Solving Mean-Payoff Games via Quasi Dominions*

Massimo Benerecetti[a], Daniele Dell'Erba[b], Fabio Mogavero[a]

[a] *Università degli Studi di Napoli Federico II, Naples, Italy*
[b] *University of Liverpool, Liverpool, UK*

Abstract

We propose a novel algorithm for the solution of *mean-payoff games* that merges together two seemingly unrelated concepts introduced in the context of parity games, namely *small progress measures* and *quasi dominions*. We show that the integration of the two notions can be highly beneficial and significantly speeds up convergence to the problem solution. Experiments show that the resulting algorithm performs orders of magnitude better than the asymptotically-best solution algorithm currently known, without sacrificing on the worst-case complexity.

1. Introduction

In this article we consider the problem of solving *mean-payoff games*, namely infinite-duration perfect-information two-player games played on weighted directed graphs, each of whose vertexes is controlled by one of the two players. The game starts at an arbitrary vertex and, during its evolution, each player can take moves at the vertexes it controls, by choosing one of the outgoing edges. The moves selected by the two players induce an infinite sequence of vertexes, called play. The payoff of any prefix of a play is the sum of the weights of its edges. A play is winning if it satisfies the game objective, called *mean-payoff objective*, which requires that the limit of the *mean payoff*, taken over the prefixes lengths, never falls below a given *threshold* ν.

Mean-payoff games have been first introduced and studied by Ehrenfeucht and Mycielski (1979), who showed that positional strategies suffice to obtain the optimal value. A slightly generalised version was also considered by Gurvich et al. (1988). Positional determinacy entails that the decision problem for these games lies in NPTIME ∩ CONPTIME due to Zwick and Paterson (1996), and it was later shown to belong to UPTIME ∩ COUPTIME by Jurdziński (1998), being UPTIME the class of unambiguous non-deterministic polynomial time. This result gives the problem a rather peculiar complexity status,

*This work is based on Benerecetti et al. (2020), which appeared in TACAS'20. A direct application of the proposed approach to parity games is reported in Benerecetti et al. (2023).

shared by very few other problems, such as integer factorisation by Fellows and Koblitz (1992) and Agrawal et al. (2004) and parity games by Jurdziński (1998). Despite various attempts of Gurvich et al. (1988); Zwick and Paterson (1996); Pisaruk (1999); Dhingra and Gaubert (2006); Björklund and Vorobyov (2007), no polynomial-time algorithm (in both the number of positions and the representation-size of the maximal weights) for the mean-payoff game problem is known so far.

A different formulation of the game objective allows to define another class of quantitative games, known as *energy games*. The *energy objective* requires that, given an initial value c, called *credit*, the sum of c and the *payoff* of every prefix of the play never falls below 0. These games, however, are tightly connected to mean-payoff games, as the two type of games have been proved to be log-space equivalent by Bouyer et al. (2008). They are also related to other more complex forms of quantitative games. In particular, unambiguous polynomial-time reductions by Jurdziński (1998) exist from these games to *discounted payoff*, see Zwick and Paterson (1996) and *simple stochastic games*, see Condon (1992).

Recently, a fair amount of work in formal verification has been directed towards considering, besides correctness properties of computational systems, also quantitative specifications, in order to express performance measures and resource requirements, such as quality of service, bandwidth and power consumption and, more generally, bounded resources. Mean-payoff and energy games also have important practical applications in system verification and synthesis. Bloem et al. (2009) show how quantitative aspects, interpreted as penalties and rewards associated with the system choices, allow for expressing optimality requirements encoded as mean-payoff objectives for the automatic synthesis of systems that also satisfy parity objectives. With similar application contexts in mind, Boker et al. (2011) and Bohy et al. (2013) further contribute to that effort, by providing complexity results and practical solutions for the verification and automatic synthesis of reactive systems from quantitative specifications expressed in linear time temporal logic extended with mean-payoff and energy objectives. Further applications to temporal networks have been studied by Comin and Rizzi (2015) and Comin et al. (2017). Consequently, efficient algorithms to solve mean-payoff games become essential ingredients to tackle these problems in practice.

Several algorithms have been devised in the past for the solution of the decision problem for mean-payoff games, which asks whether there exists a strategy for one of the players that grants the mean-payoff objective. The very first deterministic algorithm was proposed by Zwick and Paterson (1996), where it is shown that the problem can be solved with $\mathrm{O}(n^3 \cdot m \cdot W)$ arithmetic operations, with n and m the number of positions and moves, respectively, and W the maximal absolute weight in the game. An indirect strategy improvement approach, based on iteratively adjusting a randomly chosen initial strategy for one player until a winning strategy is obtained, is presented by Schewe (2008), which has an exponential upper bound. The algorithm by Lifshits and Pavlov (2007), which runs in time $\mathrm{O}(n \cdot m \cdot 2^n \cdot \log_2 W)$, computes the "potential" of each game position, which corresponds to the initial credit that the player needs in order to win the game from that position. Algorithms based on the solution of linear feasibility problems over the tropical semiring have been also provided by Allamigeon et al. (2014b,a, 2015). One of the best known deterministic algorithm to date, which requires $O(n \cdot m \cdot W)$ arithmetic operations, was proposed by Brim et al. (2011). They adapt to energy and mean-payoff games the notion of progress measures of Klarlund (1991), as applied to parity games by Jurdziński (2000). The approach was further developed by Comin and Rizzi (2017) to obtain the same complexity bound for the optimal strategy synthesis problem. A strategy-improvement refinement of this technique has been introduced by Brim and Chaloupka (2012). Björklund et al. (2004), instead, proposed a randomised strategy-improvement based algorithm requiring $\mathsf{min}\{O(n^2 \cdot m \cdot W), 2^{O(\sqrt{n} \cdot \log n)}\}$ arithmetic operations. Finally, it has been recently proposed an algorithm for energy games (and therefore also for mean-payoff games) based on progress measures whose number of arithmetic operations is $O(\mathsf{min}(m \cdot n \cdot W,\ m \cdot n \cdot 2^{n/2} \cdot \log_2 W))$ by Dorfman et al. (2019). This improvement of Brim et al. (2011) uses two new ideas: it predicts sequences of update steps that would be performed repetitively in the original approach, and applies the *scaling* technique introduced by Gabow and Tarjan (1991); Goldberg (1995); Goldberg and Rao (1998). In particular the scaling algorithm halves all weights in the game and solves the halved game recursively. Once the solution of the halved game is returned, it is converted into a solution of the original game. As stated by the authors, this scaling technique seems to be required in order to use the first idea and obtain the exponential upper bound on the number of positions.

Our contribution is a novel mean-payoff progress measure approach that enriches such measures with the notion of *quasi dominions*, originally introduced by Benerecetti et al. (2016c, 2018b) to efficiently solve parity games with the *priority promotion* approach (see also Benerecetti et al. (2016a,b, 2018a)). These are sets of positions with the property that as long as the opponent chooses to play to remain in the set, it loses the game for sure, hence its best choice is always to try to escape. A quasi dominion from where escaping is not possible is a winning set for the other player. A weaker similar notion has been presented by Fearnley (2010), however, the sets here called *snares* do not share all the required properties to work with the priority promotion technique. Progress measure approaches, such as the one of Brim et al. (2011), typically focus on finding the best choices of the opponent and little information is gathered on the other player. In this sense, they are intrinsically asymmetric. Enriching the approach with quasi dominions can be viewed as a way to also encode the best choices of the player, information that can be exploited to speed up convergence significantly. The main difficulty here is that suitable lift operators in the new setting do not enjoy monotonicity. Such a property makes proving completeness of classic progress measure approaches almost straightforward, as monotonic operators do admit a least fixpoint. Instead, the lift operator we propose is only inflationary (specifically, non-decreasing) and, while still admitting fixpoints, see Bourbaki (1949); Witt (1950), need not have a least one. Hence, providing a complete solution algorithm proves more challenging. The advantages, however, are significant. On the one hand, the new algorithm still enjoys the same worst-case complexity, with respect to the weights in the game, of the best known algorithm for the problem proposed by Brim et al. (2011). On the other hand, we show that there exist families of games on which both the approach of Brim et al. (2011) and the one proposed in Dorfman et al. (2019) require a number of operations that can be made arbitrarily larger than the one required by the new approach. Experimental results also witness the fact that this phenomenon is by no means isolated, as the new algorithm performs orders of magnitude better than the algorithm developed by Brim et al. (2011).

2. Preliminaries

A two-player turn-based *arena* is a tuple $\mathcal{A} = \langle \mathrm{Ps}_{\oplus}, \mathrm{Ps}_{\boxminus}, Mv \rangle$, with $\mathrm{Ps}_{\oplus} \cap \mathrm{Ps}_{\boxminus} = \emptyset$ and $\mathrm{Ps} \triangleq \mathrm{Ps}_{\oplus} \cup \mathrm{Ps}_{\boxminus}$, such that $\langle \mathrm{Ps}, Mv \rangle$ is a finite directed graph without sinks (*e.g.* positions with no outgoing moves). $\mathrm{Ps}_{\oplus}$ (*resp.*, $\mathrm{Ps}_{\boxminus}$) is the set of positions of player $\oplus$ (*resp.*, $\boxminus$) and $Mv \subseteq \mathrm{Ps} \times \mathrm{Ps}$ is a left-total relation describing all possible moves. A *path* in $\mathrm{V} \subseteq \mathrm{Ps}$ is a finite or infinite sequence $\pi \in \mathrm{Pth}(\mathrm{V})$ of positions in V compatible with the move relation, *i.e.*, $(\pi_i, \pi_{i+1}) \in Mv$, for all $i \in [0, |\pi| - 1)$. If finite, its last element is denoted with $\mathsf{lst}(\pi)$. A positional *strategy* for player $\alpha \in \{\oplus, \boxminus\}$ on $\mathrm{V} \subseteq \mathrm{Ps}$ is a function $\sigma_\alpha \in \mathrm{Str}_\alpha(\mathrm{V}) \subseteq (\mathrm{V} \cap \mathrm{Ps}_\alpha) \to \mathrm{Ps}$, mapping each α-position v in V to a position $\sigma_\alpha(v)$ of the game compatible with the move relation, *i.e.*, $(v, \sigma_\alpha(v)) \in Mv$. With $\mathrm{Str}_\alpha(\mathrm{V})$ we denote the set of all α-strategies on V, while Str_α denotes $\bigcup_{\mathrm{V} \subseteq \mathrm{Ps}} \mathrm{Str}_\alpha(\mathrm{V})$. A *play* in $\mathrm{V} \subseteq \mathrm{Ps}$ from a position $v \in \mathrm{V}$ *w.r.t.* a pair of strategies $(\sigma_\oplus, \sigma_\boxminus) \in \mathrm{Str}_\oplus(\mathrm{V}) \times \mathrm{Str}_\boxminus(\mathrm{V})$, called $((\sigma_\oplus, \sigma_\boxminus), v)$*-play*, is a path $\pi \in \mathrm{Pth}(\mathrm{V})$ such that $\pi_0 = v$ and, for all $i \in [0, |\pi| - 1)$, if $\pi_i \in \mathrm{Ps}_\oplus$ then $\pi_{i+1} = \sigma_\oplus(\pi_i)$ else $\pi_{i+1} = \sigma_\boxminus(\pi_i)$. The *play function* $\mathsf{play} : (\mathrm{Str}_\oplus(\mathrm{V}) \times \mathrm{Str}_\boxminus(\mathrm{V})) \times \mathrm{V} \to \mathrm{Pth}(\mathrm{V})$ returns, for each position $v \in \mathrm{V}$ and pair of strategies $(\sigma_\oplus, \sigma_\boxminus) \in \mathrm{Str}_\oplus(\mathrm{V}) \times \mathrm{Str}_\boxminus(\mathrm{V})$, the maximal $((\sigma_\oplus, \sigma_\boxminus), v)$-play $\mathsf{play}((\sigma_\oplus, \sigma_\boxminus), v)$. If a pair of strategies $(\sigma_\oplus, \sigma_\boxminus) \in \mathrm{Str}_\oplus(\mathrm{V}) \times \mathrm{Str}_\boxminus(\mathrm{V})$ induces a finite play starting from a position $v \in \mathrm{V}$, then $\mathsf{play}((\sigma_\oplus, \sigma_\boxminus), v)$ identifies the maximal prefix of that play that is contained in V. If such pair induces instead an infinite play, then the play is included in V where the strategies are always defined. According to the standard notation, the $\triangleq$ symbol means the object we introduce is by definition equal to something. Given a set A, $\overline{\mathrm{A}}$ is the complement, and a partial map from A to B is denoted with $\mathrm{A} \rightharpoonup \mathrm{B}$.

A *mean-payoff game* (MPG for short) is a tuple $\eth = \langle \mathcal{A}, \mathrm{Wg}, \mathsf{wg} \rangle$, where $\mathcal{A}$ is an arena, $\mathrm{Wg} \subset \mathbb{Z}$ is a finite set of integer weights, and $\mathsf{wg} \colon \mathrm{Ps} \to \mathrm{Wg}$ is a *weight function* assigning a weight to each position. Ps^+ (*resp.*, Ps^-) denotes the set of positive-weight positions (*resp.*, non-positive-weight positions). For convenience, we shall refer to non-positive weights as negative weights. Notice that this definition of MPG is equivalent to the classic formulation in which the weights label the moves, instead. The weight function naturally extends to paths, by setting $\mathsf{wg}(\pi) \triangleq \sum_{i=0}^{|\pi|-1} \mathsf{wg}(\pi_i)$. The goal of player $\oplus$ (*resp.*, $\boxminus$) is to maximise (*resp.*, minimise) $\mathsf{v}(\pi) \triangleq \liminf_{i\to\infty} \frac{1}{i} \cdot \mathsf{wg}(\pi_{\leq i})$, where $\pi_{\leq i}$ is the prefix up to index i. Given a threshold ν, a set of positions $\mathrm{V} \subseteq \mathrm{Ps}$ is a $\oplus$-*dominion*, if there exists a $\oplus$-strategy $\sigma_\oplus \in \mathrm{Str}_\oplus(\mathrm{V})$ such that, for all $\boxminus$-strategies $\sigma_\boxminus \in \mathrm{Str}_\boxminus(\mathrm{V})$ and positions $v \in \mathrm{V}$, the induced play $\pi = \mathsf{play}((\sigma_\oplus, \sigma_\boxminus), v)$ satisfies $\mathsf{v}(\pi) > \nu$. Under this interpretation, the pair of winning sets $(\mathrm{Wn}_\oplus, \mathrm{Wn}_\boxminus)$, also called winning regions, forms a ν-mean partition of the positions of the game. Assuming ν integer, the ν-mean partition problem, in which the goal of player $\oplus$ (*resp.*, $\boxminus$) is to induce plays such that $\mathsf{v}(\pi) > \nu$ (*resp.*, $\mathsf{v}(\pi) \leq \nu$), is equivalent to the 0-mean partition one, as we can subtract ν to the weights of all the positions. As a consequence, the MPG decision problem can be equivalently restated as deciding whether player $\oplus$ (*resp.*, $\boxminus$) has a strategy to enforce $\liminf_{i\to\infty} \frac{1}{i} \cdot \mathsf{wg}(\pi_{\leq i}) > 0$ (*resp.*, $\liminf_{i\to\infty} \frac{1}{i} \cdot \mathsf{wg}(\pi_{\leq i}) \leq 0$), for all the resulting plays π.

3. Solving Mean-Payoff Games via Progress Measures

The abstract notion of progress measure, see Klarlund (1991), has been introduced as a way to encode global properties on paths of a graph by means of simpler local properties of adjacent vertexes. In the context of MPGs, the graph property of interest, called *mean-payoff property*, requires that the mean payoff of every infinite path in the graph be non-positive. More precisely, in game theoretic terms, a *mean-payoff progress measure* witnesses the existence of strategy $\sigma_\boxminus$ for player $\boxminus$ such that each path in the graph induced by fixing that strategy on the arena satisfies the desired property. A mean-payoff progress measure associates with each vertex of the underlying graph a value, called *measure*, taken from the set of extended natural numbers $\mathbb{N}_\infty \triangleq \mathbb{N} \cup \{\infty\}$, endowed with an ordering relation $\leq$ and an addition operation $+$, which extends the standard ordering and addition over the naturals in the usual way. Measures are associated with positions in the game and the measure of a position v can intuitively be interpreted as an estimate of the payoff that player $\oplus$ can enforce on the plays starting in v. In this sense, they measure "how far" v is from satisfying the mean-payoff property, with positive (*resp.* negative) weights pushing away from (*resp.*, towards) the property and the maximal measure ∞ denoting failure of the property for v. More precisely, the $\boxminus$-strategy induced by a progress measure ensures that measures do not increase along the paths of the induced graph. This ensures that every path eventually gets trapped in a non-positive-weight cycle, witnessing a win for player $\boxminus$.

