' \longrightarrow e_{g'} \in \text{old\_win}[g']$ }
10   while win  $\neq$  old_win
11   return old_win

```

Starting by assigning the empty set to each position in line 2, we then apply lines 3 to 10 adding sufficient energies to winning budgets and taking the minimal elements (which exist by well-foundedness) until we reach a fixed point to return.

For a play to be won by the attacker, it has to end in a defender deadlock position. At such a defender position g_d all energies are in the attacker winning budget and $\text{Win}_a^{\min}(g_d) = \text{Min } \mathcal{E}$. If a defender position has successors, then an energy is in the attacker winning budgets if it is sufficient to win moving to any successor the defender might choose. In Galois energy games, that is $e \in \text{Win}_a(g_d)$ for $g_d \in G_d$ if $u^{\cup}(e_{g'}) \leq e$ for some $e_{g'} \in \text{Win}_a(g')$ for each g' with $g_d \succrightarrow g'$. This is ensured by line 9. Note that the supremum always exists because the energies form a bounded join-semilattice and the sets in question are finite.

Similarly, sufficient energies for attacker positions can be calculated from energies in the winning budget of successors, as done in line 7. In particular, $g_a \in G_a$ and $e' \in \text{Win}_a(g')$ for some g' with $g_a \succrightarrow g'$ implies $u^{\cup}(e') \in \text{Win}_a(g_a)$.

■ **Table 1** Application of Algorithm 1 to Example 1.

Iteration	0	1	2	3	...
Energized	\emptyset	$\{(0, 0, 0, 0)\}$	$\{(0, 0, 0, 0)\}$	$\{(0, 0, 0, 0)\}$...
Office	\emptyset	\emptyset	$\{(0, 0, 0, 10)\}$	$\{(0, 0, 0, 10), (0, 0, 1, 9)\}$...
CoffeeMaker	\emptyset	\emptyset	\emptyset	$\{(0, 2, 0, 10)\}$...

Table 1 illustrates how the information, which positions are winnable, travels through the graph when applying Algorithm 1 to Example 1. The vectors represent cups, time, shots and energization in that order.

3.2 Correctness

Since we are considering a fixed point algorithm, we use Kleene's fixed point theorem [25, 28] to prove correctness of our algorithm and thereby our decidability result, Theorem 15. The following version of Kleene's fixed point theorem relies on the notion of directed sets, i.e. a subset of a partially ordered set where all pairs of elements have an upper bound in the subset. In a directed-complete partial order all directed sets have a (directed) supremum. A Scott-continuous function preserves directed suprema.

► **Lemma 16** (Kleene's fixed point theorem). *Let (Q, \leq_Q) be a directed-complete partial order with least element $q \in Q$ and a Scott-continuous function $f : Q \rightarrow Q$. Then, the set of fixed points of f in Q forms a complete lattice with least fixed point $\sup_Q \{f^i(q) \mid i \in \mathbb{N}\}$, where f^i refers to the i -th iterate of f .*

Proof of Theorem 15. To utilize Kleene's fixed point theorem, we define a partial order on mappings from positions to possible Pareto fronts, i.e. antichains in \mathcal{E} . We call the set of such mappings $\text{Pareto}_G \subseteq (G \rightarrow 2^{\mathcal{E}})$ and define

$$F \leq F' \iff \forall g \in G. F(g) \subseteq \uparrow F'(g) \quad \text{and} \quad (\sup_{\leq} P)(g) := \text{Min} \bigcup_{F \in P} \uparrow F(g)$$

for $F, F' \in \text{Pareto}_G$ and $P \subseteq \text{Pareto}_G$. These definitions yield that (Pareto_G, \leq) is a directed-complete partial order with minimal element $\mathbf{0} := \lambda g. \emptyset$.

We then consider the function $\text{Iteration}_G : \text{Pareto}_G \rightarrow \text{Pareto}_G$ corresponding to one iteration of the while-loop in Algorithm 1 when applied to G and show that it is Scott-continuous w.r.t. the \leq -order, i.e. $\text{Iteration}_G(\sup_{\leq} P) = \sup_{\leq} \{\text{Iteration}_G(F) \mid F \in P\}$ for all directed sets $P \subseteq \text{Pareto}_G$.

Let $P \subseteq \text{Pareto}_G$ be such a directed set. The definition of Iteration_G and monotonicity of all u^ζ together imply monotonicity of Iteration_G and thereby $\text{Iteration}_G(F) \leq \text{Iteration}_G(\sup_{\leq} P)$ for all $F \in P$. Therefore, what remains to show is $\text{Iteration}_G(\sup_{\leq} P) \leq \sup_{\leq} \{\text{Iteration}_G(F) \mid F \in P\}$. Let $g \in G$ and $e \in \text{Iteration}_G(\sup_{\leq} P)(g)$. We now show $e \in \uparrow (\sup_{\leq} \{\text{Iteration}_G(F) \mid F \in P\})(g)$. We focus on the more intricate case and assume $g \in G_d$. By definition of Iteration_G , there exist $e_{g'} \in (\sup_{\leq} P)(g')$ for each $g' \in G$ with $g \succ_{\text{u}} g'$ such that $e = \sup\{u^\zeta(e_{g'}) \mid g \succ_{\text{u}} g'\}$. By definition of \sup_{\leq} , there exist $F_{g'} \in P$ with $e_{g'} \in \uparrow F_{g'}(g')$. Since the set of positions is finite, so is $\{F_{g'} \mid g \succ_{\text{u}} g'\} \subseteq P$ and there exists an upper bound $F' \in P$ such that $e_{g'} \in \uparrow F'(g')$ for each successor g' of g . Thus, $e \in \uparrow \text{Iteration}(F')(g) \subseteq \uparrow (\sup_{\leq} \{\text{Iteration}(F) \mid F \in P\})(g)$.