To obtain a progress measure, one starts from some suitable association of position of the game with measures. The local information encoded by these measures is then propagated back along the edges of the underlying graph so as to associate with each position the information gathered along plays of some finite length starting from that position. The propagation process is performed according to the following intuition. The measures of positions adjacent to v (*i.e.* successors of v) are propagated back to v only if those measures push v further away from the property. This propagation is achieved by means of a measure stretch operation $+$, which adds, when appropriate, the measure of an adjacent position to the weight of a given position. This is established by comparing the measure of v with those of its adjacent positions, since, for each position v, the mean-payoff property is defined in terms of the sum of the weights encountered along the plays from that position. The process ends when no position can be pushed further away from the property and each position is not dominated by any, respectively one, of its adjacents, depending on whether that position belongs to player $\oplus$ or to player $\boxminus$, respectively. The positions that did not reach measure ∞ are those from which player $\boxminus$ can win the game and the set of measures currently associated with such positions forms a mean-payoff progress measure for the game.

To make the above intuitions precise, we introduce the notion of measure function, progress measure, and an algorithm for computing progress measures correctly. It is worth noticing that the progress-measure based approach as described by Brim et al. (2011), called Small Energy Progress Measure (SEPM for short) from now on, can be easily recast equivalently in the form below. A *measure function* $\mu\colon \mathrm{Ps} \to \mathbb{N}_\infty$ maps each position v in the game to a suitable measure $\mu(v)$. The order $\leq$ of the measures naturally induces a pointwise partial order $\sqsubseteq$ on the measure functions defined in the usual way, namely, for any two measure functions μ_1 and μ_2, we write $\mu_1 \sqsubseteq \mu_2$ if $\mu_1(v) \leq \mu_2(v)$, for all positions v. The set of measure functions over a measure space, together with the induced ordering $\sqsubseteq$, forms a *measure-function space*.

Definition 1 (Measure-Function Space)**.** *The* measure-function space *is the* partial order $\mathcal{F} \triangleq \langle \mathrm{MF}, \sqsubseteq \rangle$ *whose components are defined as follows:*

1. $\mathrm{MF} \triangleq \mathrm{Ps} \to \mathbb{N}_\infty$ *is the set of all functions* $\mu \in \mathrm{MF}$*, called* measure functions*, mapping each position* $v \in \mathrm{Ps}$ *to a measure* $\mu(v) \in \mathbb{N}_\infty$*;*
2. *for all* $\mu_1, \mu_2 \in \mathrm{MF}$*, it holds that* $\mu_1 \sqsubseteq \mu_2$ *if* $\mu_1(v) \leq \mu_2(v)$*, for all* $v \in \mathrm{Ps}$.

The $\oplus$-denotation *(*resp.*,* $\boxminus$-denotation*) of a measure function* $\mu \in \mathrm{MF}$ *is the set* $\|\mu\|_\oplus \triangleq \mu^{-1}(\infty)$ *(*resp.*,* $\|\mu\|_\boxminus \triangleq \overline{\mu^{-1}(\infty)}$*) of all positions having maximal (*resp.*, non-maximal) measure associated within* μ*.*

Consider a position v with a edge to an adjacent u which, in turn, has measure η. A measure update of η *w.r.t.* v is obtained by the stretch operator $+\colon \mathbb{N}_\infty \times \mathrm{Ps} \to \mathbb{N}_\infty$, defined as

$$\eta + v \triangleq \mathsf{max}\{0, \eta + \mathsf{wg}(v)\},$$

which corresponds to the payoff estimate that the given position will obtain by choosing to follow the move leading to the position u.

A *mean-payoff progress measure* is such that the measure associated with each game position v needs not be increased further in order to beat the actual payoff of the plays starting from v. In particular, it can be defined by taking into account the opposite attitude of the two players in the game. While the player $\oplus$ tries to push towards higher measures, the player $\boxminus$ will try to keep the measures as low as possible. A measure function in which the payoff of each $\oplus$-position (*resp.*, $\boxminus$-position) v is not dominated by the payoff of all (*resp.*, some of) its adjacents augmented with the weight of v itself meets the requirements.

Definition 2 (Progress Measure). *A measure function $\mu \in \mathrm{MF}$ is a* progress measure *if the following two conditions hold true, for all positions $v \in \mathrm{Ps}$:*

1. *$\mu(u) + v \leq \mu(v)$, for all adjacents $u \in Mv(v)$ of v, if $v \in \mathrm{Ps}_{\oplus}$;*
2. *$\mu(u) + v \leq \mu(v)$, for some adjacent $u \in Mv(v)$ of v, if $v \in \mathrm{Ps}_{\boxminus}$.*

The notion can be further restricted to subsets of positions by only considering the subgame induced by the given subset. The following theorem states the fundamental property of progress measures, namely, that every position with a non-maximal measures is won by player $\boxminus$.

Theorem 1 (Progress Measure). *$\|\mu\|_{\boxminus} \subseteq \mathrm{Wn}_{\boxminus}$, for all progress measures μ.*

In order to obtain a progress measure from a given measure function, one can iteratively adjust the current measure values in such a way to force the above progress condition among adjacent positions. To this end, we define the *lift operator* $\mathsf{lift}\colon \mathrm{MF} \to \mathrm{MF}$ as follows:

$$\mathsf{lift}(\mu)(v) \triangleq \begin{cases} \mathsf{max}\{\mu(w) + v \mid w \in Mv(v)\}, & \text{if } v \in \mathrm{Ps}_{\oplus}; \\ \mathsf{min}\{\mu(w) + v \mid w \in Mv(v)\}, & \text{otherwise.} \end{cases}$$

Note that the lift operator is clearly monotone and, therefore, admits a least fixpoint. A mean-payoff progress measure can be obtained by repeatedly applying this operator until a fixpoint is reached, starting from the minimal measure function $\mu_0 \triangleq \{v \in \mathrm{Ps} \mapsto 0\}$ that assigns measure 0 to all the positions in the game. The following *solver operator* applied to μ_0 computes the desired solution:

$$\mathsf{sol} \triangleq \mathsf{lfp}\, \mu \,.\, \mathsf{lift}(\mu)\colon \mathrm{MF} \to \mathrm{MF}.$$

Observe that the measures generated by the procedure outlined above have a fairly natural interpretation. Each positive measure, indeed, under-approximates the weight that player $\oplus$ can enforce along finite prefixes of the plays from the corresponding positions. This follows from the fact that, while player $\oplus$ maximises its measures along the outgoing moves, player $\boxminus$ minimises them. In this sense, each positive measure witnesses the existence of a positively-weighted finite prefix of a play that player $\oplus$ can enforce.

Let $S \triangleq \sum \{\mathsf{wg}(v) \in \mathbb{N} \mid v \in \mathrm{Ps} \wedge \mathsf{wg}(v) > 0\}$ be the sum of all the positive weights in the game. Clearly, the maximal payoff of a simple play (*i.e.*, a play with no repeated positions) in the underlying graph cannot exceed S. Therefore, a measure greater than S witnesses the existence of a cycle whose payoff diverges to infinity and is, thus, won by player $\oplus$. Hence, any measure strictly greater than S can be substituted with the value ∞. This observation establishes the termination of the algorithm and is instrumental to its completeness proof. Indeed, at the fixpoint, the measures actually coincide with the highest payoff player $\oplus$ is able to guarantee. Soundness and completeness of the above procedure have been established in Brim et al. (2011), where the authors also show that, despite the algorithm requiring $\mathrm{O}(n \cdot S) = \mathrm{O}(n^2 \cdot W)$ lift operations in the worst-case, with n the number of positions and W the maximal positive weight in the game, the overall cost of these lift operations is $\mathrm{O}(S \cdot m \cdot \log S) = \mathrm{O}(n \cdot m \cdot W \cdot \log(n \cdot W))$, with m the number of moves and $\mathrm{O}(\log S)$ the cost of each arithmetic operation necessary to compute the stretch of the measures.

4. Solving Mean-Payoff Games via Quasi Dominions

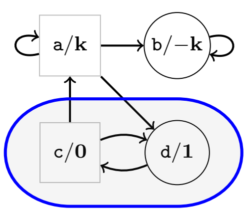


Figure 1: A simple MPG.

Let us consider the simple example game depicted in Figure 1, where the shape of each position indicates the owner, circles for player $\oplus$ and square for its opponent $\boxminus$, and, in each label of the form ℓ/w, the letter w corresponds to the associated weight, where we assume $k > 1$. Starting from the smallest measure function $\mu_0 = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d} \mapsto 0\}$, the first application of the lift operator returns $\mu_1 = \{\mathtt{a} \mapsto k; \mathtt{b}, \mathtt{c} \mapsto 0; \mathtt{d} \mapsto 1\} = \mathsf{lift}(\mu_0)$. After that step, the following iterations of the fixpoint alternatively updates positions c and d, since the other ones already satisfy the progress condition. Being $\mathtt{c} \in \mathrm{Ps}_{\boxminus}$, the lift operator chooses for it the measure computed along the move $(\mathtt{c}, \mathtt{d})$, thus obtaining $\mu_2(\mathtt{c}) = \mathsf{lift}(\mu_1)(\mathtt{c}) = \mu_1(\mathtt{d}) = 1$. Subsequently, d is updated to $\mu_3(\mathtt{d}) = \mathsf{lift}(\mu_2)(\mathtt{d}) = \mu_2(\mathtt{c}) + 1 = 2$. A progress measure is obtained after exactly $2k{+}1$ iterations, when the measure of c reaches value k and d value $k + 1$. Note, however, that the choice of the move $(\mathtt{c}, \mathtt{d})$ is clearly a losing strategy for player $\boxminus$, as remaining in the highlighted region would make the payoff from position c diverge. Therefore, the only reasonable choice for player $\boxminus$ is to exit from that region by taking the move leading to position a. An operator able to diagnose this phenomenon early on could immediately discard the move $(\mathtt{c}, \mathtt{d})$ and jump directly to the correct payoff obtained by choosing the move to position a. As we shall see, such an operator might lose the monotonicity property and recovering the completeness of the resulting approach will prove more involved. In the rest of this article we devise a progress operator that does precisely that. We start by providing a notion of *quasi dominion*, originally introduced for parity games by Benerecetti et al. (2016c, 2018b), which can be exploited in the context of MPGs.

Definition 3 (Quasi Dominion). *A set of positions* $\mathrm{Q} \subseteq \mathrm{Ps}$ *is a* quasi $\oplus$-dominion *if there exists a* $\oplus$*-strategy* $\sigma_\oplus \in \mathrm{Str}_\oplus(\mathrm{Q})$*, called* $\oplus$-witness for Q*, such that, for all* $\boxminus$*-strategies* $\sigma_\boxminus \in \mathrm{Str}_\boxminus(\mathrm{Q})$ *and positions* $v \in \mathrm{Q}$*, the play* $\pi = \mathsf{play}((\sigma_\oplus, \sigma_\boxminus), v)$*, called* $(\sigma_\oplus, v)$-play in Q*, satisfies* $\mathsf{wg}(\pi) > 0$*. If the condition* $\mathsf{wg}(\pi) > 0$ *holds only for infinite plays* π*, then* Q *is called* weak quasi $\oplus$-dominion*.*

Essentially, a quasi $\oplus$-dominion consists in a set Q of positions starting from which player $\oplus$ can force plays in Q of positive weight. Analogously, any infinite play that player $\oplus$ can force to remain in a weak quasi $\oplus$-dominion forever has positive weight. Clearly, any quasi $\oplus$-dominion is also a weak quasi $\oplus$-dominion. Moreover, the latter are closed under subsets, while the former are not. It is an immediate consequence of the definition above that all infinite plays induced by the $\oplus$-witness, if any, necessarily have infinite weight and, thus, are winning for player $\oplus$. Indeed, every such play π is regular, *i.e.* it can be decomposed into a prefix π' and a simple cycle $(\pi'')^\omega$, *i.e.* $\pi = \pi'(\pi'')^\omega$, since the strategies we are considering are memoryless. Now, $\mathsf{wg}((\pi'')^\omega) > 0$, so, $\mathsf{wg}(\pi'') > 0$, which implies $\mathsf{wg}((\pi'')^\omega) = \infty$. Hence, $\mathsf{wg}(\pi) = \infty$.

Proposition 1. *Let* Q *be a weak quasi* $\oplus$*-dominion with* $\sigma_\oplus \in \mathrm{Str}_\oplus(\mathrm{Q})$ *one of its* $\oplus$*-witnesses and* $\mathrm{Q}^\star \subseteq \mathrm{Q}$*. Then, for all* $\boxminus$*-strategies* $\sigma_\boxminus \in \mathrm{Str}_\boxminus(\mathrm{Q}^\star)$ *and positions* $v \in \mathrm{Q}^\star$ *the following holds: if the* $(\sigma_\oplus\restriction_{\mathrm{Q}^\star}, v)$*-play* $\pi = \mathsf{play}((\sigma_\oplus\restriction_{\mathrm{Q}^\star}, \sigma_\boxminus), v)$ *is infinite, then* $\mathsf{wg}(\pi) = \infty$*.*

From Proposition 1, it follows that, if a weak quasi $\oplus$-dominion Q is *closed w.r.t.* its $\oplus$*-witness*, that is all the induced plays that follow the $\oplus$*-witness* strategy are infinite, then it is a $\oplus$-dominion, hence is contained in $\mathrm{Wn}_\oplus$.

Example 1. *Consider again the example of Figure 1. The set of position* $\mathrm{Q} \triangleq \{\mathtt{a}, \mathtt{c}, \mathtt{d}\}$ *forms a quasi* $\oplus$*-dominion whose* $\oplus$*-witness is the only possible* $\oplus$*-strategy mapping position* $\mathtt{d}$ *to* $\mathtt{c}$*. Indeed, any infinite play remaining in* Q *forever and compatible with that strategy (*e.g.*, the play from position* $\mathtt{c}$ *when player* $\boxminus$ *chooses the move from* $\mathtt{c}$ *leading to* $\mathtt{d}$ *or the one from* $\mathtt{a}$ *to itself or the one from* $\mathtt{a}$ *to* $\mathtt{d}$*) grants an infinite payoff. Any finite compatible play, instead, ends in position* $\mathtt{a}$ *(*e.g.*, the play from* $\mathtt{c}$ *when player* $\boxminus$ *chooses the move from* $\mathtt{c}$ *to* $\mathtt{a}$ *and then the one from* $\mathtt{a}$ *to* $\mathtt{b}$*) giving a payoff of at least* $k > 0$*. On the other hand,* $\mathrm{Q}^\star \triangleq \{\mathtt{c}, \mathtt{d}\}$ *is only a weak quasi* $\oplus$*-dominion, as player* $\boxminus$ *can force a play of weight* 0 *from position* $\mathtt{c}$*, by choosing the exiting move* $(\mathtt{c}, \mathtt{a})$*. However, the internal move* $(\mathtt{c}, \mathtt{d})$ *would lead to an infinite play in* $\mathrm{Q}^\star$ *of infinite weight.*

The crucial observation made in Example 1 is that the best choice for player $\boxminus$ in any position of a (weak) quasi $\oplus$-dominion is to exit from it as soon as it can, while the best choice for player $\oplus$ is to remain inside it as long as possible. The idea of the algorithm we propose in this section is to precisely exploit the information provided by the quasi dominions in the following way.

Example 2. *Consider the example above. In position* $\mathtt{a}$ *player* $\boxminus$ *must choose to exit from* $\mathrm{Q} = \{\mathtt{a}, \mathtt{c}, \mathtt{d}\}$*, by taking the move* $(\mathtt{a}, \mathtt{b})$*, without changing its measure, which would correspond to its weight* k*. On the other hand, the best choice for player* $\boxminus$ *in position* $\mathtt{c}$ *is to exit from the weak quasi-dominion* $\mathrm{Q}^\star = \{\mathtt{c}, \mathtt{d}\}$*, by choosing the move* $(\mathtt{c}, \mathtt{a})$ *and lifting its measure from* 0 *to* k*. Note that this contrasts with the minimal measure-increase policy for player* $\boxminus$ *employed by Brim et al. (2011), which would keep choosing to leave* $\mathtt{c}$ *in the quasi-dominion by following the move to* $\mathtt{d}$*, which gives the minimal increase in measure of value* 1*. Once* $\mathtt{c}$ *is out of the quasi-dominion, though, the only possible move for player* $\oplus$

in position d *is to exit towards* c*, which will give* d *measure* $k+1$*. The resulting measure function is the desired progress measure.*

In order to make the intuitive idea of Example 2 precise, we need to be able to identify quasi dominions first. Interestingly enough, the measure functions μ defined in the previous section do allow to identify a quasi dominion, namely the set of positions $\overline{\mu^{-1}(0)}$ having positive measure. Indeed, as observed at the end of that section, a positive measure witnesses the existence of a positively-weighted finite play that player $\oplus$ can enforce from that position onward, which is precisely the requirement of Definition 3.

Example 3. *In the example of Figure 1,* $\overline{\mu_0^{-1}(0)} = \emptyset$ *and* $\overline{\mu_1^{-1}(0)} = \{\mathsf{a}, \mathsf{c}, \mathsf{d}\}$ *are both quasi dominions, the first one* w.r.t. *the empty* $\oplus$*-witness and the second one* w.r.t. *the* $\oplus$*-witness* $\sigma_\oplus(\mathsf{d}) = \mathsf{c}$*.*

We shall keep the quasi-dominion information in pairs (μ, σ), called *quasi-dominion representations* (QDR, for short), composed of a measure function μ and a $\oplus$-strategy σ, which corresponds to one of the $\oplus$-witnesses of the set of positions with positive measure in μ. The connection between these two components is formalised in the definition below that also provides the partial order over which the new algorithm operates.

Definition 4 (QDR Space)**.** *The* quasi-dominion-representation space *is the* partial order $\mathcal{M} \triangleq \langle \mathrm{R}, \sqsubseteq \rangle$*, whose components are defined as follows:*

1. $\mathrm{R} \subseteq \mathrm{MF} \times \mathrm{Str}_\oplus$ *is the set of all pairs* $\varrho \triangleq (\mu_\varrho, \sigma_\varrho) \in \mathrm{R}$*, called* quasi-dominion-representations*, composed of a measure function* $\mu_\varrho \in \mathrm{MF}$ *and a* $\oplus$*-strategy* $\sigma_\varrho \in \mathrm{Str}_\oplus(\mathsf{qsi}(\varrho))$*, where* $\mathsf{qsi}(\varrho) \triangleq \overline{\mu_\varrho^{-1}(0)}$*, for which:*
   - *(a)* $\mathsf{qsi}(\varrho)$ *is a quasi* $\oplus$*-dominion having* σ_ϱ *as a* $\oplus$*-witness;*
   - *(b)* $\|\mu_\varrho\|_\oplus$ *is a* $\oplus$*-dominion;*
   - *(c)* $\mu_\varrho(v) \leq \mu_\varrho(\sigma_\varrho(v)) + v$*, for all* $\oplus$*-positions* $v \in \mathsf{qsi}(\varrho) \cap \mathrm{Ps}_\oplus$*;*
   - *(d)* $\mu_\varrho(v) \leq \mu_\varrho(u) + v$*, for all* $\boxminus$*-positions* $v \in \mathsf{qsi}(\varrho) \cap \mathrm{Ps}_\boxminus$ *and* $u \in Mv(v)$*;*
2. *for all* $\varrho_1, \varrho_2 \in \mathrm{R}$*, it holds that* $\varrho_1 \sqsubseteq \varrho_2$ *if* $\mu_{\varrho_1} \sqsubseteq \mu_{\varrho_2}$ *and* $\sigma_{\varrho_1}(v) = \sigma_{\varrho_2}(v)$*, for all* $\oplus$*-positions* $v \in \mathsf{qsi}(\varrho_1) \cap \mathrm{Ps}_\oplus$ *with* $\mu_{\varrho_1}(v) = \mu_{\varrho_2}(v)$*.*

The α-denotation $\|\varrho\|_\alpha$ *of a* QDR ϱ*, with* $\alpha \in \{\oplus, \boxminus\}$*, is the* α*-denotation* $\|\mu_\varrho\|_\alpha$ *of its measure function.*

Condition 1a is obvious, while Condition 1b, instead, requires every position with infinite measure to be won by player $\oplus$ and is crucial to guarantee the completeness of the algorithm. Finally, Conditions 1c and 1d ensure that every positive measure under approximates the actual weight of some finite play within the induced quasi dominion. This is formally captured by the following proposition, which can be easily proved by induction on the length of the play.

Proposition 2. *Let ϱ be a* QDR *and $v\pi u$ a finite path starting at position $v \in \mathrm{Ps}$ and terminating in position $u \in \mathrm{Ps}$ compatible with the $\oplus$-strategy σ_ϱ. Then, $\mu_\varrho(v) \leq \mathsf{wg}(v\pi) + \mu_\varrho(u)$.*

It is easy to see that every MPG admits a non-trivial QDR space, since the pair (μ_0, σ_0), with μ_0 the smallest measure function and σ_0 the empty strategy, clearly satisfies all the required conditions.

Proposition 3. *Every* MPG *has a non-empty* QDR *space associated with it.*

The solution procedure we propose, called QDPM from now on, can intuitively be broken down as an alternation of two phases. The first one tries to lift the measures of positions outside the quasi dominion $\mathsf{qsi}(\varrho)$ in order to extend it, while the second one lifts the positions inside $\mathsf{qsi}(\varrho)$ that can be forced to exit from it by player $\boxminus$. The algorithm terminates when no new position can be absorbed in the quasi dominion and no measure needs to be lifted to allow the $\boxminus$-winning positions to exit from it, when possible. To this end, we define a controlled lift operator $\mathsf{lift}\colon \mathrm{R} \times 2^{\mathrm{Ps}} \times 2^{\mathrm{Ps}} \rightharpoonup \mathrm{R}$ that works on QDRs and takes two additional parameters, a source and a target set of positions. The intended meaning is that we want to restrict the application of the lift operation to the positions in the source set S, while using only the moves leading to the target set T. The different nature of the two types of lifting operations applied in the two phases is reflected in the actual values of their source and target parameters.

$$\mathsf{lift}(\varrho, \mathrm{S}, \mathrm{T}) \triangleq \varrho^\star, \text{ where}$$

$$\mu_{\varrho^\star}(v) \triangleq \begin{cases} \mathsf{max}\{\mu_\varrho(u) + v \mid u \in Mv(v) \cap \mathrm{T}\}, & \text{if } v \in \mathrm{S} \cap \mathrm{Ps}_\oplus; \\ \mathsf{min}\{\mu_\varrho(u) + v \mid u \in Mv(v) \cap \mathrm{T}\}, & \text{if } v \in \mathrm{S} \cap \mathrm{Ps}_\boxminus; \\ \mu_\varrho(v), & \text{otherwise;} \end{cases}$$

and, for all $\oplus$-positions $v \in \mathsf{qsi}(\varrho^\star) \cap \mathrm{Ps}_\oplus$,

$$\sigma_{\varrho^\star}(v) \in \mathsf{argmax}_{u \in Mv(v) \cap \mathrm{T}}\, \mu_\varrho(u) + v, \text{ if } \mu_{\varrho^\star}(v) \neq \mu_\varrho(v), \text{ and}$$

$$\sigma_{\varrho^\star}(v) = \sigma_\varrho(v), \text{ otherwise.}$$

Except for the restriction on the outgoing moves considered, which are those leading to the targets in T, the lift operator acts on the measure component of a QDR very much like the original lift operator does. In order to ensure that the result is still a QDR, however, the lift operator must also update the $\oplus$-witness of the quasi dominion. This is required to guarantee that Conditions 1a and 1c of Definition 4 are preserved. If the measure of a $\oplus$-position v is not affected by the lift, the $\oplus$-witness must not change for that position. However, if the application of the lift operation increases the measure, then the $\oplus$-witness on v needs to be updated to any move (v, u) that grants measure $\mu_{\varrho^\star}(v)$ to v. In principle, more than one such move may exist and any one of them can serve as witness.

The solution of a game corresponds now to the inflationary fixpoint, see Bourbaki (1949); Witt (1950), of the two phases mentioned above, which are realised by the progress operators prg_0 and prg_+.

$$\mathsf{sol} \triangleq \mathsf{ifp}\, \varrho\, .\, \mathsf{prg}_+(\mathsf{prg}_0(\varrho)) \colon \mathrm{R} \rightharpoonup \mathrm{R}.$$

The first phase is computed by the operator $\mathsf{prg}_0 \colon \mathrm{R} \rightharpoonup \mathrm{R}$:

$$\mathsf{prg}_0(\varrho) \triangleq \mathsf{sup}\{\varrho, \mathsf{lift}(\varrho, \overline{\mathsf{qsi}(\varrho)}, \mathrm{Ps})\}.$$

This operator is responsible for enforcing the progress condition on the positions outside the quasi dominion $\mathsf{qsi}(\varrho)$ that do not satisfy the inequalities between the measures obtained by following the moves leading to $\mathsf{qsi}(\varrho)$. It does that by applying the lift operator with $\overline{\mathsf{qsi}(\varrho)}$ as source and no restrictions on the moves. Those positions that acquire a positive measure in this phase contribute to enlarging the current quasi dominion. Observe that the strategy component of the QDR is updated so that it is a $\oplus$-witness of the new quasi dominion. To guarantee that measures never decrease, the supremum *w.r.t.* the QDR-space ordering is taken as the result.

Lemma 1. *μ_ϱ is a progress measure over $\overline{\mathsf{qsi}(\varrho)}$, for all fixpoints ϱ of prg_0.*

The second phase, instead, implements the mechanism intuitively described above when analysing the simple example of Figure 1. This is achieved by the operator prg_+ reported in Algorithm 1. The procedure iteratively examines the current quasi dominion and lifts the measures of the positions that must exit from it. Specifically, it processes $\mathsf{qsi}(\varrho)$ layer by layer, starting from the outer layer of positions (the ones who have adjacents out of the set) that must escape. The process ends when a, possibly empty, closed weak quasi dominion is obtained. Recall that all the positions in a closed weak quasi dominion are necessarily winning for player $\oplus$, due to Proposition 1. We distinguish two sets of positions in $\mathsf{qsi}(\varrho)$. Those that already satisfy the progress condition and those that do not. The measures of the first ones already witness an escape route from $\mathsf{qsi}(\varrho)$. The other ones, instead, are those whose current choice is to remain inside it.

Algorithm 1: Progress Operator

signature $\mathsf{prg}_+ \colon \mathrm{R} \rightharpoonup \mathrm{R}$
function $\mathsf{prg}_+(\varrho)$
- $\mathrm{Q} \leftarrow \Delta(\varrho)$
- **while** $\mathsf{esc}(\varrho, \mathrm{Q}) \neq \emptyset$ **do**
  - $\mathrm{E} \leftarrow \mathsf{bep}(\varrho, \mathrm{Q})$
  - $\varrho \leftarrow \mathsf{lift}(\varrho, \mathrm{E}, \overline{\mathrm{Q}})$
  - $\mathrm{Q} \leftarrow \mathrm{Q} \setminus \mathrm{E}$
- $\varrho \leftarrow \mathsf{win}(\varrho, \mathrm{Q})$
- **return** ϱ

For instance, when considering the measure function μ_2 in the example of Figure 1, position a belongs to the first set, while positions c and d to the second one, since the choice of c is to follow the internal move (c, d). Since the only positions that change measure are those in the second set, only such positions need to be examined. Observe that those positions form a weak quasi dominion $\Delta(\varrho)$ strictly contained in $\mathsf{qsi}(\varrho)$. To identify them we proceed as follows. First, we collect the set $\mathsf{npp}(\varrho)$ of positions in $\mathsf{qsi}(\varrho)$ that do not satisfy the progress condition, called the *non-progress* *positions*. Then, we compute the set of positions that will have no choice other than reaching $\mathsf{npp}(\varrho)$.

$$\begin{aligned}\mathsf{npp}(\varrho) \triangleq \{v \in \mathsf{qsi}(\varrho) \cap \mathrm{Ps}_{\oplus} \mid \exists u \in Mv(v) \,.\, \mu_{\varrho}(v) < \mu_{\varrho}(u) + v\} \\ \cup \{v \in \mathsf{qsi}(\varrho) \cap \mathrm{Ps}_{\boxminus} \mid \forall u \in Mv(v) \,.\, \mu_{\varrho}(v) < \mu_{\varrho}(u) + v\}.\end{aligned}$$

The remaining positions in $\Delta(\varrho)$ are collected as the inflationary fixpoint of the following operator.

$$\begin{aligned}\mathsf{pre}(\varrho, \mathrm{Q}) \triangleq \mathrm{Q} \cup \{v \in \mathsf{qsi}(\varrho) \cap \mathrm{Ps}_{\oplus} \mid \sigma_{\varrho}(v) \in \mathrm{Q}\} \\ \cup \{v \in \mathsf{qsi}(\varrho) \cap \mathrm{Ps}_{\boxminus} \mid \forall u \in Mv(v) \setminus \mathrm{Q} \,.\, \mu_{\varrho}(v) < \mu_{\varrho}(u) + v\}.\end{aligned}$$

The final result is

$$\Delta(\varrho) \triangleq (\mathsf{ifp}\, \mathrm{Q} \,.\, \mathsf{pre}(\varrho, \mathrm{Q}))(\mathsf{npp}(\varrho)).$$

Intuitively, $\Delta(\varrho)$ contains all the $\oplus$-positions that are forced to reach $\mathsf{npp}(\varrho)$ via the quasi-dominion $\oplus$-witness and all the $\boxminus$-positions that can only avoid reaching $\mathsf{npp}(\varrho)$ by strictly increasing their measure, which is something that player $\boxminus$ obviously wants to prevent.

It is important to observe that, from a functional view-point, the progress operator prg_{+} would work just as well if applied to the entire quasi dominion $\mathsf{qsi}(\varrho)$, since it would simply leave unchanged the measure of those positions that already satisfy the progress condition. However, it is crucial that only the positions in $\Delta(\varrho)$ are processed in order to achieve the best asymptotic complexity bound known to date. We shall reiterate on this point later on. At each iteration of the while-loop of Algorithm 1, let Q denote the current (weak) quasi dominion, initially set to $\Delta(\varrho)$ (Line 1). It first identifies the positions in Q that can immediately escape from it (Line 2). Those are *(i)* all the $\boxminus$-position with a move leading outside of Q and *(ii)* the $\oplus$-positions v whose $\oplus$-witness σ_{ϱ} forces v to exit from Q, namely $\sigma_{\varrho}(v) \notin \mathrm{Q}$, and that cannot strictly increase their measure by choosing to remain in Q. While the condition for $\boxminus$-position is obvious, the one for $\oplus$-positions requires some explanation. The crucial observation here is that, while player $\oplus$ does indeed prefer to remain in the quasi dominion, it can only do so while ensuring that by changing strategy it does not enable infinite plays within Q that are winning for the adversary. In other words, the new $\oplus$-strategy must still be a $\oplus$-witness for Q and this can only be ensured if the new choice strictly increases its measure. The operator $\mathsf{esc}\colon \mathrm{R} \times 2^{\mathrm{Ps}} \to 2^{\mathrm{Ps}}$ formalises the idea:

$$\begin{aligned}\mathsf{esc}(\varrho, \mathrm{Q}) \triangleq \{v \in \mathrm{Q} \cap \mathrm{Ps}_{\boxminus} \mid Mv(v) \setminus \mathrm{Q} \neq \emptyset\} \\ \cup \{v \in \mathrm{Q} \cap \mathrm{Ps}_{\oplus} \mid \sigma_{\varrho}(v) \notin \mathrm{Q} \wedge \forall u \in Mv(v) \cap \mathrm{Q} \,.\, \mu_{\varrho}(u) + v \leq \mu_{\varrho}(v))\}.\end{aligned}$$

Example 4. *Consider, for instance, the example in Figure 2 and a* QDR ϱ *such that* $\mu_{\varrho} = \{\mathtt{a} \mapsto 3; \mathtt{b} \mapsto 2; \mathtt{c}, \mathtt{d}, \mathtt{f} \mapsto 1; \mathtt{e} \mapsto 0\}$ *and* $\sigma_{\varrho} = \{\mathtt{b} \mapsto \mathtt{a}; \mathtt{f} \mapsto \mathtt{d}\}$. *In this case, we have* $\mathrm{Q} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{f}\}$ *and* $\Delta(\varrho) = \{\mathtt{c}, \mathtt{d}, \mathtt{f}\}$, *since* c *is the only non-progress position,* d *is forced to follow* c *in order to avoid the measure increase required to reach* b*, and* f *is forced by the* $\oplus$*-witness to reach* d*. Now,* *consider the situation where the current weak quasi dominion is* $\mathrm{Q} = \{\mathtt{c}, \mathtt{f}\}$, i.e. *after* $\mathtt{d}$ *has escaped from* $\Delta(\varrho)$*. The escape set of* Q *is* $\{\mathtt{c}, \mathtt{f}\}$*. To see why the* $\oplus$*-position* $\mathtt{f}$ *is escaping, observe that* $\mu_{\varrho}(\mathtt{f}) + \mathtt{f} = 1 = \mu_{\varrho}(\mathtt{f})$ *and that, indeed, should player* $\oplus$ *choose to change its strategy and take the move* $(\mathtt{f}, \mathtt{f})$ *to remain in* Q*, it would obtain an infinite play with payoff* 0*, thus violating the definition of weak quasi dominion.*

Before proceeding, we want to emphasise an easy consequence of the definition of the notion of escape set and Conditions 1c and 1d of Definition 4, *i.e.*, that every escape position of the quasi dominion $\mathsf{qsi}(\varrho)$ can only assume its weight as possible measure inside a QDR ϱ, as reported is the following proposition. This observation, together with Proposition 2, ensures that the measure of a position $v \in \mathsf{qsi}(\varrho)$ is an under approximation of the weight of all finite plays leaving $\mathsf{qsi}(\varrho)$.

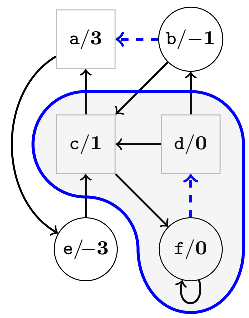


Figure 2: Another MPG.

Proposition 4. *Let* ϱ *be a* QDR*. Then,* $\mu_{\varrho}(v) = \mathsf{wg}(v) > 0$*, for all* $v \in \mathsf{esc}(\varrho, \mathsf{qsi}(\varrho))$*.*

Now, going back to the analysis of the algorithm, if the escape set is non-empty, we select the escape positions that need to be lifted in order to satisfy the progress condition. The main difficulty is to do so in such a way that the resulting measure function still satisfies Condition 1d of Definition 4, for all the $\boxminus$-positions with positive measure. The problem occurs when a $\boxminus$-position can exit either immediately or by following a path leading to another position in the escape set.