With this, we can apply Lemma 16 and conclude that a least fixed point, namely $\sup_{\leq} \{\text{Iteration}_G^i(\mathbf{0}) \mid i \in \mathbb{N}\}$, exists. We now show, that the least fixed point is Win_a^{\min} by first arguing that Win_a^{\min} is a fixed point. For this, we introduce inductively-defined sets $S(g)$ for $g \in G$ mimicking the process of the iteration of the while-loop without taking the minimum:

$$\frac{g \in G_a \quad g \succ_{\text{u}} g' \quad e' \in S(g')}{u^\zeta(e') \in S(g)} \quad \frac{g \in G_d \quad \forall g'. g \succ_{\text{u}} g' \longrightarrow e_{g'} \in S(g')}{(\sup\{u^\zeta(e_{g'}) \mid g \succ_{\text{u}} g'\}) \in S(g)}$$

Then, $S(g) \subseteq \text{Win}_a(g)$ for all $g \in G$ follows from a simple induction over the structure of S , utilizing the inductive characterization of attacker winning budgets as well as Lemma 10 and 13. Further, $\text{Win}_a(g) \subseteq \uparrow S(g)$ can be shown by a well-founded induction over the strategy-induced order introduced in the proof of Lemma 11 using Lemma 13. With Lemma 10, we can conclude $\text{Win}_a(g) = \uparrow S(g)$ and thereby $\text{Win}_a^{\min} = \text{Min} \circ S$. Further, the definition of S directly implies that $\text{Min} \circ S$ (and therefore Win_a^{\min}) is a fixed point of Iteration_G . For any fixed point F of Iteration_G , we have $S(g) \subseteq \uparrow F(g)$ for all $g \in G$ by induction over the structure of S . Thus, $\text{Min} \circ S$ (and therefore Win_a^{\min}) is the least fixed point of Iteration_G .

Finally, we argue for termination by noting that, for each $g \in G$ and $e \in \text{Win}_a^{\min}(g)$, there exists $i_g^e \in \mathbb{N}$ such that e is added in the i_g^e -th iteration at the latest, i.e. $e \in \text{Iteration}_G^{i_g^e}(\mathbf{0})(g)$. This holds true since $\text{Win}_a^{\min} = \sup_{\leq} \{\text{Iteration}_G^i(\mathbf{0}) \mid i \in \mathbb{N}\}$. Thus, a fixed point is reached after at most $\max_{g \in G} \max_{e \in \text{Win}_a^{\min}(g)} i_g^e$ iterations of the while-loop. \blacktriangleleft

The proof of Theorem 15 also proves that without any computability assumptions Algorithm 1 is still correct, i.e. the least fixed point of Iteration_G exists and is Win_a^{\min} .

3.3 Isabelle/HOL Formalization

The proof of decidability of Galois energy games (Theorem 15) is formalized in Isabelle/HOL. Isabelle [23] is a generic interactive proof assistant supporting the formalization of mathematical theories. Theorems formalized in Isabelle undergo automated verification, ensuring that edge cases are not overlooked. While Isabelle supports multiple logical frameworks, we use the instantiation Isabelle/HOL based on Higher Order Logic (HOL). The Isabelle theories are available at <https://github.com/crmrtz/galois-energy-games>.

Our Isabelle formalization starts by providing the first formalization of this kind for any type of energy game. This includes the inductive characterization of attacker winning budgets (Lemma 11) assuming energy-positional determinacy (Lemma 7). Fixing an arbitrary Galois energy game over a well-founded bounded join-semilattice, we then formalize the function Iteration_G introduced in the proof of Theorem 15. We formalize the correctness of Algorithm 1 by showing that the minimal attacker winning budgets are the least fixed point of Iteration_G (correctness) and show termination. Thus, we have formalized the decidability of such Galois

energy games. Finally, we formalize Galois energy games over vectors of (extended) naturals with the component-wise order and conclude decidability, particularly for energy games with vector addition and taking minima of components only.

■ **Table 2** Overview of the Isabelle/HOL formalization.

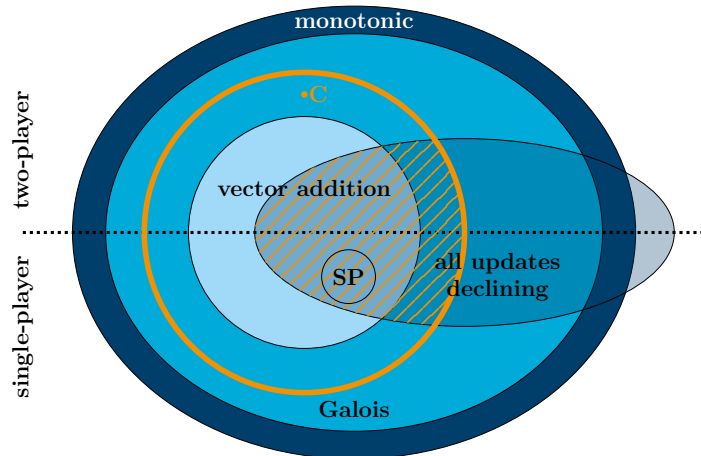
	Result	Formalization link	Proof adjustments
Lemma 7	Energy-positional determinacy	-	
Lemma 10	Upwards-closure of Win_a	<code>upward_closure_wb_nonpos</code>	
Lemma 11	Inductive characterization of Win_a	<code>inductive_winning_budget</code>	assuming Lemma 7
Lemma 13	Galois connection equivalences	<code>galois_properties</code>	only 1. \rightarrow 2.
Lemma 16	Kleene's fixed point theorem	by Yamada and Dubut [28]	
Theorem 15	Decidability	Correctness: <code>a_win_min_is_lfp</code> Termination: <code>finite_iterations</code>	assuming Lemma 7

Table 2 provides an overview of all previously discussed lemmas and theorems, along with information on which ones have been formalized and links to their formalization. For the instances discussed in Section 4 the formalization links are given in footnotes.

4 Instances of Galois Energy Games

Let us discuss how Galois energy games and the algorithm can be used to solve quantitative problems by instantiating energy orders and updates over vectors of naturals.

For the illustration, we will take an extremely classical problem and two recent ones. In Subsection 4.1, we show how shortest paths problems on graphs are encoded as single-player energy games and relate the latter to vector addition systems with states (VASS). As more recent problems, Subsection 4.2 will discuss the declining energy games of Bisping [3] and the multi-weighted reachability games of Brihaye and Goeminne [8], in a generalized version.



■ **Figure 2** Energy games over vectors of naturals with the component-wise order.

Figure 2 lays out the kinds of energy games to be discussed: The yellow circle includes all energy games with updates that mix positive, negative, and min-updates, in particular, our example energy game of the researcher getting coffee placed as yellow **C**. The crosshatched area includes the “declining energy games” discussed by Bisping [3], where the subclass with vector addition only corresponds to the games discussed by Brihaye and Goeminne [8]. The

bottom half-circle of vector addition contains energy games solving the coverability problem on VASS. **SP** in that set refers to the single-player energy games corresponding to shortest path problems for the one-dimensional case as will be discussed in Subsection 4.1.

The class of monotonic energy games is decidable too. For this, Lemke [19] generalizes Algorithm 1 by calculating sets of the form $\text{Min}\{e \in \mathcal{E} \mid e' \leq u(e)\}$ if a unique minimum does not exist, which adds another level of combinatorial explosion. For details on the boundaries of decidability for monotonic games, we refer to Abdulla et al. [1]. We focus our attention on Galois energy games, which generalize previous results in a simple yet concrete manner.