Example 5. *Consider again the example above, where* $\mathrm{Q} = \Delta(\varrho) = \{\mathtt{c}, \mathtt{d}, \mathtt{f}\}$*. If position* $\mathtt{d}$ *immediately escapes from* Q *using the move* $(\mathtt{d}, \mathtt{b})$*, it would change its measure to* $\mu'(\mathtt{d}) = \mu(\mathtt{b}) + \mathtt{d} = 2 > \mu(\mathtt{d}) = 1$*. Now, position* $\mathtt{c}$ *has two ways to escape, either directly with move* $(\mathtt{c}, \mathtt{a})$ *or by reaching the other escape position* $\mathtt{d}$ *passing through* $\mathtt{f}$*. The first choice would set its measure to* $\mu(\mathtt{a}) + \mathtt{c} = 4$*. The resulting measure function, however, would not satisfy Condition 1d of Definition 4, as the new measure of* $\mathtt{c}$ *would be greater than* $\mu'(\mathtt{d}) + \mathtt{c} = 2$*, preventing us to obtain a* QDR*. Similarly, if position* $\mathtt{d}$ *escapes from* Q *passing through* $\mathtt{c}$ *via the move* $(\mathtt{c}, \mathtt{a})$*, we would have* $\mu''(\mathtt{d}) = \mu''(\mathtt{c}) + \mathtt{d} = (\mu(\mathtt{a}) + \mathtt{c}) + \mathtt{d} = 4 > 2 = \mu(\mathtt{b}) + \mathtt{d}$*, still violating Condition 1d. Therefore, in this specific case, the only possible way to escape is to reach* $\mathtt{b}$*. The solution to this problem is simply to lift, in the current iteration, only those positions that obtain the lowest possible measure increase, hence position* $\mathtt{d}$ *in the example, leaving the lift of* $\mathtt{c}$ *to some subsequent iteration of the algorithm that would choose the correct escape route via* $\mathtt{d}$*.*

In order to generalize the solution idea given in the example above, we proceed as follows. We start by computing the minimal measure increase, called the *best-escape forfeit*, that each position in the escape set would obtain by exiting the quasi dominion immediately. At this point, the positions with the lowest possible forfeit, called *best-escape positions*, can all be lifted at the same time. The intuition is that the measure of all the positions that escape from a (weak) quasi dominion will necessarily be increased of at least the minimal best-escape forfeit. This observation is at the core of the proof of Theorem 2 (see the appendix) ensuring that the desired properties of QDRs are preserved by the operator prg_{+}. The set of best-escape positions is computed by the operator $\mathsf{bep}\colon \mathrm{R} \times 2^{\mathrm{Ps}} \to 2^{\mathrm{Ps}}$ as follows:

$$\mathsf{bep}(\varrho, \mathrm{Q}) \triangleq \mathsf{argmin}_{v \in \mathsf{esc}(\varrho, \mathrm{Q})} \mathsf{bef}(\mu_{\varrho}, \mathrm{Q}, v),$$

where the operator $\mathsf{bef}\colon \mathrm{MF} \times 2^{\mathrm{Ps}} \times \mathrm{Ps} \to \mathbb{N}_{\infty}$ computes, for each position v in a quasi dominion Q, its best-escape forfeit as follows:

$$\mathsf{bef}(\mu, \mathrm{Q}, v) \triangleq \begin{cases} \mathsf{max}\{\mu(u) + v - \mu(v) \mid u \in Mv(v) \setminus \mathrm{Q}\}, & \text{if } v \in \mathrm{Ps}_{\oplus}; \\ \mathsf{min}\{\mu(u) + v - \mu(v) \mid u \in Mv(v) \setminus \mathrm{Q}\}, & \text{otherwise.} \end{cases}$$

In Example 5, $\mathsf{bef}(\mu, \mathrm{Q}, \mathtt{c}) = \mu(\mathtt{a}) + \mathtt{c} - \mu(\mathtt{c}) = 4 - 1 = 3$, while $\mathsf{bef}(\mu, \mathrm{Q}, \mathtt{d}) = \mu(\mathtt{b}) + \mathtt{d} - \mu(\mathtt{d}) = 2 - 1 = 1$. Therefore, $\mathsf{bep}(\varrho, \mathrm{Q}) = \{\mathtt{d}\}$.

Once the set E of best-escape positions is identified (Line 3), the procedure lifts them restricting the possible moves to those leading outside the current quasi dominion (Line 4). Those positions are, then, removed from the set (Line 5), thus obtaining a smaller weak quasi dominion ready for the next iteration.

The algorithm terminates when the (possibly empty) current quasi dominion Q is closed. By virtue of Proposition 1, all those positions belong to $\mathrm{Wn}_{\oplus}$ and their measure is set to ∞ by means of the operator $\mathsf{w}\colon \mathrm{R} \times 2^{\mathrm{Ps}} \rightharpoonup \mathrm{QDR}$ (Line 6), which also computes the winning $\oplus$-strategy on those positions.

$$\mathsf{win}(\varrho, \mathrm{Q}) \triangleq \varrho^{\star}, \quad \text{where} \quad \mu_{\varrho^{\star}} \triangleq \mu_{\varrho}[\mathrm{Q} \mapsto \infty]$$

and, for all $\oplus$-positions $v \in \mathsf{qsi}(\varrho^{\star}) \cap \mathrm{Ps}_{\oplus}$,

$$\sigma_{\varrho^{\star}}(v) \in \mathsf{argmax}_{u \in Mv(v) \cap \mathrm{Q}} \mu_{\varrho}(u) + v, \text{ if } \sigma_{\varrho}(v) \notin \mathrm{Q} \text{ and } \sigma_{\varrho^{\star}}(v) = \sigma_{\varrho}(v), \text{ otherwise.}$$

Observe that, since we know that every $\oplus$-position $v \in \mathrm{Q} \cap \mathrm{Ps}_{\oplus}$, whose current $\oplus$-witness leads outside Q, is not an escape position, any move (v, u) remaining inside Q that grants the maximal stretch $\mu_{\varrho}(u) + v$ strictly increases its measure and, therefore, is a possible choice for a $\oplus$-witness of the $\oplus$-dominion Q.

At this point, it should be quite evident that the progress operator prg_{+} is responsible for enforcing the progress condition on the positions inside the quasi dominion $\mathsf{qsi}(\varrho)$, thus, the following necessarily holds.

Lemma 2. *μ_{ϱ} is a progress measure over $\mathsf{qsi}(\varrho)$, for all fixpoints ϱ of prg_{+}.*

Example 6. *The lack of monotonicity of the progress operator prg_{+} is illustrated by the following example. Consider the game of Figure 3 and the two* QDR*s $\varrho_1 = (\mu_{\varrho_1}, \sigma_{\varrho_1})$ and $\varrho_2 = (\mu_{\varrho_2}, \sigma_{\varrho_2})$, whose components are: $\mu_{\varrho_1} = \{\mathtt{a} \mapsto 3; \mathtt{b} \mapsto 0; \mathtt{c} \mapsto 2; \mathtt{d}, \mathtt{e} \mapsto 1\}$ and $\sigma_{\varrho_1} = \{\mathtt{e} \mapsto \mathtt{d}\}$; $\mu_{\varrho_2} = \{\mathtt{a} \mapsto 3; \mathtt{b} \mapsto$*

$0; \mathtt{c}, \mathtt{e} \mapsto 2; \mathtt{d} \mapsto 1\}$ and $\sigma_{\varrho_2} = \{\mathtt{e} \mapsto \mathtt{c}\}$. Obviously, $\varrho_1 \sqsubset \varrho_2$. However, $\varrho_1^\star \triangleq \mathsf{prg}_+(\varrho_1) \not\sqsubseteq \varrho_2^\star \triangleq \mathsf{prg}_+(\varrho_2)$. Indeed, $\mu_{\varrho_1^\star} = \{\mathtt{a} \mapsto 3; \mathtt{b} \mapsto 0; \mathtt{c} \mapsto 2; \mathtt{d}, \mathtt{e} \mapsto 4\}$, while $\mu_{\varrho_2^\star} = \{\mathtt{a} \mapsto 3; \mathtt{b} \mapsto 0; \mathtt{c}, \mathtt{e} \mapsto 2; \mathtt{d}, \mapsto 3\}$, which implies that $\varrho_2^\star \sqsubset \varrho_1^\star$. Moreover, $\varrho_1^\star$ is already a progress measure, while $\varrho_2^\star$ requires another application of prg_+ in order to solve the game, since $\varrho_1^\star = \mathsf{prg}_+(\varrho_2^\star)$.

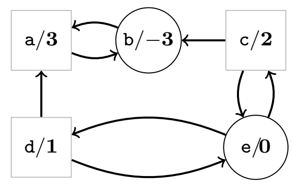


Figure 3: Yet another MPG.

In order to prove the correctness of the proposed algorithm, we first need to ensure that any quasi-dominion space $\mathcal{M}$ is indeed closed under the operators prg_0 and prg_+. This is established by the following theorem, which states that the operators are total functions on that space.

Theorem 2. *The operators prg_0 and prg_+ are total inflationary functions.*

Since both operators are inflationary, so is their composition, which admits a fixpoint. Therefore, the operator sol is well defined. In particular, following the same considerations discussed at the end of Section 3, it can be proved that the fixpoint is obtained after at most $n \cdot (S + 1)$ iterations, where $S \triangleq \sum \{\mathsf{wg}(v) \in \mathbb{N} \mid v \in \mathrm{Ps} \wedge \mathsf{wg}(v) > 0\}$.

Such a bound can actually be improved by noticing that the measure associated with a position at each iteration is always obtained by summing up the weights along some simple path in the quasi dominion leading to an escape position. Let $\mathrm{SPth}(\mathrm{V}) \subseteq \mathrm{Pth}(\mathrm{V})$ be the set of all simple paths over the set of positions $\mathrm{V} \subseteq \mathrm{Ps}$. We say that a QDR $\varrho \in \mathrm{R}$ is *simple* if, for every position $v \in \mathsf{qsi}(\varrho)$ with $\mu_\varrho(v) \neq \infty$, it holds true that $\mu_\varrho(v) = \mathsf{wg}(\pi)$, for some simple path $\pi \in \mathrm{SPth}(\mathsf{qsi}(\varrho))$ ending in an escape position, *i.e.*, $\mathsf{lst}(\pi) \in \mathsf{esc}(\varrho, \mathsf{qsi}(\varrho))$. For a simple QDR ϱ, we say that $\mu_\varrho(v)$ is a *simple measure*. Both operators applied to a simple QDR ϱ return a simple QDR as well. Indeed, prg_0 adds positions outside $\mathsf{qsi}(\varrho)$ to the quasi dominion. Therefore, the measure obtained by each such position v is obtained by adding its weight $\mathsf{wg}(v)$ to the measure of an adjacent position $w \in \mathsf{qsi}(\varrho)$. Hence, since $\mu_\varrho(w)$ is a simple measure by assumption and $v \notin \mathsf{qsi}(\varrho)$, we have that $\mathsf{prg}_0(\varrho)(v)$ is indeed a simple measure. As to prg_+, observe that all positions $v \in \mathsf{qsi}(\varrho)$ outside $\Delta(\varrho)$ have a measure of a simple path leading to $\mathsf{esc}(\varrho, \mathsf{qsi}(\varrho))$ that does not pass through $\Delta(\varrho)$, since, otherwise, v would belong to $\Delta(\varrho)$ by definition. This property is preserved on $\mathsf{qsi}(\varrho) \setminus \mathrm{Q}$ by each iteration of the while-loop (Lines 2-5), since any position v lifted at Line 4 takes a measure obtained by adding its weight $\mathsf{wg}(v)$ to the measure of an adjacent position $w \in \mathsf{qsi}(\varrho) \setminus \mathrm{Q}$, which, by inductive hypothesis, has a simple measure. These observations lead to the following proposition.

Proposition 5. *The operators prg_0 and prg_+ map simple* QDR*s in simple* QDR*s.*

Thanks to this result, the number of iteration required by sol is bounded by $n \cdot (Z + 1)$, where $Z \triangleq |\{\mathsf{wg}(\pi) \in \mathbb{N} \mid \pi \in \mathrm{SPth}(\mathrm{Ps})\}| \leq S$ is the number of possible positive weights associated with a simple paths in the game. The value Z, in turn, is $\mathrm{O}(n \cdot \mathsf{min}\{W, (n-1)!\})$, as there are at most $n!$ simple paths, each one having weight at most equal to $S = \mathrm{O}(n \cdot W)$.

Let $\mathsf{ifp}_k\, X \,.\, \mathsf{F}(X)$ denote the k-th iteration of an inflationary operator F. Then, we have the following theorem.

Theorem 3 (Termination)**.** *The solver operator* $\mathsf{sol} \triangleq \mathsf{ifp}\, \varrho \,.\, \mathsf{prg}_+(\mathsf{prg}_0(\varrho))$ *is a well-defined total function. Moreover, for every simple* QDR $\varrho \in \mathrm{R}$ *it holds that* $\mathsf{sol}(\varrho) = (\mathsf{ifp}_k\, \varrho^\star .\, \mathsf{prg}_+(\mathsf{prg}_0(\varrho^\star)))(\varrho)$*, for some index* $k \leq n \cdot (Z+1)$*, where* n *is the number of positions in the* MPG *and* Z *is the number of positive weights of all its simple paths,* i.e., $Z \triangleq |\{\mathsf{wg}(\pi) \in \mathbb{N} \mid \pi \in \mathrm{SPth}(\mathrm{Ps})\}|$.

As already observed before, Figure 1 exemplifies an infinite family of games with a fixed number of positions and increasing maximal weight k over which the SEPM algorithm requires $2k+1$ iterations of the lift operator. On the contrary, QDPM needs exactly two iterations of the solver operator sol to find the progress measure, starting from the smallest measure function μ_0. Indeed, the first iteration returns a measure function $\mu_1 = \mathsf{sol}(\mu_0)$, with $\mu_1(\mathsf{a}) = k$, $\mu_1(\mathsf{b}) = \mu_1(\mathsf{c}) = 0$, and $\mu_1(\mathsf{d}) = 1$, while the second one $\mu_2 = \mathsf{sol}(\mu_1)$ identifies the smallest progress measure, with $\mu_1(\mathsf{a}) = \mu_1(\mathsf{c}) = k$, $\mu_1(\mathsf{b}) = 0$, and $\mu_1(\mathsf{d}) = k+1$. A more detailed analysis of another family of games exhibiting a similar behaviour is provided at the end of this section. From these observations, the next result immediately follows.

Theorem 4. *An infinite family of* MPG*s* $\{\eth_k\}_k$ *exists on which* QDPM *requires a constant number of measure updates, while* SEPM *requires* $\mathrm{O}(k)$ *such updates.*

From Theorem 1 and Lemmas 1 and 2 it follows that the solution provided by the algorithm is indeed a progress measure, hence establishing soundness.

Theorem 5 (Soundness)**.** $\|\mathsf{sol}(\varrho)\|_{\boxminus} \subseteq \mathrm{Wn}_{\boxminus}$*, for every* $\varrho \in \mathrm{R}$*.*

Completeness follows from Theorem 3 and from Condition 1b of Definition 4 that ensures that all the positions with infinite measure are winning for player $\oplus$.

Theorem 6 (Completeness)**.** $\|\mathsf{sol}(\varrho)\|_{\oplus} \subseteq \mathrm{Wn}_{\oplus}$*, for every* $\varrho \in \mathrm{R}$*.*

The following lemma ensures that each execution of the operator prg_+ strictly increases the measure of all the positions in $\Delta(\varrho)$.

Lemma 3. *Let* $\varrho^\star \triangleq \mathsf{prg}_+(\varrho)$*. Then,* $\mu_{\varrho^\star}(v) > \mu_\varrho(v)$*, for all positions* $v \in \Delta(\varrho)$*.*

Recall that each position can be lifted at most $Z+1 = \mathrm{O}(n \cdot \mathsf{min}\{W, (n-1)!\})$ times and, by the previous lemma, the complexity of sol only depends on the cumulative cost of such lift operations. We can express, then, the total cost as the sum, over the set of positions in the game, of the cost of all the lift operations performed on those positions. Each such operation can be computed in time linear in the number of the incoming and outgoing moves of the corresponding lifted position v, namely $\mathrm{O}\big((|Mv(v)| + |Mv^{-1}(v)|) \cdot \log S\big)$, with $\mathrm{O}(\log S)$ the cost of each arithmetic operation involved. Summing everything up, the actual asymptotic complexity of the procedure can, therefore, be expressed as $\mathrm{O}(n \cdot m \cdot \mathsf{min}\{W, (n-1)!\} \cdot \log(n \cdot W))$.

Example 7. *The following example shows an execution of the algorithm on the game depicted in Figure 4, where $k > 2$. The numbers in the labels of the positions in the first picture labelled (0) denote the weights. In the remaining pictures, instead, they denote the measures assigned in the remaining iterations of the procedure. Each picture also features both the $\oplus$-witness strategy in dashed blue and the best counter $\boxminus$-strategy in dashed red for the current quasi dominion. Moreover, moves along which the measure strictly increases are depicted as solid coloured arrows. Below each picture, we also indicate the phase,* prg_0 *or* prg_+*, that produces the displayed result.*

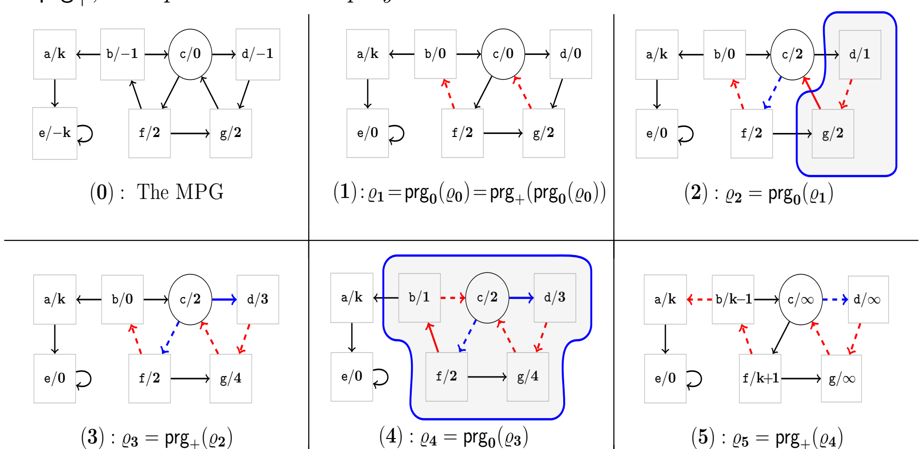


Figure 4: A simulation of a simple MPG.

The computation starts from the initial QDR $\varrho_0 = (\mu_0, \sigma_0)$*, assigning measure* 0 *to all the positions of the game with the associated empty strategy. The first iteration applies* prg_0 *to* ϱ_0*, which lifts positions* $\mathtt{a}$, $\mathtt{f}$*, and* $\mathtt{g}$ *to their respective weights, leading to* ϱ_1 *as shown in Picture* (1)*. At this point,* $\mathsf{qsi}(\varrho_1) = \{\mathtt{a}, \mathtt{f}, \mathtt{g}\}$*, but* $\Delta(\varrho_1)$ *is empty, since all the positions in* $\mathsf{qsi}(\varrho_1)$ *already satisfy the progress condition, thus,* prg_+ *leaves the measures unchanged. In the next iteration,* prg_0 *applied to* ϱ_1 *results in the lifting of positions* $\mathtt{c}$ *and* $\mathtt{d}$*, as reported in Picture* (2)*. Position* $\mathtt{c}$ *is a* $\oplus$*-position and the lift operator chooses* $(\mathtt{c}, \mathtt{f})$ *as its strategy. The resulting quasi-dominion is* $\mathsf{qsi}(\varrho_2) = \{\mathtt{a}, \mathtt{c}, \mathtt{d}, \mathtt{f}, \mathtt{g}\}$ *and* $\Delta(\varrho_2) = \{\mathtt{d}, \mathtt{g}\}$*, with* $\mathtt{g}$ *the only escape position that is also non-progress. The measure of* $\mathtt{g}$ *is lifted to* $\mu_2(\mathtt{c}) + \mathtt{g} = 4$*. Finally, it is the turn of position* $\mathtt{d}$ *to be lifted to* $\mu_2(\mathtt{g}) + \mathtt{d} = 3$*. Picture* (3) *shows the resulting* QDR ϱ_3*. The final iteration first applies* prg_0 *to* ϱ_3 *(Picture* (4)*), lifting position* $\mathtt{b}$ *to measure* 1 *via the move* $(\mathtt{b}, \mathtt{c})$*. This change of measure triggers another application of* prg_+*, as position* $\mathtt{f}$ *is now non-progress. The resulting* QDR ϱ_4 *is such that* $\mathsf{qsi}(\varrho_4) = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{f}, \mathtt{g}\}$ *and* $\Delta(\varrho_4) = \{\mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{f}, \mathtt{g}\}$*. The only escape position is* $\mathtt{b}$*, which is lifted directly to measure* $k - 1$*. In the remaining set* $\{\mathtt{c}, \mathtt{d}, \mathtt{f}, \mathtt{g}\}$*, the only escape position is* $\mathtt{f}$*, which is lifted to measure* $k + 1$*. The resulting weak quasi dominion* $\{\mathtt{c}, \mathtt{d}, \mathtt{g}\}$*, however, is closed, since* $\mu_{\varrho_4}(\mathtt{c}) = 2 < \mu_{\varrho_4}(\mathtt{d}) + \mathtt{c} = 3$*. Therefore, player* $\oplus$ *changes strategy and chooses the move* $(\mathtt{c}, \mathtt{d})$*. Since no escape positions remain,* *the set* $\{\mathtt{c}, \mathtt{d}, \mathtt{g}\}$ *is winning for player* $\oplus$ *and the* win *operator lifts all their measures to* ∞*, leading to* ϱ_5 *in Picture* (5)*. The measure function* μ_5 *is now a progress measure and the algorithm terminates. The total number of single measure updates for* QDPM *to reach the fixpoint is* 13*, regardless of the value of the maximal weight* k *in the game assigned to position* a*.*

The example above, similarly to the one in Figure 1, shows a family of MPGs indexed by the parameter k, corresponding to the absolute value of the weights of positions a and e, on which the proposed algorithm requires a constant number of arithmetic operations, regardless of the weights in the game. This contrasts with what happens with both the algorithms of Brim et al. (2011) and Dorfman et al. (2019), which require a number of operations that is linear, *resp.* logarithmic, on k. To see this, we consider the algorithm SEPM Brim et al. (2011) first and show that it requires $3k + 8$ applications of its lift operator to compute a progress measure, for a total of $5k + 9$ measure updates. Indeed, the first two evaluations of lift, starting from μ_0, lead to $\mu_2 = \{\mathtt{a} \mapsto k; \mathtt{b}, \mathtt{e} \mapsto 0; \mathtt{c}, \mathtt{f}, \mathtt{g} \mapsto 2; \mathtt{d} \mapsto 1\}$, as in Picture (2), and require 5 measure lifts. Then, the algorithm iteratively increases the measures of b, g, d, f, and c by applying $3(k - 1)$ times the lift operator, for a total of $5(k - 1)$ measure lifts: $\mu_{3i} = \mu_{3i-1}[\mathtt{b} \mapsto i; \mathtt{g} \mapsto i + 3]$, $\mu_{3i+1} = \mu_{3i}[\mathtt{d}, \mathtt{f} \mapsto i + 2]$, and $\mu_{3i+2} = \mu_{3i+1}[\mathtt{c} \mapsto i + 2]$, for all $i \in [1, k - 1]$. At this point, b and f have obtained measures $k - 1$ and $k + 1$, respectively, which suffice to satisfy the progress relation along the moves $(\mathtt{f}, \mathtt{b})$ and $(\mathtt{b}, \mathtt{a})$. However, the $\boxminus$-position g does not satisfy such a relation along its unique move $(\mathtt{g}, \mathtt{c})$, since $\mu_{3k-1}(\mathtt{g}) = k + 2 < \mu_{3k-1}(\mathtt{c}) + \mathtt{g} = (k + 1) + 2 = k + 3$. Therefore, other six applications of lift are needed before g can exceed the bound $S = \mathsf{wg}(\mathtt{a}) + \mathsf{wg}(\mathtt{f}) + \mathsf{wg}(\mathtt{g}) = k + 4$. Each such lift modifies the measure of one position only, for a total of 6 lifts: $\mu_{3(k+i)} = \mu_{3(k+i)-1}[\mathtt{g} \mapsto k + 3 + i]$, $\mu_{3(k+i)+1} = \mu_{3(k+i)}[\mathtt{d} \mapsto k + 2 + i]$, and $\mu_{3(k+i)+2} = \mu_{3(k+i)+1}[\mathtt{c} \mapsto k + 2 + i]$, for $i \in \{0, 1\}$. We have then $\mu_{3k+6} = \mu_{3k+5}[\mathtt{g} \mapsto \infty]$, $\mu_{3k+7} = \mu_{3k+6}[\mathtt{d} \mapsto \infty]$, and, finally, $\mu_{3k+8} = \mu_{3k+7}[\mathtt{c} \mapsto \infty]$, which contribute with the remaining 3 lifts.

As to the algorithm of Dorfman et al. (2019), the logarithmic dependence on k is a direct consequence of the employed scaling technique. As mentioned in the introduction, at each recursive call the algorithm halves the weights of the game, taking the floor when the result is not integral, until no strictly negative weights remain in the game. This process alone requires a number of nested calls which is bounded from below by the logarithm of the maximal negative value among the weights, in the worst case. Hence, on both the families of Figures 1 and 4 it requires $\Omega(\log(k))$ arithmetic operations.

5. An Efficient Solution Algorithm

To compute sol efficiently, we provide here an imperative reformulation of the functional fixpoint algorithm $\mathsf{sol} \triangleq \mathsf{ifp}\, \varrho \,.\, \mathsf{prg}_+(\mathsf{prg}_0(\varrho))$ that improves on the complexity $\mathrm{O}(n \cdot m \cdot W \cdot \log(n {\cdot} W))$ of the SEPM algorithm of Brim et al. (2011). Recall that, by Lemma 3, each position can only be lifted at most $Z + 1 = \mathrm{O}(n \cdot \min\{W, (n - 1)!\})$ times, where $Z = \{\mathsf{wg}(\pi) \in \mathbb{N} \mid \pi \in \mathrm{SPth}(\mathrm{Ps})\} \leq S$ and

$S = \mathrm{O}(n \cdot W)$. Therefore, to obtain the desired complexity, we have to guarantee that the cost of all the computational steps be at most quasi-linear in the number of measure increases. To do so, it suffices to ensure that the algorithm explores the incoming and outgoing moves only of those positions whose measures are actually lifted. This is clearly the case for the lift operator itself, since it only explores the outgoing moves of each position in its source set. The only remaining problem is to be able to identify the positions that need to be lifted in the next iteration, by exploring only the incoming moves of the positions just lifted. Solving this problem requires some technical elementary tricks.

Algorithm 2: MPG Solver

signature $\mathsf{sol} \colon \mathrm{MPG} \to \mathrm{R}$
procedure $\mathsf{sol}(\Game)$
1 $\varrho \leftarrow (\{v \in \mathrm{Ps} \mapsto 0\}, \varnothing)$
2 $\mathsf{c} \leftarrow \{v \in \mathrm{Ps}_{\boxminus} \mapsto |\{u \in Mv(v) \,|\, \mu_{\varrho}(v) \geq \mu_{\varrho}(u) + v\}|\}$
3 $(\mathrm{N}_0, \mathrm{N}_+) \leftarrow (\{v \in \mathrm{Ps} \,|\, \mathsf{wg}(v) > 0\}, \emptyset)$
4 **while** $\mathrm{N}_0 \neq \emptyset \vee \mathrm{N}_+ \neq \emptyset$ **do**
5 $(\mathrm{N}_0, \mathrm{A}) \leftarrow \mathsf{prg}_0(\mathrm{N}_0)$
6 $\mathrm{N}_+ \leftarrow \mathrm{N}_+ \cup \mathrm{A}$
7 $(\mathrm{A}, \mathrm{N}_+) \leftarrow \mathsf{prg}_+(\mathrm{N}_+)$
8 $\mathrm{N}_0 \leftarrow \mathrm{N}_0 \cup \mathrm{A}$
9 **return** ϱ

Specifically, inspired by Brim et al. (2011), which in turn generalises the standard approach to obtain the optimal $\mathrm{O}(m)$ complexity for reachability, we employ three vectors of counters: c, d and g [1]. These vectors assign to the $\oplus$-positions the number of moves violating the progress condition, and to the $\boxminus$-positions the number of moves that satisfy it. The idea is that vectors c, d and g can be used to check in constant time whether a position needs to be lifted in the next iteration. In addition, we will also use a *binary trie* (*a.k.a. prefix tree*) data-structure T to efficiently identify the *best-escape positions*, during the computation of the operator prg_+. Finally, the set-theoretic operations of membership and emptiness can easily be computed in constant time when *indicator function* (*a.k.a.*, *characteristic function* or *bitset*) representations of the sets are available. Analogously, unions and intersections of two sets require linear time in the size of the smaller set.

Algorithm 2 reports the procedural implementation of $\mathsf{sol}(\varrho_0)$, where ϱ_0 is the smallest possible QDR, as defined at Line 1. Line 2 is used to initialise, for each $\boxminus$-position $v \in \mathrm{Ps}_{\boxminus}$, the counter $\mathsf{c}(v)$ to the number of adjacents $u \in Mv(v)$ of v that satisfy the progress inequality $\mu_{\varrho_0}(v) \geq \mu_{\varrho_0}(u) + v$, used by the algorithms that compute the operators prg_0 and prg_+.

At the beginning of each iteration $i \in \mathbb{N}$ of the while-loop at Line 4, the variable ϱ maintains the QDR ϱ_i computed by applying i times the composition $\mathsf{prg}_+ \circ \mathsf{prg}_0$, starting from ϱ_0. Moreover, the sets N_0 and N_+ contain, respectively, the positions that need to be lifted by prg_0 and the non-progress positions in ϱ_i. This last property is formalised by the following invariants that hold at

[1] After a detailed analysis of the proposed algorithm, one can observe that there is actually no need to have three distinct counters; however, for the sake of presentation, we prefer to distinguish the values by their associated intuitive semantics.

Algorithm 3: Efficient Progress Zero Operator

```
   signature prg_0 : 2^Ps → 2^Ps × 2^Ps
   procedure prg_0(N)
1      ϱ ← sup{ϱ, lift(ϱ, N, Ps)}
2      c ← c[v ∈ N ∩ Ps_⊟ ↦ |{u ∈ Mv(v) | μ_ϱ(v) ≥ μ_ϱ(u) + v}|]
3      Z ← ∅
4      foreach (v, u) ∈ Mv; u ∈ N; μ_ϱ(v) < μ_ϱ(u) + v do
5          Z ← add_0(Z, v)
6      return (Z ∩ μ_ϱ^{-1}(0), Z \ μ_ϱ^{-1}(0))
   signature add_0 : 2^Ps × Ps → 2^Ps
   procedure add_0(Z, v)
7      if v ∈ Ps_⊕ then
8          Z ← Z ∪ v
       else
9          if v ∉ N ∧ μ_ϱ(v) ≥ 0 + v then c(v) ← c(v) − 1
10         if c(v) = 0 then Z ← Z ∪ v
11     return Z
```

Line 4: $\mathrm{N}_0 = \{v \in \mathrm{Ps} \,|\, \mu_{\varrho_i}(v) = 0 \neq \mu_{\varrho_{i+1}}(v)\}$ and $\mathrm{N}_+ = \mathsf{npp}(\varrho_i)$. Observe that these invariants are trivially satisfied for $i = 0$, thanks to Line 3. Each iteration of the loop applies in sequence the operators prg_0 and prg_+, computed by Algorithms 3 and 5, respectively. Each of those algorithm returns a pair of sets that collect positions that change their status, as consequence of the application of the corresponding operator. Specifically, the first set collects the newly discovered positions with measure zero (*i.e.*, outside the current QDR) that need to be lifted in the next iteration, while the second one contains newly discovered positions that do not satisfy the progress condition in the QDR identified by the resulting measure function. After the execution of the progress procedure prg_0 at Line 5, we have that $\mathrm{N}_0 \subseteq \{v \in \mathrm{Ps} \,|\, \mu_{\varrho_{i+1}}(v) = 0 \neq \mu_{\varrho_{i+2}}(v)\}$ and $\mathrm{N}_+ \cup \mathrm{A} = \mathsf{npp}(\varrho_i^\star)$, where $\varrho_i^\star \triangleq \mathsf{prg}_0(\varrho_i)$. Thus, Line 6 collects in N_+ all the non-progress positions in $\varrho_i^\star$. Line 7 calls the progress procedure prg_+ and forces the lift of the measures of all the positions in $\Delta(\varrho_i^\star)$, as stated by Lemma 3. In addition, the verified invariants are $\mathrm{N}_0 \cup \mathrm{A} = \{v \in \mathrm{Ps} \,|\, \mu_{\varrho_{i+1}}(v) = 0 \neq \mu_{\varrho_{i+2}}(v)\}$ and $\mathrm{N}_+ = \mathsf{npp}(\varrho_{i+1})$. Finally, as required by the previously discussed invariants for the next iteration $i+1$, after Line 8 we have $\mathrm{N}_0 = \{v \in \mathrm{Ps} \,|\, \mu_{\varrho_{i+1}}(v) = 0 \neq \mu_{\varrho_{i+2}}(v)\}$.

The operators prg_0 and prg_+ are computed by Algorithms 3 and 5, while Algorithm 4 shows how to compute the operator Δ efficiently. Both the current QDR ϱ and the vector of counters c are shared among all the algorithms, including Algorithm 2, as global variables.

Algorithm 3 first computes the lift operation on all the positions contained in its input set N (Line 3) and, then, identifies the new positions that will be lifted either by the next application of prg_0, namely $\mathrm{Z} \cap \mu_\varrho^{-1}(0)$, or by the subsequent application of prg_+, namely $\mathrm{Z} \setminus \mu_\varrho^{-1}(0)$.

To do so, it first reinitialises the counter for the positions just lifted (Line 4) and, then, for each of their incoming moves (Line 5), verifies if there exists a new position whose measure needs to be increased. The case of an incoming $\oplus$-move is trivial (Lines 6-7). Therefore, let us consider the opponent player. A position $v \in \mathrm{Ps}_{\boxminus}$ needs to be lifted only if $\mu_{\varrho}(v) < \mu_{\varrho}(u) + v$, for all adjacents $u \in Mv(v)$. Therefore, we decrement the associated counter (Line 8) every time a non-progress move is identified that previously satisfied the progress condition *w.r.t.* the unlifted QDR. When the counter reaches zero, the above condition is satisfied and the considered position needs to be lifted in the next iteration (Line 9).