4.1 Single-Player Instances

Single-player energy games are energy games, where all defender positions are deadlocks and, therefore, only the attacker makes choices.

Shortest Paths and Distances

Shortest path and shortest distance problems [22] are classical quantitative problems. These can be seen as instances of single-player energy games, where the minimal attacker winning budgets correspond to a kind of *shortest distances* from a position to any defender position. Let us illustrate this for the one-dimensional case:

► **Proposition 17.** *Assume we have a graph (V, E) with $E \subseteq V \times \mathbb{N} \times V$, and ask for the length of shortest paths from some source node $s \in V$ to target node $t \in V$. Consider the energy game $(V \cup \{\perp\}, V, \succrightarrow)$ over (\mathbb{N}, \leq) with $v \xrightarrow{-w} v'$ if $(v, w, v') \in E$ and $t \xrightarrow{0} \perp$. Then, $e \in \text{Win}_a^{\min}(s)$ on the derived game precisely if e is the shortest distance from s to t in the original graph.*

On this instance, our Algorithm 1 boils down to a single-destination variant of the Bellman–Ford algorithm for shortest paths.

► **Remark 18.** Mixing positive and negative weights, the construction of Proposition 17 encounters differences to classical shortest-path solutions, which assume commutativity of edge costs. For example, in the graphs $s \xrightarrow{2} \circ \xrightarrow{-1} t$ and $s \xrightarrow{-1} \circ \xrightarrow{2} t$, Bellman–Ford will consider both s - t -paths to have length 1. However, the energy game interpretation leads to a different result: Moving $\xrightarrow{-2} \xrightarrow{1}$ requires an attacker budget of 2, while $\xrightarrow{1} \xrightarrow{-2}$ only needs 1. In some cases the energy interpretation is more accurate in reality: If one has an empty tank, it makes a difference whether one can refuel now or only later.

Also, conventional shortest-path algorithms struggle with negative cycles, which correspond to positive cycles in our games. In the energy interpretation, we easily determine the cost to reach a target even with an infinite charging cycle: The cost of getting to the cycle.

Vector Addition Systems with States

The single-player version of a multi-dimensional energy game over \mathbb{N}^n with the component-wise order \leq_c where edges are labeled by functions adding a vector in \mathbb{Z}^n can be used to answer the *coverability problem* [17] on VASS. In VASS, this problem asks whether a target position can be reached from a given initial configuration (a pair of a position and an energy) with at least a fixed energy remaining.

► **Proposition 19.** *Let $n \in \mathbb{N}$. Assume we have a VASS (Q, T) with $T \subseteq Q \times \mathbb{Z}^n \times Q$ with a fixed initial configuration $(s, e_s) \in Q \times \mathbb{N}^n$ and a target configuration $(t, e_t) \in Q \times \mathbb{N}^n$. Consider the energy game $(Q \cup \{\perp\}, Q, \succrightarrow)$ over (\mathbb{N}^n, \leq_c) with $v \xrightarrow{-w} v'$ if $(v, w, v') \in T$ and $t \xrightarrow{-e_t} \perp$. Then, $e_s \in \text{Win}_a(s)$ on the derived game precisely if coverability holds in the original VASS.*

The *reachability problem* [6] on VASS asks whether a configuration is reachable from another. Since the target energy has to be reached exactly, this is a stricter version of coverability. As in Proposition 19, this can be modeled as an energy game by labeling the edge $t \rightarrow \perp$ with a function to check if the current energy matches the target (that is undefined if not). However, since such a function lacks an upward-closed domain, our results are not applicable and we refer to existing literature [6, 12, 13] instead.

4.2 Energy Games over Vectors of Naturals

In recent research, energy games with reachability winning conditions have garnered some attention [3, 8]. We now demonstrate that we generalize their results. We do so with a syntactical approach to define a set of updates permitted in Galois energy games over vectors of naturals. We introduce the set \mathcal{U}_n of possible updates that allow changing entries in two ways: By adding an integer or replacing it with the minimum of components. Such updates can be represented as vectors where each entry contains the information how the respective component is updated. First, we fix a set of energies: The set of vectors of extended naturals \mathbb{N}_∞^n of dimension $n \in \mathbb{N}$ with the component-wise order \leq_c .

► **Definition 20.** Let $n \in \mathbb{N}$ and $\mathcal{U}_n := (\mathbb{Z} \cup 2^{\{0, \dots, n-1\}} \setminus \{\emptyset\})^n$. Then, we interpret $u = (u_0, \dots, u_{n-1}) \in \mathcal{U}_n$ as a partial function $u : \mathbb{N}_\infty^n \rightarrow \mathbb{N}_\infty^n$ by setting

$$u(e_0, \dots, e_{n-1}) := (e'_0, \dots, e'_{n-1}) \text{ with } e'_i := \begin{cases} e_i + z, & \text{if } u_i = z \in \mathbb{Z} \wedge -z \leq e_i \\ \min_{k \in D} e_k, & \text{if } u_i = D \subseteq \{0, \dots, n-1\} \end{cases}$$

if all e'_i are defined.

As a consequence of Theorem 15, we obtain the following.

► **Corollary 21.** Let $n \in \mathbb{N}$ and let $\mathcal{G} = (G, G_a, \rightarrow)$ be an energy game over $(\mathbb{N}_\infty^n, \leq_c)$ with updates in \mathcal{U}_n . Then, \mathcal{G} is a Galois energy game. Further, if the set of positions is finite, the game is decidable.¹

Proof. We argue that \mathcal{G} is a Galois energy game: By definition, all updates in \mathcal{U}_n are monotonic with an upward-closed domain. Generalizing the inversion in [3], we show that $u^\circ(e') = \min\{e \mid e' \leq u(e)\}$ exists for all $u \in \mathcal{U}_n$ and $e' \in \mathcal{E}$ by giving a computable calculation.

Let u be an update in \mathcal{G} , i.e. $u = (u_0, \dots, u_{n-1}) \in \mathcal{U}_n$. For $e' = (e'_0, \dots, e'_{n-1}) \in \mathbb{N}_\infty^n$ we set $u^\circ(e')$ such that the i -th component is the maximum of

- $e'_i - z$, if this is not negative and the update adds z in the i -th component,
- e'_j , if the i -th component is used when taking the minimum resulting in the j -th component,
- 0 else.