Proposition 6. *Algorithm 3 on input a set of positions* N *requires time*

$$\mathrm{O}\left(\sum_{v \in \mathrm{N}} (|Mv(v)| + |Mv^{-1}(v)|) \cdot \log S\right).$$

Proof. The number of iterations of the while loop is clearly bounded by the number of moves entering N, *i.e.*, $\sum_{v \in \mathrm{N}} |Mv^{-1}(v)|$. Each iteration of the loop can be computed in time bounded by $\mathrm{O}\big(\sum_{v \in \mathrm{N}} |Mv^{-1}(v)| \cdot \log S\big)$, which corresponds to the total time needed for the arithmetic operations in the condition at Line 8. The cost of Line 2 is $\mathrm{O}(|N| \cdot \log S)$, where each assignment requires time $\mathrm{O}(\log S)$. Line 3, instead, iterates over the moves exiting from the positions in N, therefore its cost is bounded by $\mathrm{O}\big(\sum_{v \in \mathrm{N}} |Mv(v)| \cdot \log S\big)$, where the factor $\log S$ is due to the arithmetic operations of the lift operator. Line 4 counts the number of moves exiting from $\boxminus$-positions in N satisfying an arithmetic condition on the measures, thus requiring again time $\mathrm{O}\big(\sum_{v \in \mathrm{N}} |Mv(v)| \cdot \log S\big)$. The total time required by the algorithm is, therefore, $\mathrm{O}\big(\sum_{v \in \mathrm{N}} (|Mv(v)| + |Mv^{-1}(v)|) \cdot \log S\big)$. □

Algorithm 4 computes the weak quasi dominion $\Delta(\varrho)$, starting from the set $\mathrm{N} = \mathsf{npp}(\varrho)$, which contains all the non-progress positions in $\mathsf{qsi}(\varrho)$. The implementation almost precisely follows the functional definition of the two operators Δ and pre, except that it keeps the whole computation cost linear in the number of incoming moves in each position contained in the resulting set.

To do so, it exploits the same tricks used in the previous procedure, by employing a counter d for the $\boxminus$-positions. Note that, d contains a copy of the values in c.

Finally, Algorithms 5 and 6 implement the procedure described in Algorithm 1. It first computes the weak quasi dominion $\Delta(\varrho)$, by calling Algorithm 4 (Line 2). After that, it identifies its escape positions and the associated forfeit, in order to identify the set of best-escape positions that need to be lifted (Line 4). To do so, we employ a binary trie T, which will contain at most S different forfeit values during the entire execution of the algorithm. Edges to the children of each node in the tree are labelled by 0, for the left child, and 1, for the right one, and each path to a leaf is labelled by the binary representation of the corresponding forfeit. Positions with the same forfeit are clustered

Algorithm 4: Efficient Quasi Dominion Operator

```
signature Δ: 2^Ps → 2^Ps
procedure Δ(N)
    Q ← ∅
    while N ≠ ∅ do
        Q ← Q ∪ N
        Z ← ∅
        foreach (v, u) ∈ Mv; u ∈ N; v ∉ (μ_ϱ^{-1}(0) ∪ Q) do
            Z ← add_Δ(Z, (v, u))
        N ← Z
    return Q
signature add_Δ: 2^Ps × Mv → 2^Ps
procedure add_Δ(Z, (v, u))
    if v ∈ Ps_⊕ then
        if σ_ϱ(v) = u then Z ← Z ∪ v
    else
        if μ_ϱ(v) ≥ μ_ϱ(u) + v then c(v) ← c(v) − 1
        if c(v) = 0 then Z ← Z ∪ v
    return Z
```

together and associated with that value at the corresponding leaf in the trie. Inserting a position with a given forfeit in the trie requires traversing a path in the tree following the edges labelled by the corresponding bit in the binary representation of the forfeit. Therefore, the time required for insertion is linear in the length of this binary representation, namely $\mathrm{O}(\log S)$. Extraction of the set of positions with the minimal forfeit in T simply requires accessing the leaf reached by following the leftmost path in the tree and can be done in time $\mathrm{O}(\log S)$ as well.

The while-loop at Line 6 simulates the while-loop at Line 2 of Algorithm 1, where instructions at Lines 7-10 here precisely correspond to those at Lines 3-5 in the original algorithm. Note that the trie T is allowed to contain multiple copies of a position v in different nodes of the trie, one for each move from that position in the worst case. Hence, there can be at most $|Mv(v)|$ copies of the same position in T. For this reason, Line 8 discards from E copies of positions already processed in some previous iteration of the loop that have already been removed from Q. After the measure update of the best-escape positions in E, the associated counters in c are reinitialised (line 11). At this point, an analysis on the incoming moves of E takes place (Line 12). For all moves $(v, u) \in Mv$ with $u \in \mathrm{E}$ and $v \notin \mathrm{Q}$, the algorithm performs, at Lines 18-23, almost exactly the same operations done by Algorithm 3 at Lines 6-9. The only difference here is that ⊟-positions can only be forced to lift their measure if they are not yet contained in the quasi dominion $\mathsf{qsi}(\varrho)$. The case $v \in \mathrm{Q}$, instead, identifies a possible discovering of a new escape of the remaining weak quasi dominion

Algorithm 5: Efficient Progress Plus Operator

signature $\mathsf{prg}_+ : 2^{\mathrm{Ps}} \to 2^{\mathrm{Ps}} \times 2^{\mathrm{Ps}}$

procedure $\mathsf{prg}_+(\mathrm{N})$

1 $\mathrm{Q} \leftarrow \Delta(\mathrm{N})$

2 $\widehat{\mu} \leftarrow \{v \in \mathrm{Q} \mapsto \mu_\varrho(v)\}$

3 $\mathrm{T} \leftarrow \{(v, \mathsf{bef}(\mu_\varrho, \mathrm{Q}, v)) \in \mathsf{esc}(\varrho, \mathrm{Q}) \times \mathbb{N}\}$

4 $\mathsf{c} \leftarrow \{v \in \mathrm{Q} \cap \mathrm{Ps}_\oplus \mapsto$
$|\{u \in Mv(v) \cap \mathrm{Q} \mid \sigma_\varrho(v) = u \vee \mu_\varrho(v) < \mu_\varrho(u) + v\}|\}$

5 $\mathrm{Z} \leftarrow \emptyset$

6 **while** $\mathrm{T} \neq \emptyset$ **do**

7 $\quad (\mathrm{E}, \mathrm{T}) \leftarrow \mathsf{extmin}(\mathrm{T})$

8 $\quad \mathrm{E} = \mathrm{E} \cap \mathrm{Q}$

9 $\quad \varrho \leftarrow \mathsf{lift}(\varrho, \mathrm{E}, \overline{\mathrm{Q}})$

10 $\quad \mathrm{Q} \leftarrow \mathrm{Q} \setminus \mathrm{E}$

11 $\quad \mathsf{c} \leftarrow \mathsf{c}[v \in \mathrm{E} \cap \mathrm{Ps}_\boxminus \mapsto |\{u \in Mv(v) \mid \mu_\varrho(v) \geq \mu_\varrho(u) + v\}|]$

12 $\quad$ **foreach** $(v, u) \in Mv;\ u \in \mathrm{E}$ **do**

13 $\quad\quad$ **if** $v \in \mathrm{Q}$ **then**

14 $\quad\quad\quad$ **if** $v \in \mathrm{Ps}_\boxminus$ **then**

15 $\quad\quad\quad\quad \mathrm{T} \leftarrow \mathrm{T} \cup (v, \mu_\varrho(u) + v - \mu_\varrho(v))$

$\quad\quad\quad$ **else**

16 $\quad\quad\quad\quad$ **if** $\sigma_\varrho(v) = u \vee \mu_\varrho(v) < \widehat{\mu}(u) + v$ **then** $\mathsf{c}(v) \leftarrow \mathsf{c}(v) - 1$

17 $\quad\quad\quad\quad$ **if** $\mathsf{c}(v) = 0$ **then** $\mathrm{T} \leftarrow \mathrm{T} \cup (v, \mathsf{bef}(\mu_\varrho, \mathrm{Q}, v))$

18 $\quad\quad$ **else if** $\mu_\varrho(v) < \mu_\varrho(u) + v$ **then**

19 $\quad\quad\quad \mathrm{Z} \leftarrow \mathsf{add}_+(\mathrm{Z}, (v, u))$

20 $\varrho \leftarrow \mathsf{win}(\varrho, \mathrm{Q})$

21 **foreach** $(v, u) \in Mv;\ u \in \mathrm{Q}$ **do**

22 $\quad \mathrm{Z} \leftarrow \mathsf{add}_+(\mathrm{Z}, (v, u))$

23 **return** $(\mathrm{Z} \cap \mu_\varrho^{-1}(0), \mathrm{Z} \setminus \mu_\varrho^{-1}(0))$

(Line 13). If $v \in \mathrm{Ps}_\boxminus$, this is obviously an escape from Q, thus, it needs to be added to the trie T paired with the associated best-escape forfeit computed along the move (v, u) (Lines 14-15).

The case $v \in \mathrm{Ps}_\oplus$ is more complicated, since a $\oplus$-position is an escape *iff* its current strategy exits from Q and it has no move within Q that allows an increase of its measure. To do this check, once again, we employ the counter trick, where this time we associate with a $\oplus$-position in $\Delta(\varrho)$ the number of moves that satisfy the above property (Line 5). If the move (v, u) satisfies the property *w.r.t.* the unlifted QDR (*i.e.*, before the lift of u occurs), then the corresponding counter $\mathsf{g}(v)$ is decreased (Line 16). When the counter reaches value 0, the position is necessarily an escape, so, it is added to the trie paired with its best possible forfeit (Line 17). Line 24 calls the win function in order

Algorithm 6: Efficient Progress Plus Operator

signature $\mathsf{add}_{+} : 2^{\mathrm{Ps}} \times Mv \rightarrow 2^{\mathrm{Ps}}$

procedure $\mathsf{add}_{+}(\mathrm{Z}, (v, u))$

 if $v \in \mathrm{Ps}_{\oplus}$ **then**

  $\mathrm{Z} \leftarrow \mathrm{Z} \cup v$

 else if $\mu_{\varrho}(v) = 0$ **then**

  if $\widehat{\mu}(u) + v \leq 0$ **then** $\mathsf{c}(v) \leftarrow \mathsf{c}(v) - 1$

  if $\mathsf{c}(v) = 0$ **then** $\mathrm{Z} \leftarrow \mathrm{Z} \cup v$

 return Z

to identify a possible new $\oplus$-dominion. Finally, Lines 25-30 update both the set of positions Z to be lifted in the next iteration and the counter c, by executing exactly the same instructions as those at Lines 19-23 on the moves that reach the dominion Q.

Proposition 7. *Algorithm 5 on input a set of positions* N *requires time*

$$\mathrm{O}\left(\sum_{v \in \Delta(\mathrm{N})} (|Mv(v)| + |Mv^{-1}(v)|) \cdot \log S\right).$$

Proof. First observe that the set $\Delta(\mathrm{N})$ is assigned to Q at Line 2. Clearly, Lines 1-3 and 5 can be computed in time $\mathrm{O}(|Mv(\mathrm{Q})| \cdot \log S)$. As to Line 4, assuming a indicator function representation of Q and $\mathrm{Ps}_{\oplus}$, that allows for set-membership tests in constant time, checking if a position v is an escape position and the computation of its best-escape forfeit can be done in time $\mathrm{O}(|Mv(v)| \cdot \log S)$. Finally, the insertion in T requires time $\mathrm{O}(\log S)$, assuming a trie representation of T. Hence, the total time upper bound for Line 4 is $\mathrm{O}(|Mv(\mathrm{Q})| \cdot \log S)$ as well.

Let us consider the loop at Lines 6-23. First, observe that, during the loop, at most $Mv(\mathrm{Q})$ distinct elements can be contained in T, as there can be at most $Mv(\mathrm{Q})$ different forfeit, one for each move outgoing from a position in Q. Each extraction of the minimal element from T (Line 7) requires time $\mathrm{O}(\log S)$, and $\mathrm{O}(|Mv(\mathrm{Q})| \cdot \log S)$ overall. Line 8 discards all the copies of positions that have already been processed (*i.e.*, escaped from Q) in some previous iteration. Therefore, the successive operations in the loop are executed at most once for each position in Q. In particular, the cost of the operations in Lines 9-11, for each position $v \in \mathrm{Q}$, is bounded by $\mathrm{O}(|Mv(v)| \cdot \log S)$.

The inner loop at Lines 12-23, is executed at most once for each move entering a position in Q. The cost of every operation in the body of this loop is clearly bounded by the cost of Line 15, which is $\mathrm{O}(\log S)$. Hence, the overall cost of the loop is $\mathrm{O}\big(|Mv^{-1}(v)| \cdot \log S\big)$.

The computation of the operator win at Line 24, which sets the residual positions in Q as winning for player $\oplus$ and computes the corresponding winning strategy by inspecting the outgoing moves, clearly requires time

$O(|Mv(Q)| \cdot \log S)$. Finally, the loop at Lines 25-31 examines the moves entering Q. The most expansive operation for each such move is the tests at Line 29, which requires time $O(\log S)$, leading to a total time $O\big(|Mv^{-1}(Q)| \cdot \log S\big)$. The thesis, then, follows. □

We can now establish the following result.

Theorem 7 (Complexity)**.** QDPM *requires time*

$$O(n \cdot m \cdot \mathsf{min}\{W, (n-1)!\} \cdot \log(n \cdot W))$$

to solve an MPG *with* n *positions,* m *moves, and maximal positive weight* W*.*

Proof. We know from Proposition 6 and Proposition 7 that the procedures $\mathsf{prg}_0(\varrho, \mathsf{c}, N_0)$ and $\mathsf{prg}_+(\varrho, \mathsf{c}, N_+)$ require time

$$O\left(\sum_{v \in N_0} (|Mv(v)| + |Mv^{-1}(v)|) \cdot \log S\right)$$

and

$$O\left(\sum_{v \in \Delta(\varrho)} (|Mv(v)| + |Mv^{-1}(v)|) \cdot \log S\right),$$

respectively, where $\mathsf{npp}(\varrho) = N_+$. In particular, the factor $\log S$ is due to all the arithmetic operations required to compute the stretch of the measures. Since during the entire execution of the algorithm each position $v \in Ps$ can appear at most once in some N_0 and at most $Z = O(n \cdot \mathsf{min}\{W, (n-1)!\})$ times in some $\Delta(\varrho)$, it follows that the total cost of Algorithm 2 is

$$O\left(n + (Z+1) \cdot \sum_{v \in Ps} (|Mv(v)| + |Mv^{-1}(v)|) \cdot \log S\right) = O(n + Z \cdot m \cdot \log S) =$$
$$O(n \cdot m \cdot \mathsf{min}\{W, (n-1)!\} \cdot \log(n \cdot W)),$$

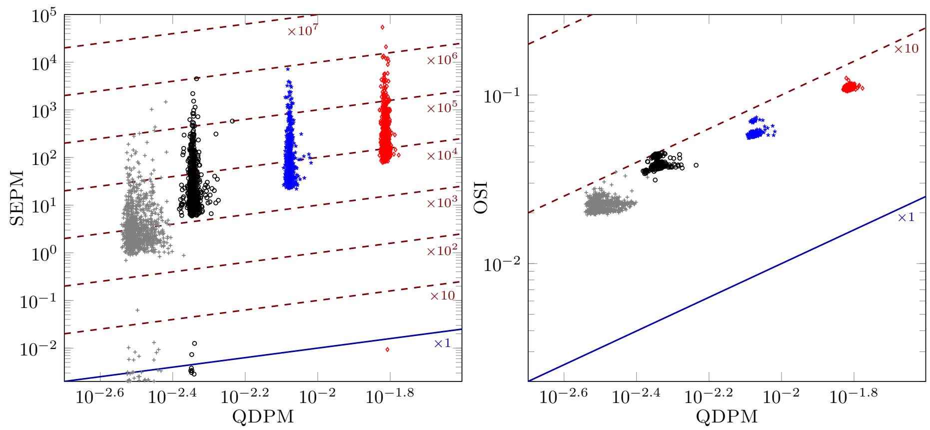
where the term n in the sum is due to the initialisation operations at Lines 1-3. □

6. Experimental Evaluation

In order to assess the effectiveness of the proposed approach we implemented the novel algorithm QDPM, the Optimal Strategy Improvement algorithm (OSI by Schewe (2008)), the Small Energy Progress Measure algorithm (SEPM by Brim et al. (2011)) and DKZ, the algorithm proposed by Dorfman et al. (2019). [2] These are the most efficient known solutions to the problem

[2] The algorithm proposed by Dorfman et al. (2019) presents few inaccuracies that make it incorrect. We implemented here the fixed version described in the technical report by Austin and Dell'Erba (2023).

and the more closely related ones to QDPM. All the algorithms have been implemented in C++ within the framework OINK from van Dijk (2018), which was originally developed as a tool to compare parity game solvers. However, extending the OINK to deal with MPGs is not difficult. The form of the arenas of the two types of games essentially coincide, the only relevant difference being that MPGs allow negative numbers to label game positions. We ran the solvers against randomly generated MPGs of various sizes.[3]



(a) Comparison of QDPM and SEPM.