That is

$$u^\circ(e')_i := \max(\{e'_i - z \mid u_i = +z \in \mathbb{Z} \wedge z \leq e'_i\} \cup \{e'_j \mid i \in u_j \subseteq \{0, \dots, n-1\}\} \cup \{0\}). \blacktriangleleft$$

► **Remark 22.** All our arguments for \mathcal{U}_n still hold, if we allow multiplication with a nonzero natural. When calculating $u^\circ(e')_i$ we then have to add $\lceil \frac{e'_i}{m} \rceil$ to the maximum, if the update multiplies the i -th component with m .

¹ This is formalized using Remark 25 to establish a subclass relationship, i.e. a **sublocale** relationship.

► **Example 23.** Note that all occurring updates in Example 1 aside from the loop at CoffeeMaker are elements of \mathcal{U}_4 . That loop can be replaced by adding a Brew-position such that $\text{CoffeeMaker} \xrightarrow{+1 \text{ Shots}} \text{Brew} \xrightarrow{\text{Shots} \leftarrow \min\{\text{Shots}, \text{Cups}\}} \text{CoffeeMaker}$.

Actually, the adjustment of the game graph described in Example 23 is not necessary to prove the game to be a Galois energy game given that Galois connections compose.

► **Lemma 24.** Let $u_1, u_2 : \mathcal{E} \rightarrow \mathcal{E}$ be partial functions with upward-closed domains such that u_i° and u_i forms a Galois connection between \mathcal{E} and $\text{dom}(u_i)$ for $i \in \{1, 2\}$. Then, $(u_2 \circ u_1)^\circ := u_1^\circ \circ u_2^\circ$ and $u_2 \circ u_1$ form a Galois connection between \mathcal{E} and the reverse image under u_1 of $\text{dom}(u_2)$, i.e. $u_1^{-1}(\text{dom}(u_2))$.²

► **Remark 25.** Lemma 24 implies that it suffices to study simple updates as generators and check that they are permitted in a Galois energy game, to prove decidability. In particular, we may focus on updates in $\{-1, 0, +1\}^n \subseteq \mathbb{Z}^n$ that add or subtract at most one in any component to understand all updates in $\mathbb{Z}^n \subseteq \mathcal{U}_n$.

Now we can turn to instances where multi-weighted energy games with reachability conditions were considered. Bisping et al. [3, 4, 5] introduce games generalizing the bisimulation game to simultaneously decide all common notions of behavioral equivalence, i.e. the linear-time-branching-time spectrum [26]. In particular, they use energy games over $(\mathbb{N}_\infty^n, \leq_c)$ with declining updates in \mathcal{U}_n . The decidability of those games is a direct implication of Corollary 21.

The multi-weighted reachability games described by Brihaye and Goeminne [8] can easily be transformed to energy games with declining updates in \mathbb{Z}^n by utilizing the construction from Remark 8. When considering the component-wise order, decidability follows directly from Corollary 21. When considering the lexicographical order \leq_l , decidability can be concluded using Theorem 15.

4.3 Implementations

We now illustrate the practical feasibility of Algorithm 1, which has been implemented twice in open source software: Once, in the *Linear-Time-Branching-Time Spectroscope*, a web tool to decide spectra of behavioral equivalence [4], and once in *gpuequiv*, a GPU-accelerated version thereof [27]. Both work on energy games of arbitrary-dimensional vectors of naturals.

The *Spectroscope* is implemented in Scala.js, powering the tool on <https://equiv.io/>. Its energy game class supports positive, negative, and min-updates. Example 1 is reproduced as a unit test in `io.equiv.eqfiddle.game.GaloisEnergyGameTests`.

gpuequiv [27] uses Rust and the WebGPU system to solve declining energy games on the GPU. The core is an implementation of our algorithm split up across several shaders in order to exploit GPU parallelism in `gpuequiv::energygame::EnergyGame`. For the shader implementation, it is particularly important that the buffer size for the Pareto fronts can be bounded, as will be discussed in Section 5. (In order to conserve memory, the shader will initially reserve space for fronts of up to 64 entries during updates, and must retry positions with more space whenever the allocated space does not suffice.) So far, *gpuequiv* only supports declining energy games, for which it includes several tests. According to benchmarking [27, Chapter 5], *gpuequiv* tends to solve the games 10 to 20 times faster than the Scala implementation (even if the latter is compiled and run through JVM instead of JS).

Both implementations increase our confidence that the approach works well in practical scenarios, and hopefully can be of help to others who want to solve similar problems.

² This is formalized as `galois_composition`.

5 Complexity

The complexity of deciding Galois energy games varies based on the energy order and the updates. Since deciding multi-weighted reachability games is PSPACE-complete as stated by Brihaye and Goeminne [8], deciding energy games over vectors of naturals with the component-wise order and only vector-subtraction is PSPACE-hard:

► **Proposition 26.** *Deciding Galois energy games over n -dimensional vectors of naturals with the component-wise order is PSPACE-hard with respect to n .*

In this section, we give more fine-grained upper complexity bounds for the generic algorithm and for the instantiations according to the previous section.

The running time of Algorithm 1 depends on factors beyond input size, such as the occurring updates which affect the number of iterations needed. To analyze the influence of the order on complexity, we introduce functions that will help establish upper bounds for attacker winning budgets, which are then used in our complexity results.

► **Definition 27.** *For $e \in \mathcal{E}$ let $\text{antichain}_{\leq e}$ be the set of antichains in $\{e' \in \mathcal{E} \mid e' \leq e\}$. Then, $\text{hgt}(e)$ is the cardinality of $\{e' \in \mathcal{E} \mid e' \leq e\}$ and $\text{wdh}(e)$ is the maximal cardinality of elements in $\text{antichain}_{\leq e}$, i.e. $\text{wdh}(e) := \sup_{\mathbb{N}} \{|A| \mid A \in \text{antichain}_{\leq e}\}$.*

Fixing even more parameters, we can state our complexity result.

► **Theorem 28 (Complexity).** *Let $\mathcal{G} = (G, G_a, \succrightarrow)$ be a Galois energy game over (\mathcal{E}, \leq) . Further, let*

- o be the branching degree of the underlying graph,
- t_{sup} be an upper bound of the time to compute the supremum of two energies,
- t_{\leq} be an upper bound of the time to compute the comparison of two energies,
- $t_{\mathcal{G}}$ be an upper bound of the time to compute $u^{\mathcal{G}}(e)$ for any update u in \mathcal{G} and $e \in \mathcal{E}$, and
- $e_{\text{worst}} := \sup\{u_1^{\mathcal{G}} \circ \dots \circ u_i^{\mathcal{G}}(\min \mathcal{E}) \mid i \in \{0, \dots, |G| - 1\} \wedge u_1, \dots, u_i \text{ are updates in } \mathcal{G}\}$ be the highest energy obtainable using up to $|G| - 1$ reverse updates applied to the least element of \mathcal{E} .