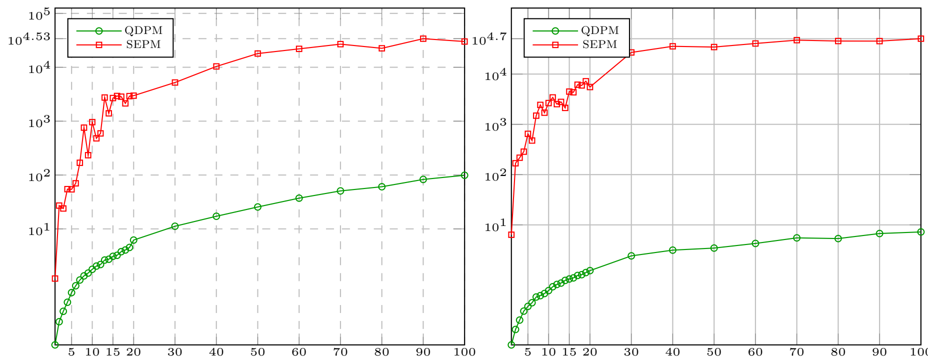
(b) Comparison of QDPM and OSI.

Figure 5: Comparisons on random games with 5000 positions.

In the first benchmark set reported in Figures 5a and 5b we compare the solution time, expressed in seconds, of QDPM against SEPM and OSI respectively, on 4000 games, each with 5000 positions and randomly assigned weights in the range $[-15000, 15000]$. The scale of both axes is logarithmic. The experiments are divided in 4 clusters, each containing 1000 games. The benchmarks in different clusters differ in the maximal number m of outgoing moves per position, with $m \in \{10\ \text{(grey)}, 20\ \text{(black)}, 40\ \text{(blue)}, 80\ \text{(red)}\}$. These experiments clearly show that QDPM substantially outperforms SEPM. Most often, the gap between the two algorithms is between two and three orders of magnitude, as indicated by the dashed diagonal lines. It also shows that SEPM is particularly sensitive to the density of the underlying graph, as its performance degrades significantly as the number of moves increases. The maximal solution time was 8940 sec. for SEPM and 0.5 sec. for QDPM. The gap with OSI, instead, remains essentially constant at about one order of magnitude.

Figure 6a, instead, compares QDPM and SEPM fixing the maximal out-degree of the underlying graphs to 2, while in Figure 6b the maximal out-degree is 40. In both figures, the number of positions goes from 10^3 to 10^5 along the

[3]The experiments were carried out on a 64-bit 3.9GHz quad-core machine, with INTEL i5-6600K processor and 8GB of RAM, running UBUNTU 18.04.

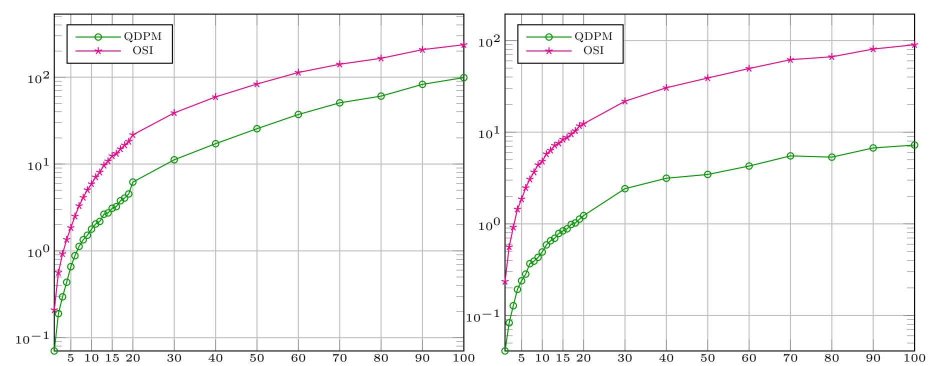


(a) Games with two moves per position.

(b) Games with 40 moves per position.

Figure 6: Total solution times in seconds of SEPM and QDPM on 2800 random games.

x-axis, while the y-axis represents the solution time in seconds. The pictures display the performance results on 2800 games. Each point shows the total time to solve 100 randomly generated games with that given number of positions, which increases by 1000 up to size $2 \cdot 10^3$ and by 10000, thereafter. In both pictures the scale is logarithmic. For the experiments in Figure 6a we had to set a timeout for SEPM to 45 minutes per game, which was hit most of the times on the bigger instances.



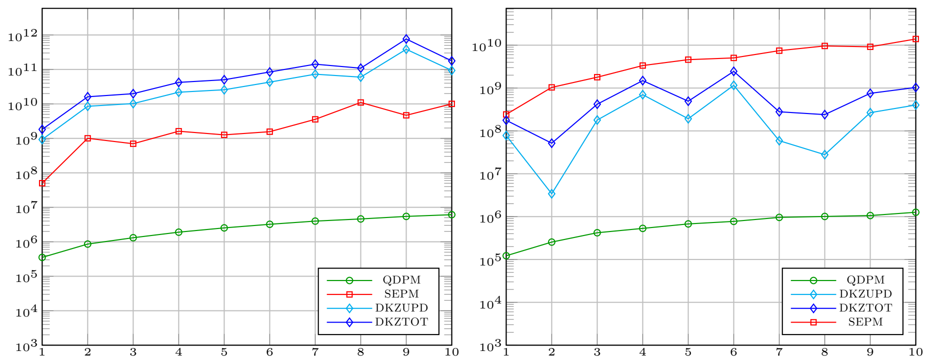
(a) Games with two moves per position.

(b) Games with 40 moves per position.

Figure 7: Total solution times in seconds of OSI and QDPM on 2800 random games.

Once again, the QDPM significantly outperforms SEPM on both kinds of benchmarks, with a gap of more than an order of magnitude on the first ones, and a gap of more than three orders of magnitude on the second ones. The results also confirm that the performance gap grows considerably as the number of moves per position increases.

On the same benchmarks, Figures 7a and 7b compare, instead, QDPM and OSI. Also in this case QDPM outperforms OSI. However, while for games with two moves per position the gap between the two algorithms is rather small, games with a higher number of moves per positions proved to be much easier to solve for QDPM, and, on such games the gap with QDPM significantly increases at about one order of magnitude.



(a) Games with two moves per position.

(b) Games with 40 moves per position.

Figure 8: Arithmetic operations of DKZ, SEPM and QDPM on 2000 random games.

Figures 8a and 8b compare QDPM with both SEPM and DKZ, this time reporting the number of arithmetic operations, corresponding to updates of the measures of the game positions, required to obtain the solution. This is a particularly interesting performance measure as it does not depend on the concrete implementation but only on the actual solution techniques. More specifically, the charts report the overall number of operations that each solver requires to solve the 100 games tested for each size, for a total of 2000 randomly generated games. The size ranges from 10^3 to 10^4 positions with 2 (on the left) and 40 (on the right) outgoing moves per positions. The benchmarks are composed of the same games used in the previous charts, except that we limit the maximal size to 10^4. Also in this case, the results on the y-axis are depicted on a logarithmic scale. For DKZ we report the number of measure update operations (DKZUPD), as well as the total number of of arithmetic operations (DKZTOT) that also includes the weight halving operations performed by the recursive calls of the algorithm.

Depending on the class of benchmarks, the experiments show that the behaviour of DKZ is much closer to SEPM's in terms of number of operations and is typically worse on games with small number of moves, but slightly better on games with higher number of moves, which are usually simpler for all the solver. Clearly, the number of operations directly correlates with solution times, hence the higher the number of operation is, the higher the solution time becomes. However, it is worth noting that even when the number of operations required by SEPM is higher, the solver is usually about as fast as DKZ in practice, as it does not incur in the computational overhead that the latter requires in order to guarantee its theoretical upper bound on computation time. On the most difficult games with two moves per position, in particular, the gap in solution time between the two algorithms can go up to three orders of magnitude in favour of SEPM. These results seem to suggest that while interesting from a theoretical standpoint, the approach followed by DKZ is not likely to translate into meaningful performance uplift compared to SEPM, let alone QDPM or OSI, in most application scenarios.

Benchmark	Positions	Moves	SEPM	DKZ	QDPM	OSI
Elevator 1	144	234	0.04	0.0001	**0.0002**	**0.0002**
Elevator 2	564	950	8.80	0.14	0.0007	**0.0006**
Elevator 3	2688	4544	4675.71	10.95	0.0062	**0.0028**
Elevator 4	15683	26354	×	⊥	0.0528	**0.0379**
Lang. Incl. 1	170	1094	3.18	0.056	**0.0002**	0.0003
Lang. Incl. 2	304	1222	16.76	0.7	**0.0002**	0.0004
Lang. Incl. 3	428	878	20.25	40.59	**0.0002**	0.0004
Lang. Incl. 4	628	1538	135.51	274.89	**0.0003**	0.0006
Lang. Incl. 5	509	2126	148.37	⊥	**0.0003**	0.0005
Lang. Incl. 6	835	2914	834.90	30.19	**0.0005**	0.0010
Lang. Incl. 7	1658	4544	2277.87	⊥	**0.0002**	0.0009
Lang. Incl. 8	14578	17278	×	⊥	**0.0008**	0.0064
Lang. Incl. 9	25838	29438	×	⊥	**0.0015**	0.0137
Lang. Incl. 10	29874	34956	×	⊥	**0.0063**	0.0424

Table 1: Experiments on concrete verification problems.

We are not aware of actual concrete benchmarks for MPGs. However, exploiting the standard encoding of parity games into mean-payoff games of Jurdziński (1998), we can compare the behaviour of the algorithms on concrete verification problems encoded as parity games. For completeness, Table 1 reports some experiments on such problems. The table reports the execution times, expressed in seconds, required by the algorithms to solve instances of two classic verification problems: the Elevator Verification and the Language Inclusion problems. These two benchmarks are included in the PGSolver toolkit, see Friedmann and Lange (2009), and are often used as benchmarks for parity games solvers. The first benchmark is a *verification under fairness* constraints of a simple model of an elevator, while the second one encodes the *language inclusion* problem between a non-deterministic Büchi automaton and a deterministic one. In the table, the symbol × indicates that the solver hit the time-out, set at 5000 seconds, before finding a solution. The symbol ⊥, instead, is used to indicate that the solver could not solve the game due to an overflow error. This happens only to DKZ, which needs to perform a preprocessing of the game to add a fictitious position with incoming moves with weight $-2nW$, where n is the number of postisions and W the maximum weight value. For large enough instances of these concrete games, the preprocessing computes weight values that are too big to fit into a long integer, causing an overflow error.

The results on various instances of those problems confirm that QDPM significantly outperforms both SEPM and DKZ, while trading blows with OSI. Note also that the translation into MPGs, which encodes priorities as weights whose absolute value is exponential in the values of the priorities, leads to games with weights of high magnitude. Hence, the results in Table 1 provide further evidence that QDPM is far less dependent on the absolute value of the weights. They also show that QDPM can be very effective for the solution of real-world qualitative verification problems. It is worth noting, though, that the translation from parity to MPGs gives rise to weights that are exponentially distant from each other, see Jurdziński (1998). As a consequence, the resulting benchmarks are not necessarily representative of MPGs, being a very restricted subclass. Nonetheless, they provide evidence of the applicability of the approach in practical scenarios.

7. Discussion

We proposed a novel solution algorithm for the decision problem of MPGs that integrates progress measures and quasi dominions. We argue that the integration of these two concepts may offer significant speed up in convergence to the solution, at no additional computational cost. This is evidenced by the existence of a family of games on which the combined approach can perform arbitrarily better than a classic progress measure based solution. Experimental results also show that the introduction of quasi dominions can often reduce solution times up to three order of magnitude, suggesting that the approach may be very effective in practical applications as well. We believe that the integration approach we devised is general enough to be applied to other types of games. In particular, the application of quasi dominions in conjunction with progress measure based approaches, such as those of Jurdziński and Lazic (2017) and Fearnley et al. (2017), may lead to practically efficient quasi polynomial algorithms for parity games and their quantitative extensions.

Acknowledgments

Partially supported by the GNCS 2020 project "Ragionamento Strategico e Sintesi Automatica di Sistemi Multi-Agente", GNCS 2022 project "Elaborazione del Linguaggio Naturale e Logica Temporale per la Formalizzazione di Testi", and GNCS 2023 project "Analisi Simbolica e Numerica di Sistemi Ciberfisici". The second author has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 101032464.

We also thank Peter Austin from the University of Liverpool for providing an implementation of the DKZ algorithm.

Appendix A. Proofs

Theorem 1 (Progress Measure)**.** $\|\mu\|_{\boxminus} \subseteq \mathrm{Wn}_{\boxminus}$*, for all progress measures* μ*.*

Proof. Consider a $\boxminus$-strategy $\sigma_{\boxminus} \in \mathrm{Str}_{\boxminus}$ for which all measures $\mu(v)$ of positions $v \in \|\mu\|_{\boxminus} \cap \mathrm{Ps}_{\boxminus}$ are a progress at v *w.r.t.* the measures $\mu(\sigma_{\boxminus}(v))$ of their adjacents $\sigma_{\boxminus}(v)$, formally, $\mu(\sigma_{\boxminus}(v)) + v \leq \mu(v)$. The existence of such a strategy is ensured by the fact that μ is a progress measure. Indeed, by Condition 2 of Definition 2, there necessarily exists a adjacent $u^{\star} \in Mv(v)$ of v such that $\mu(u^{\star}) + v \leq \mu(v)$. Now, it can be shown that $\sigma_{\boxminus}$ is a winning strategy for player $\boxminus$ from all the positions in $\|\mu\|_{\boxminus}$, which implies that $\|\mu\|_{\boxminus} \subseteq \mathrm{Wn}_{\boxminus}$. To do this, let us consider a $\oplus$-strategy $\sigma_{\oplus} \in \mathrm{Str}_{\oplus}$ and the associated play $\pi = \mathsf{play}((\sigma_{\oplus}, \sigma_{\boxminus}), v)$ starting at a position $v \in \|\mu\|_{\boxminus}$. Assume, by contradiction, that π is won by player $\oplus$. Since the game $\eth$ is finite, π must contain a finite simple cycle, and so a finite simple path, with strictly positive total weight sum. In other words, there exist two natural numbers $h \in \mathbb{N}$ and $k \in \mathbb{N}_{+}$ such that $(\pi)_h = (\pi)_{h+k}$ and $\mathsf{wg}(\rho) = \sum_{i=h}^{h+k-1} \mathsf{wg}((\pi)_i) > 0$, where $\rho \triangleq ((\pi)_{\geq h})_{<h+k}$ is the simple path named above. Now, recall that, $((\pi)_i, (\pi)_{i+1}) \in Mv$, for all indexes $i \in \mathbb{N}$. Thus, by both conditions of Definition 2, and the notion of play, we have that

$$\mu((\pi)_{i+1}) + (\pi)_i \leq \mu((\pi)_i).$$

Via a trivial induction, it is immediate to see that $\mu((\pi)_i) \leq S$, where

$$S \triangleq \sum \{\mathsf{wg}(v) \in \mathbb{N} \mid v \in \mathrm{Ps} \wedge \mathsf{wg}(v) > 0\} < \infty$$

for all $i \in \mathbb{N}$, since $\mu((\pi)_0) = \mu(v) \neq \infty$, being $v \in \|\mu\|_{\boxminus}$. As a consequence, due to the definition of the measure stretch operator, it holds that

$$\mu((\pi)_{i+1}) + \mathsf{wg}((\pi)_i) \leq \mu((\pi)_i) \leq S.$$

Hence, by summing together all the inequalities having indexes $i \in \mathbb{N}$ with $h \leq i < h + k$, we obtain

$$\sum_{i=h+1}^{h+k} \mathsf{pf}_{\mu}((\pi)_i) + \sum_{i=h}^{h+k-1} \mathsf{wg}((\pi)_i) \leq \sum_{i=h}^{h+k-1} \mathsf{pf}_{\mu}((\pi)_i) < \infty,$$

which simplifies in $\mathsf{wg}(\rho) = \sum_{i=h}^{h+k-1} \mathsf{wg}((\pi)_i) \leq 0$, since $\mathsf{pf}_{\mu}((\pi)_{h+k}) = \mathsf{pf}_{\mu}((\pi)_h)$. However, this contradicts the above assumption $\mathsf{wg}(\rho) > 0$. Therefore, $\sigma_{\boxminus}$ is a winning strategy for player $\boxminus$ on $\|\mu\|_{\boxminus}$ as required by the theorem statement. □

Lemma 1. μ_{ϱ} *is a progress measure over* $\overline{\mathsf{qsi}(\varrho)}$*, for all fixpoints* ϱ *of* prg_0*.*

Proof. By definition of the progress operator prg_0, we have that

$$\varrho = \mathsf{prg}_0(\varrho) = \mathsf{sup}\{\varrho, \mathsf{lift}(\varrho, \overline{\mathsf{qsi}(\varrho)}, \mathrm{Ps})\}$$

from which we derive $\varrho^{\star} \triangleq \mathsf{lift}(\varrho, \overline{\mathsf{qsi}(\varrho)}, \mathrm{Ps}) \sqsubseteq \varrho$. Now, consider an arbitrary position $v \in \overline{\mathsf{qsi}(\varrho)}$ and observe that $\mu_{\varrho^{\star}}(v) \leq \mu_{\varrho}(v)$, due to Item 2 of Definition 4. At this point, the proof proceeds by a case analysis on the owner of the position v itself.