Then, Algorithm 1 on input \mathcal{G} terminates in a time in

$$\mathcal{O}(|G|^2 \cdot o \cdot \text{hgt}(e_{\text{worst}}) \cdot \text{wdh}(e_{\text{worst}})^2 \cdot (t_{\mathcal{G}} + t_{\text{sup}} + \text{wdh}(e_{\text{worst}}) \cdot t_{\leq})),$$

and in $\mathcal{O}(|G|^2 \cdot o \cdot \text{wdh}(e_{\text{worst}})^2 \cdot (t_{\mathcal{G}} + t_{\text{sup}} + \text{wdh}(e_{\text{worst}}) \cdot t_{\leq})),$ if \mathcal{G} is declining.

In both cases, the output is calculated using space in $\mathcal{O}(|G| \cdot \text{wdh}(e_{\text{worst}}) \cdot s_{\mathcal{E}})$ where $s_{\mathcal{E}}$ is an upper bound for the space needed to store any energy.

For a detailed proof, we refer to the appendix. Compared to the approach by Bisping [3], this result significantly improves complexity – see Table 3. The key argument leading to this improvement is the calculation of line 9 of Algorithm 1, i.e. $\text{win}[g]$ for a defender position g . Inspired by Brihaye and Goeminne [8], we apply the following procedure:

```

1 def compute_new_win( $\mathcal{G}$ , old_win,  $g$ ):
2   new := Min  $\mathcal{E}$ 
3   for  $g'$  with  $g \succrightarrow g'$ :
4     new := Min $\{\sup\{e', u^{\mathcal{G}}(e_{g'})\} \mid e' \in \text{new} \wedge e_{g'} \in \text{old\_win}[g']\}$ 
5   return new

```

Table 3 compares the complexity results we obtain by Theorem 28 to results stated by Bisping [3] on declining energy games and Brihaye and Goeminne [8] on multi-weighted reachability games, respectively. Further, it states the complexity of Algorithm 1 when applied to a shortest path problem as discussed in Proposition 17 and for energy games with updates in $\mathbb{Z}^n \subseteq \mathcal{U}_n$, i.e. integer vector addition, subsuming the complexity of solving coverability on VASS as seen in Proposition 19.

To do so, we first give upper bounds for some variables: In all cases, we over-approximate o with $|G|$ when applying Theorem 28. Further, we assume $t_{\leq_l}, t_{\leq_c}, t_{\sup} \in \mathcal{O}(n)$. We use w as an upper bound of the absolute value of the integers being added to a component at any edge.³ Then, we have $e_{worst} = w \cdot (|G|, \dots, |G|)$ and $\text{hgt}(e_{worst}) \in \mathcal{O}((w \cdot |G|)^n)$. Other upper bounds are stated in Table 3, where all entries are to be understood in big \mathcal{O} -notation.

■ **Table 3** Application of Theorem 28 to energy games over vectors of naturals.

Game model	$t_{\mathcal{G}}$	$\text{wdh}(e_{worst})$	time complexity	complexity in cited work
Declining energies [3]	n^2	$ G ^{n-1}$	$n^2 \cdot G ^{3n}$	$ \rightarrow \cdot G ^n \cdot (o + G ^{(n-1)o})$
Multi-reachability, component-wise \leq_c [8]	n	$w^{n-1} \cdot G ^{n-1}$	$n \cdot w^{3(n-1)} \cdot G ^{3n}$	$n^4 \cdot w^{4n} \cdot G ^{4n+1}$
Multi-reachability, lexicographic \leq_l [8]	n	1	$n \cdot G ^3$	$n \cdot G ^3$
Shortest paths (with \mathbb{N} -edges)	1	1	$ G ^3$.
Integer vector addition, component-wise \leq_c	n	$w^{n-1} \cdot G ^{n-1}$	$n \cdot w^{4n-3} \cdot G ^{4n}$.

Due to similarities in both the algorithm and approximation techniques, it is not surprising that our complexity result (stated in Table 3) for multi-weighted reachability games with the lexicographic order coincides with previous work by Brihaye and Goeminne [8]. The improvement, when considering the component-wise order, can be attributed primarily to a finer approximation. However, we do improve Bisping's [3] result (in the case of a branching degree greater than two). This comparison demonstrates that our algorithm achieves a competitive performance while offering generality.

6 How to Handle Other Winning Conditions

Reachability winning conditions are fundamental to the analysis of games, both in their simplicity and their importance. In this section, we examine how other types of winning conditions relate to reachability (Definition 5 and 6) studying related games.

6.1 Within the Scope: Captured Games

Weak Upper Bounds. The defender winning if the energy level becomes undefined is another way of saying that there is a lower bound for the energy (and the attacker is energy-bound). We now consider energy games with an additional *upper bound* [15, 16] of resources the attacker may hold. A *weak upper bound* caps the energies. In Example 1, the number of cups resembles a weak upper bound of the number of shots. Similarly, a weak upper bound can be added to any energy game, where the minimum is monotonic, by adding a new dimension for each dimension that should be bounded and composing every update with the function taking the minimum of the dimension and its upper bound. Then, Algorithm 1 can be used in the case of both, a known and an unknown weak upper bound.

³ To get a fair comparison we consider the complexity of declining energy games as formally defined by Bisping [3] where an update may subtract at most one from a component, i.e. $w = 1$.

Generalized Reachability. *Generalized reachability* [9] refers to reachability games with multiple target sets where the attacker wins a play if each set has been visited at least once (without running out of energy). Combining the construction in Remark 8 and that for weak upper bounds, an energy game can model a generalized reachability game by encoding the target sets using additional dimensions that track the visited sets. For this construction, we add positions and edges such that the attacker may choose after reaching any target set, whether to continue or to move to the only deadlock defender position at the cost of one in every dimension corresponding to a target set. Note that this construction increases the dimension and thereby increases time complexity of our algorithm for computing minimal winning budgets exponentially in the number of target sets.

6.2 At the Borders: Related but Distinct Games

Strong Upper Bounds. We reconsider energy games with an upper bound of the attackers resources. Exceeding a *strict upper bound* [16] results in the defender winning. Such an upper bound can be integrated similarly to a weak upper bound by adding dimensions for the bounded dimensions and instead of taking the minimum adding the difference between the energy and the updated energy in the dimensions corresponding to the bounds at every edge. (Inspired by Juhl et al. [16] we additionally apply the mapping $b_i \mapsto b_i + (e_i - u(e)_i)$ to each added bound b_i for the bounded dimension i when $e \mapsto u(e)$ is calculated.) However, if the updates are more complex than vector addition, this construction might not yield a Galois energy game – the game might not even be monotonic.

Safety Winning Conditions. Instead of aiming to reach a target set, the goal of *safety* [1] winning conditions is to avoid a set of unsafe positions. This corresponds to changing the winner of infinite plays to the attacker (if the energy level does not become undefined). Proving decidability then is symmetric to the proofs presented in this paper and can be achieved by adjusting the definitions and statements accordingly. In particular, this includes calculating maximal defender winning budgets instead and defining Galois energy games using Lemma 13.3, i.e. as monotonic energy games where $u^\cup(e') := \max\{e \in \mathcal{E} \uplus \{\perp\} \mid u(e) \leq e'\}$ always exists and is computable. (Here \perp represents undefined energies with $\perp < e$ for all $e \in \mathcal{E}$ and $u(\perp) := \perp$.)

Parity Games. An *energy parity game* [2, 24] is an energy game with a ranking function that assigns each position a natural number (called priority) and has a finite image. Then, infinite plays where the energy level never becomes undefined are won by the attacker if and only if the lowest priority appearing infinitely often is even. A stronger objective for the attacker would be to never visit more than l positions with odd priority before visiting a position with a smaller even priority for a fixed $l \in \mathbb{N}$. Abdulla et al. [2] show that deciding games with that stronger objective in some cases is equivalent to deciding energy parity games. Such energy parity games with vector addition only can be formulated as an energy game with safety winning conditions as described by Chatterjee et al. [11] where dimensions corresponding to the odd priorities are added. Whether such a procedure can be applied to other kinds of updates remains an open question.

6.3 Beyond the Horizon: Games Outside the Framework

Both-Bounded Energy Games. In *both-bounded energy games* [18] both players are energy-bound. This is modeled by assigning two updates to each edge, operating on pairs of current energies – one for the attacker, one for the defender. Considering such games over vectors of

naturals, undecidability arises if one player has one-dimensional energies and the other two, so our results cannot be transferred.

Payoffs of Plays. To compare infinite plays and decide a winner, different *payoffs* [7] can be calculated. The *mean-payoff* refers to the limit average gain per edge, i.e. $\limsup_{n \rightarrow \infty} \frac{1}{n} EL(\rho, e_0, n)$ for a play ρ w.r.t. an initial energy e_0 , while the *total-payoff* is the limit energy level, i.e. $\limsup_{n \rightarrow \infty} EL(\rho, e_0, n)$, and the *average-energy* refers to the limit average energy level, i.e. $\limsup_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^n EL(\rho, e_0, i)$. Since Algorithm 1 relies on the fact that it suffices to consider finite plays, our results cannot be transferred directly.

7 Conclusion

In this paper, we presented a simple algorithm for calculating minimal winning budgets, solving the (un)known initial credit problem for energy games over any well-founded bounded join-semilattice. We generalized two recent approaches [3, 8] and improved the running time. Our results demonstrate that Galois connections offer a simple framework for constructing decidable energy games. Compared to Abdulla et al.’s monotonic games [1] or to Mohri’s generalized shortest-distances [22], our work is more concrete in its application. For a relevant class of energy games over vectors of naturals, we offer an algorithm implementation for practical use. Additionally, we provide an Isabelle/HOL formalization of the general decidability proof, ensuring confidence in our results. By exploring various winning conditions and relating games to Galois energy games, we highlighted the theory’s applicability and limitations and identified areas for further research, such as using Galois connections to classify energy parity games.

References

- 1 Parosh Aziz Abdulla, Ahmed Bouajjani, and Julien d’Orso. Deciding monotonic games. In *Computer Science Logic (CSL)*, volume 2803 of *Lecture Notes in Computer Science*, pages 1–14. Springer Berlin Heidelberg, 2003. doi:10.1007/978-3-540-45220-1_1.
- 2 Parosh Aziz Abdulla, Richard Mayr, Arnaud Sangnier, and Jeremy Sproston. Solving parity games on integer vectors. In *Concurrency Theory (CONCUR)*, volume 8052 of *Lecture Notes in Computer Science*, pages 106–120. Springer Berlin Heidelberg, 2013. doi:10.1007/978-3-642-40184-8_9.
- 3 Benjamin Bisping. Process equivalence problems as energy games. In *Computer Aided Verification (CAV)*, volume 13964 of *Lecture Notes in Computer Science*, pages 85–106. Springer Nature Switzerland, 2023. doi:10.1007/978-3-031-37706-8_5.
- 4 Benjamin Bisping. *Generalized Equivalence Checking of Concurrent Programs*. PhD thesis, Technische Universität Berlin, 2025. URL: <https://generalized-equivalence-checking.equiv.io/>.
- 5 Benjamin Bisping and David N. Jansen. One energy game for the spectrum between branching bisimilarity and weak trace semantics. In Georgiana Caltais and Cinzia Di Giusto, editors, *Proceedings Combined 31st International Workshop on Expressiveness in Concurrency and 21st Workshop on Structural Operational Semantics*, Calgary, Canada, 9th September 2024, volume 412, pages 71–88. Open Publishing Association, 2024. doi:10.4204/EPTCS.412.6.
- 6 Michael Blondin, Matthias Englert, Alain Finkel, Stefan Göller, Christoph Haase, Ranko Lazic, Pierre McKenzie, and Patrick Totzke. The reachability problem for two-dimensional vector addition systems with states. *Journal of the ACM*, 68(5):32–43, 2021. doi:10.1145/3464794.
- 7 Patricia Bouyer, Nicolas Markey, Mickael Randour, Kim G Larsen, and Simon Laursen. Average-energy games. *Acta Informatica*, 55:91–127, 2018. doi:10.1007/s00236-016-0274-1.

- 8 Thomas Brihaye and Aline Goeminne. Multi-weighted reachability games. In *Reachability Problems*, volume 14235 of *Lecture Notes in Computer Science*, pages 85–97. Springer Nature Switzerland, 2023. doi:10.1007/978-3-031-45286-4_7.
- 9 Thomas Brihaye, Aline Goeminne, James C. A. Main, and Mickael Randour. Reachability games and friends: A journey through the lens of memory and complexity. In Patricia Bouyer and Srikanth Srinivasan, editors, *43rd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 284 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 1:1–1:26, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.FSTTCS.2023.1.
- 10 Krishnendu Chatterjee, Laurent Doyen, Thomas A. Henzinger, and Jean-François Raskin. Generalized Mean-payoff and Energy Games. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 8 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 505–516, Dagstuhl, Germany, 2010. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.FSTTCS.2010.505.
- 11 Krishnendu Chatterjee, Mickael Randour, and Jean-François Raskin. Strategy synthesis for multi-dimensional quantitative objectives. In *Concurrency Theory (CONCUR)*, volume 7454 of *Lecture Notes in Computer Science*, pages 115–131. Springer Berlin Heidelberg, 2012. doi:10.1007/978-3-642-32940-1_10.
- 12 Wojciech Czerwiński, Ismaël Jecker, Sławomir Lasota, and Łukasz Orlikowski. Reachability in 3-vass is elementary. *arXiv preprint arXiv:2502.13916*, 2025. doi:10.48550/arXiv.2502.13916.
- 13 Wojciech Czerwiński and Łukasz Orlikowski. Reachability in vector addition systems is ackermann-complete. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1229–1240, 2021. doi:10.1109/FOCS52979.2021.00120.
- 14 Marcel Ern , J rgen Koslowski, Austin Melton, and George E. Strecker. A primer on galois connections. *Annals of the New York Academy of Sciences*, 704(1):103–125, 1993. doi:10.1111/j.1749-6632.1993.tb52513.x.
- 15 Uli Fahrenberg, Line Juhl, Kim G. Larsen, and Ji r  Srba. Energy games in multiweighted automata. In Antonio Cerone and Pekka Pihlajasaari, editors, *Theoretical Aspects of Computing (ICTAC)*, pages 95–115. Springer Berlin Heidelberg, 2011. doi:10.1007/978-3-642-23283-1_9.
- 16 Line Juhl, Kim Guldstrand Larsen, and Jean-Fran ois Raskin. *Optimal Bounds for Multi-weighted and Parametrised Energy Games*, pages 244–255. Springer Berlin Heidelberg, 2013. doi:10.1007/978-3-642-39698-4_15.
- 17 Marvin K nnemann, Filip Mazowiecki, Lia Sch tze, Henry Sinclair-Banks, and Karol W grzycki. Coverability in vass revisited: Improving rackoff’s bound to obtain conditional optimality. In *50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*, volume 261 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 131:1–131:20, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum f r Informatik. doi:10.4230/LIPIcs.ICALP.2023.131.
- 18 Orna Kupferman and Naama Shamash Halevy. Energy games with resource-bounded environments. In *Concurrency Theory (CONCUR)*, volume 243 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 19:1–19:23, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum f r Informatik. doi:10.4230/LIPIcs.CONCUR.2022.19.
- 19 Caroline Lemke. crmrtz/galois-energy-games – A formal proof of decidability of multi-weighted declining energy games. Master’s thesis, Technische Universit t Berlin, 2024. Collection (visited on 2025-08-05). URL: <https://github.com/crmrtz/galois-energy-games>, doi:10.4230/artifacts.24329.
- 20 Caroline Lemke. Galois energy games. *Archive of Formal Proofs*, 2025. Formal proof development. URL: https://isa-afp.org/entries/Galois_Energy_Games.html.
- 21 Caroline Lemke and Benjamin Bisping. Galois energy games: To solve all kinds of quantitative reachability problems, 2025. doi:10.48550/arXiv.2505.14691.

- 22 Mehryar Mohri. Semiring frameworks and algorithms for shortest-distance problems. *Journal of Automata, Languages and Combinatorics*, 7(3):321–350, 2002. doi:10.25596/jalc-2002-321.
- 23 Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. *Isabelle/HOL – A Proof Assistant for Higher-Order Logic*, volume 2283 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2002. doi:10.1007/3-540-45949-9.
- 24 Sven Schewe, Alexander Weinert, and Martin Zimmermann. Parity games with weights. *Logical Methods in Computer Science*, 15, 2019. doi:10.23638/LMCS-15(3:20)2019.
- 25 Joseph E Stoy. *Denotational semantics : the Scott-Strachey approach to programming language theory*. The MIT Press series in computer science. The MIT Press, 3. print edition, 1981.
- 26 Rob J. van Glabbeek. The linear time-branching time spectrum (extended abstract). In Jos C. M. Baeten and Jan Willem Klop, editors, *Theories of Concurrency: Unification and Extension (CONCUR)*, volume 458 of *Lecture Notes in Computer Science*, pages 278–297. Springer Berlin Heidelberg, 1990. doi:10.1007/BFb0039066.
- 27 Gabriel Vogel. Accelerating process equivalence energy games using WebGPU. Bachelor’s thesis, Technische Universität Berlin, 2024. URL: <https://github.com/Gobbel2000/gpuequiv>.
- 28 Akihisa Yamada and J       Dubut. Complete non-orders and fixed points. *Archive of Formal Proofs*, 2019. URL: https://www.isa-afp.org/entries/Complete_Non_Orders.html.

A

 Appendix

This appendix contains the proofs of selected results from the main body of the paper.

Energy-positional Determinacy

► **Lemma 7.** *Let \mathcal{G} be an energy game over (\mathcal{E}, \leq) . For all positions $g \in G$ and energies $e \in \mathcal{E}$ either the defender or the attacker wins $\mathcal{G}[g, e]$. Further, there exists an energy-positional winning strategy, i.e. a winning strategy that depends only on the current position and energy level.*

Proof. The (energy-positional) determinacy follows from the positional determinacy of parity games. We define a parity game $\mathcal{P}_{\mathcal{G}} := (G_{\mathcal{G}}, G_{\mathcal{G}0}, E_{\mathcal{G}}, r_{\mathcal{G}})$ where

- $G_{\mathcal{G}} := \{(g, e) \mid g \in G, e \in \mathcal{E} \uplus \{\perp\}\},$
- $G_{\mathcal{G}0} := G_{\mathcal{G}} \setminus G_{\mathcal{G}1}$ with $G_{\mathcal{G}1} := \{(g, e) \in G_{\mathcal{G}} \mid g \in G_a \vee e = \perp\},$
- $E_{\mathcal{G}} := \{((g, e), (g', e')) \in G_{\mathcal{G}} \times G_{\mathcal{G}} \mid g \rightsquigarrow g' \wedge e \neq \perp \wedge u(e) = e'\}$ and
- $r_{\mathcal{G}} : G_{\mathcal{G}} \rightarrow \mathbb{N}, (g, e) \mapsto 0.$

This construction is similar to that of a configuration graph. Thereby, a play in $\mathcal{P}_{\mathcal{G}}$ corresponds to a play in \mathcal{G} and vice versa. Note that player 0 wins all infinite plays in $\mathcal{P}_{\mathcal{G}}$ while the defender wins all infinite plays in \mathcal{G} . By construction a position (g, e) is a deadend in $\mathcal{P}_{\mathcal{G}}$ if and only if g is a deadend in \mathcal{G} or $e = \perp$. Hence, a play is won by the defender (resp. attacker) in \mathcal{G} if and only if the corresponding play is won by player 0 (resp. player 1) in $\mathcal{P}_{\mathcal{G}}$.

Let $e \in \mathcal{E}$ and $g \in G$. Since parity games are positionally determined, there exists a positional winning strategy s for player 0 or player 1 when starting in (g, e) . Define $s_p : G^*G_p \rightarrow G$ by setting $s_p(g_0 \dots g_n) := g'$ with $s(g_n, EL(g_0 \dots g_n, e)) = (g', e')$ if $g_n \in G_p$, where $G_p = G_d$ if player 0 wins and $G_p = G_a$ else. Note that this strategy is an energy-positional winning strategy. ◀

Complexity

► **Theorem 28 (Complexity).** *Let $\mathcal{G} = (G, G_a, \rightsquigarrow)$ be a Galois energy game over (\mathcal{E}, \leq) . Further, let*

- o be the branching degree of the underlying graph,

- t_{sup} be an upper bound of the time to compute the supremum of two energies,
- t_{\leq} be an upper bound of the time to compute the comparison of two energies,
- $t_{\mathcal{G}}$ be an upper bound of the time to compute $u^{\mathcal{G}}(e)$ for any update u in \mathcal{G} and $e \in \mathcal{E}$, and
- $e_{\text{worst}} := \sup\{u_1^{\mathcal{G}} \circ \dots \circ u_i^{\mathcal{G}}(\min \mathcal{E}) \mid i \in \{0, \dots, |G| - 1\} \wedge u_1, \dots, u_i \text{ are updates in } \mathcal{G}\}$ be the highest energy obtainable using up to $|G| - 1$ reverse updates applied to the least element of \mathcal{E} .

Then, Algorithm 1 on input \mathcal{G} terminates in a time in

$$\mathcal{O}(|G|^2 \cdot o \cdot \text{hgt}(e_{\text{worst}}) \cdot \text{wdh}(e_{\text{worst}})^2 \cdot (t_{\mathcal{G}} + t_{\text{sup}} + \text{wdh}(e_{\text{worst}}) \cdot t_{\leq})),$$

and in $\mathcal{O}(|G|^2 \cdot o \cdot \text{wdh}(e_{\text{worst}})^2 \cdot (t_{\mathcal{G}} + t_{\text{sup}} + \text{wdh}(e_{\text{worst}}) \cdot t_{\leq}))$, if \mathcal{G} is declining.

In both cases, the output is calculated using space in $\mathcal{O}(|G| \cdot \text{wdh}(e_{\text{worst}}) \cdot s_{\mathcal{E}})$ where $s_{\mathcal{E}}$ is an upper bound for the space needed to store any energy.

Proof. We start our arguments by considering the number of needed iterations. If all successors of a defender position g_d were previously assigned a non-empty set, then g_d will be assigned a non-empty set in the next iteration. Similarly, if any successor of an attacker position g_a was assigned a non-empty set, then g_a will be assigned a non-empty set next. After $|G|$ iterations the information, which positions are winnable for the attacker, has traveled from defender deadlocks to all other winnable positions, i.e. $\forall g \in G. \text{Win}_a(g) \neq \emptyset \iff \text{Iteration}^{|G|}(\mathbf{0})(g) \neq \emptyset$. This allows us to calculate an upper bound for the energies being assigned to positions during the run of the algorithm, i.e. e_{worst} .

If \mathcal{G} is declining, then cycles do not benefit the attacker and a fixed point is reached in $\mathcal{O}(|G|)$ iterations. In both, declining and non-declining games, only sets in $\text{antichain}_{\leq e_{\text{worst}}}$ are assigned to positions as candidates for minimal attacker winning budgets. Each set in $\text{antichain}_{\leq e_{\text{worst}}}$ can only truly be updated at most as many times as the maximal length of chains in $\text{antichain}_{\leq e_{\text{worst}}}$ w.r.t. inclusion of upward-closures, i.e. at most $\sup_{\mathbb{N}}\{i \in \mathbb{N} \mid \exists A_1, \dots, A_i \in \text{antichain}_{\leq e_{\text{worst}}}. \forall j \in \{1, \dots, i-1\}. A_j \subsetneq \uparrow A_{j+1}\} = \text{hgt}(e_{\text{worst}})$ times. Thus, in non-declining energy games, the algorithm terminates after at most $1 + |G| + |G| \cdot \text{hgt}(e_{\text{worst}}) \in \mathcal{O}(|G| \cdot \text{hgt}(e_{\text{worst}}))$ iterations of the while-loop.

The computing time needed for operations outside the while-loop is negligible. We now focus on the time needed to execute one iteration, where line 7's time is dominated by that of line 9. The following procedure inspired by Brihaye and Goeminne [8] calculates line 9, i.e. $\text{win}[g]$ for a defender position g .

```

1 def compute_new_win( $\mathcal{G}$ , old_win,  $g$ ):
2   new := Min  $\mathcal{E}$ 
3   for  $g'$  with  $g \succ^u g'$  :
4     new := Min{sup{ $e'$ ,  $u^{\mathcal{G}}(e_{g'})$ } |  $e' \in \text{new} \wedge e_{g'} \in \text{old\_win}[g']$ }
5   return new

```

Note that the repeated application of Min ensures that $\text{wdh}(e_{\text{worst}})$ is an upper bound for the size of new as well as $\text{old_win}[g']$ for all $g' \in G$. Therefore, line 4 in the procedure can be calculated in $\mathcal{O}(\text{wdh}(e_{\text{worst}})^2 \cdot (t_{\mathcal{G}} + t_{\text{sup}} + \text{wdh}(e_{\text{worst}}) \cdot t_{\leq}))$. The for-loop in line 3 adds a factor of the branching degree o , while line 5 adds $|G|$. This yields a running time of Algorithm 1 in $\mathcal{O}(|G|^2 \cdot o \cdot \text{hgt}(e_{\text{worst}}) \cdot \text{wdh}(e_{\text{worst}})^2 \cdot (t_{\mathcal{G}} + t_{\text{sup}} + \text{wdh}(e_{\text{worst}}) \cdot t_{\leq}))$ where $\text{hgt}(e_{\text{worst}})$ may be omitted in the case of declining energy games.

Since each $\text{win}[g]$ calculated contains at most $\text{wdh}(e_{\text{worst}})$ elements, the space needed to compute the output is proportional the space of $|G| \cdot \text{wdh}(e_{\text{worst}})$ energies. ◀