- $[v \in \mathrm{Ps}_{\oplus}]$. By definition of the lift operator, we have that

$$\mu_{\varrho}(u) + v \leq \mathsf{max}\,\{\mu_{\varrho}(u) + v \,|\, u \in Mv(v)\} = \mu_{\varrho^{\star}}(v)$$

  for all adjacents $u \in Mv(v)$ of v. Thus, $\mu_{\varrho}(u)+v \leq \mu_{\varrho^{\star}}(v) \leq \mu_{\varrho}(v)$, thanks to the above observation. Consequently, Condition 1 of Definition 2 is satisfied on $\overline{\mathsf{qsi}(\varrho)}$.

- $[v \in \mathrm{Ps}_{\boxminus}]$. Again by definition of the lift operator, we have that

$$\mu_{\varrho}(u) + v \leq \mathsf{min}\,\{\mu_{\varrho}(u) + v \,|\, u \in Mv(v)\} = \mu_{\varrho^{\star}}(v)$$

  for some adjacent $u \in Mv(v)$ of v. Due to the above observation, it holds that $\mu_{\varrho}(u) + v \leq \mu_{\varrho^{\star}}(v) \leq \mu_{\varrho}(v)$. Hence, Condition 2 of Definition 2 is satisfied on $\overline{\mathsf{qsi}(\varrho)}$ as well.

□

Lemma 2. *μ_{ϱ} is a progress measure over $\mathsf{qsi}(\varrho)$, for all fixpoints ϱ of prg_{+}.*

Proof. Let us consider the infinite monotone sequence of position sets $\mathrm{Q}_0 \supseteq \mathrm{Q}_1 \supseteq \ldots$ defined as follows: $\mathrm{Q}_0 \triangleq \Delta(\varrho)$; $\mathrm{Q}_{i+1} \triangleq \mathrm{Q}_i \setminus \mathrm{E}_i$, where $\mathrm{E}_i \triangleq \mathsf{bep}(\varrho, \mathrm{Q}_i)$, for all $i \in \mathbb{N}$. Since $|\mathrm{Q}_0| < \infty$, there necessarily exists an index $k \in \mathbb{N}$ such that $\mathrm{Q}_{k+1} = \mathrm{Q}_k$. By definition of the progress operator prg_{+} and the equality $\varrho = \mathsf{prg}_{+}(\varrho)$, we have that $\varrho = \mathsf{lift}(\varrho, \mathrm{E}_i, \overline{\mathrm{Q}_i})$, for all $i \in [0, k)$, and $\varrho = \mathsf{win}(\varrho, \mathrm{Q}_k)$. Now, consider an arbitrary position $v \in \mathsf{qsi}(\varrho)$. If $v \notin \Delta(\varrho)$, due to the definition of the set $\Delta(\varrho)$, the position v satisfies by definition of the appropriate condition of Definition 2 on $\mathsf{qsi}(\varrho)$. Therefore, let us assume $v \in \Delta(\varrho)$. Then, it is obvious that either $v \in \mathrm{Q}_k$ or there is a unique index $i \in [0, k)$ such that $v \in \mathrm{Q}_i \setminus \mathrm{Q}_{i+1}$, *i.e.*, $v \in \mathrm{E}_i$. In the first case, we have $\mu_{\varrho}(v) = \infty$, due to the definition of the function win. Therefore, v is a progress position. In the other case, the proof proceeds by a case analysis on the owner of the position v itself.

- $[v \in \mathrm{Ps}_{\oplus}]$. First observe that $\mathsf{bep}(\varrho, \mathrm{Q}_i) \subseteq \mathsf{esc}(\varrho, \mathrm{Q}_i)$. Thus, due to the definition of the function esc, we have that $\mu_{\varrho}(u) + v \leq \mu_{\varrho}(v)$, for all positions $u \in Mv(v) \cap \mathrm{Q}_i$. Now, by the definition of the lift operator, we have that $\mu_{\varrho}(u) + v \leq \mathsf{max}\left\{\mu_{\varrho}(u) + v \,\middle|\, u \in Mv(v) \cap \overline{\mathrm{Q}_i}\right\} = \mu_{\varrho}(v)$, for all adjacents $u \in Mv(v) \cap \overline{\mathrm{Q}_i}$ of v. Consequently, $\mu_{\varrho}(u) + v \leq \mu_{\varrho}(v)$, for all positions $u \in Mv(v)$, as required by Condition 1 of Definition 2 on $\mathsf{qsi}(\varrho)$.

- $[v \in \mathrm{Ps}_{\boxminus}]$. Again by definition of the lift operator, we have that

$$\mu_{\varrho}(u) + v \leq \mathsf{min}\left\{\mu_{\varrho}(u) + v \,\middle|\, u \in Mv(v) \cap \overline{\mathrm{Q}_i}\right\} = \mu_{\varrho}(v)$$

  for some adjacent $u \in Mv(v) \cap \overline{\mathrm{Q}_i} \subseteq Mv(v)$ of v. Hence, Condition 2 of Definition 2 is satisfied on $\mathsf{qsi}(\varrho)$ as well.

□

Theorem 2. *The operators prg_0 and prg_{+} are total inflationary functions.*

Proof. The proof proceeds by showing that, for each $\varrho \in \mathrm{R}$, the elements $\mathsf{prg}_0(\varrho)$ and $\mathsf{prg}_+(\varrho)$ are QDR too. We also prove that $\varrho \sqsubseteq \mathsf{prg}_0(\varrho)$ and $\varrho \sqsubseteq \mathsf{prg}_+(\varrho)$. The two operators are analysed separately.

- $[\mathsf{prg}_0]$. Let $\varrho^\star \triangleq \mathsf{prg}_0(\varrho) = \sup\{\varrho, \mathsf{lift}(\varrho, \overline{\mathsf{qsi}(\varrho)}, \mathrm{Ps})\} \sqsupseteq \varrho$. It is obvious, so, that prg_0 is inflationary. Consider now a position $v \in \mathsf{qsi}(\varrho^\star)$. Recall that $\mu_{\varrho^\star}(v) > 0$. If $v \in \mathsf{qsi}(\varrho)$, by definition of the lift operator, it holds that $\mu_{\varrho^\star}(v) = \mu_\varrho(v)$ and $\sigma_{\varrho^\star}(v) = \sigma_\varrho(v)$, thus the appropriate condition between Conditions 1c and 1d of Definition 4 is verified, since $\varrho \in \mathrm{R}$. Thus, assume $v \in \overline{\mathsf{qsi}(\varrho)}$. If $v \in \mathrm{Ps}_\oplus$, we have that $\mu_{\varrho^\star}(v) = \max\{\mu_\varrho(u) + v \mid u \in Mv(v)\} = \mu_\varrho(\sigma_{\varrho^\star}(v)) + v = \mu_{\varrho^\star}(\sigma_{\varrho^\star}(v)) + v$, since $\sigma_{\varrho^\star}(v) \in \mathsf{qsi}(\varrho)$. As a consequence, Condition 1c is satisfied. If $v \in \mathrm{Ps}_\boxminus$, instead, we have that $\mu_{\varrho^\star}(v) = \min\{\mu_\varrho(u) + v \mid u \in Mv(v)\}$, which implies $\mu_{\varrho^\star}(v) \leq \mu_\varrho(u) + v = \mu_{\varrho^\star}(u) + v$, for all adjacents $u \in Mv(v)$, as required by Condition 1d. To complete the proof that prg_0 is a total function from R to itself, we need to show that $\varrho^\star$ satisfies Conditions 1b and 1a too. It is immediate to see that $\|\mu_\varrho\|_\oplus \subseteq \|\mu_{\varrho^\star}\|_\oplus$. Since ϱ is a QDR, $\|\mu_\varrho\|_\oplus$ is a $\oplus$-dominion. Moreover, for all positions $v \in \|\mu_{\varrho^\star}\|_\oplus \setminus \|\mu_\varrho\|_\oplus$, it holds that $\sigma_{\varrho^\star}(v) \in \|\mu_\varrho\|_\oplus$, if $v \in \mathrm{Ps}_\oplus$, and $Mv(v) \subseteq \|\mu_\varrho\|_\oplus$, otherwise. Therefore, $\|\mu_{\varrho^\star}\|_\oplus$ is necessarily a $\oplus$-dominion, so Condition 1b is verified. Finally, let us focus on Condition 1a and consider a $(\sigma_{\varrho^\star}, v)$-play $v\pi$. If, on the one hand, π is infinite and does not meet v, thanks to Proposition 1, we have $\mathsf{wg}(\pi) = \infty$, thus $\mathsf{wg}(v\pi) = \infty$ and, so, $\mathsf{wg}(v\pi) > 0$. If π is finite, instead, it holds that $\mathsf{lst}(\pi) \in \mathsf{esc}(\varrho, \mathsf{qsi}(\varrho))$ and, so, $\mu_{\varrho^\star}(\mathsf{lst}(\pi)) = \mathsf{wg}(\mathsf{lst}(\pi))$, due to Proposition 4. Now, by Proposition 2, we have that $\mu_\varrho(\mathsf{fst}(\pi)) \leq \mu_\varrho(\mathsf{lst}(\pi)) + \mathsf{wg}(\pi_{<\ell-1}) = \mathsf{wg}(\mathsf{lst}(\pi)) + \mathsf{wg}(\pi_{<\ell-1}) = \mathsf{wg}(\pi)$, where $\ell \in \mathbb{N}$ is the length of π Moreover, $0 < \mu_\varrho(v) \leq \mu_\varrho(\mathsf{fst}(\pi)) + v = \mu_\varrho(\mathsf{fst}(\pi)) + \mathsf{wg}(v)$, thanks to the previously proved Conditions 1c and 1d. Hence, $0 < \mu_\varrho(v) \leq \mu_\varrho(\mathsf{fst}(\pi)) + \mathsf{wg}(v) \leq \mathsf{wg}(v) + \mathsf{wg}(\pi) = \mathsf{wg}(v\pi)$, as required by the definition of quasi $\oplus$-dominion. Finally, if π is infinite and does meet v, it can be decomposed as $(v\pi')^\omega$, where π is a non-empty finite path that does not meet v. Then, by exploiting the same reasoning done above for the case where π is finite, we have that $\mathsf{wg}(v\pi') > 0$, which implies $\mathsf{wg}(\pi) = \mathsf{wg}((v\pi')^\omega) = \infty$.

- $[\mathsf{prg}_+]$. Let $\varrho^\star \triangleq \mathsf{prg}_+(\varrho)$ and consider the two infinite monotone sequences $\mathrm{Q}_0 \supseteq \mathrm{Q}_1 \supseteq \ldots$ and $\varrho_0 \sqsubseteq \varrho_1 \sqsubseteq \ldots$ defined as follows: $\mathrm{Q}_0 \triangleq \Delta(\varrho)$ and $\varrho_0 \triangleq \varrho$; $\mathrm{Q}_{i+1} \triangleq \mathrm{Q}_i \setminus \mathrm{E}_i$ and $\varrho_{i+1} = \mathsf{lift}(\varrho_i, \mathrm{E}_i, \overline{\mathrm{Q}_i})$, where $\mathrm{E}_i \triangleq \mathsf{bep}(\varrho_i, \mathrm{Q}_i) \subseteq \mathsf{esc}(\varrho_i, \mathrm{Q}_i)$, for all $i \in \mathbb{N}$. Since $|\mathrm{Q}_0| < \infty$, there necessarily exists an index $k \in \mathbb{N}$ such that $\mathrm{Q}_{k+1} = \mathrm{Q}_k$, $\varrho_{k+1} = \varrho_k$. Moreover, observe that $\varrho^\star = \mathsf{win}(\varrho_k, \mathrm{Q}_k)$. We first prove, by induction on the index $i \in \mathbb{N}$ of the sequences, that every ϱ_i satisfies Conditions 1a and 1c of Definition 4. Finally, we show that $\varrho^\star$ is a QDR.

  The base case $i = 0$ is trivial, since $\varrho_i = \varrho$ is a QDR. Now, let us consider the inductive case $i > 0$. Since the lift operator only modifies the measure of positions belonging to $\mathrm{E}_{i-1} \subseteq \mathrm{Q}_{i-1} \subseteq \Delta(\varrho) \subseteq \mathsf{qsi}(\varrho)$, it immediately follows that $\mathsf{qsi}(\varrho_i) = \mathsf{qsi}(\varrho_{i-1}) = \mathsf{qsi}(\varrho)$. Moreover, if $\sigma_{\varrho_i}(v) \neq \sigma_{\varrho_{i-1}}(v)$, we have that $\mu_{\varrho_{i-1}}(v) < \mu_{\varrho_i}(v) = \mu_{\varrho_i}(\sigma_{\varrho_i}(v)) = \mu_{\varrho_{i-1}}(\sigma_{\varrho_i}(v))$, for all positions $v \in \mathsf{qsi}(\varrho_i) \cap \mathrm{Ps}_{\oplus}$, where the latter equality is due to the fact that $\sigma_{\varrho_i}(v) \notin \mathrm{E}_{i-1}$. Thus, by Lemma 4, it holds that σ_{ϱ_i} is a $\oplus$-witness for $\mathsf{qsi}(\varrho_i)$, *i.e.*, Condition 1a is verified. Also, Condition 1c directly follows from the definition of the $\oplus$-strategy inside the lift operator.

At this point, we can conclude the proof by showing that $\varrho^\star$ is a QDR. Indeed, by Lemma 4, $\sigma_{\varrho^\star}$ is a $\oplus$-witness for $\mathsf{qsi}(\varrho_k) = \mathsf{qsi}(\varrho)$, so, Condition 1a is satisfied. Similarly to the inductive analysis developed above, Condition 1c directly follows from the definition of the $\oplus$-strategy inside the win function. Moreover, the set Q_k is a closed subset of $\mathsf{qsi}(\varrho_k)$, since $\mathrm{E}_k = \emptyset$ and, so, $\mathsf{esc}(\varrho_k, \mathrm{Q}_k) = \emptyset$. Therefore, $\mathrm{Q}_k \subseteq \mathrm{Wn}_{\oplus}$, by Proposition 1. In addition, all positions in $\|\mu_{\varrho_k}\|_{\oplus} \setminus (\|\mu\|_{\oplus} \cup \mathrm{Q}_k)$ necessarily reach $(\|\mu\|_{\oplus} \cup \mathrm{Q}_k) \subseteq \mathrm{Wn}_{\oplus}$. As a consequence, Condition 1b is verified as well.

It remains to prove Condition 1d. To do so, let

$$f_i \triangleq \mathsf{min}_{v \in \mathsf{esc}(\varrho_i, \mathrm{Q}_i)} \mathsf{bef}(\mu_{\varrho_i}, \mathrm{Q}_i, v)$$

We now first show that the sequence of natural numbers $f_0, f_1, \ldots$ is monotone, *i.e.*, $f_i \leq f_{i+1}$. Suppose by contradiction that $f_i > f_{i+1}$, for some index $i \in \mathbb{N}$. Then, there necessarily exists a position $v \in \mathsf{esc}(\varrho_{i+1}, \mathrm{Q}_{i+1}) \setminus \mathsf{esc}(\varrho_i, \mathrm{Q}_i)$ with $v \in \mathrm{E}_{i+1}$ such that $f_{i+1} = \mathsf{bef}(\mu_{\varrho_{i+1}}, \mathrm{Q}_{i+1}, v) < f_i$. We proceeds by a case analysis on the owner of the position v.

- **$[v \in \mathrm{Ps}_{\oplus}]$.** By definition of the best-escape forfeit function, we have that

  $$\begin{aligned} f_{i+1} &= \mathsf{max}\left\{\mu_{\varrho_{i+1}}(u) + v - \mu_{\varrho_{i+1}}(v) \,\middle|\, u \in Mv(v) \setminus \mathrm{Q}_{i+1}\right\} \\ &\geq \mu_{\varrho_{i+1}}(\sigma_{\varrho_i}(v)) + v - \mu_{\varrho_{i+1}}(v) \end{aligned}$$

  since $\sigma_{\varrho_i}(v) \in \mathrm{E}_i$ and, so, $\sigma_{\varrho_i}(v) \notin \mathrm{Q}_{i+1}$. Therefore, the following equalities and inequalities hold, which lead to the contradiction $f_i \leq f_{i+1} < f_i$:

  $$\begin{aligned} f_{i+1} &\geq \mu_{\varrho_{i+1}}(\sigma_{\varrho_i}(v)) + v - \mu_{\varrho_{i+1}}(v) \\ &= \mu_{\varrho_{i+1}}(\sigma_{\varrho_i}(v)) + \mathsf{wg}(v) - \mu_{\varrho_{i+1}}(v) \\ &= \mu_{\varrho_i}(\sigma_{\varrho_i}(v)) + f_i + \mathsf{wg}(v) - \mu_{\varrho_{i+1}}(v) \\ &= \mu_{\varrho_i}(\sigma_{\varrho_i}(v)) + f_i + \mathsf{wg}(v) - \mu_{\varrho_i}(v) \\ &= \mu_{\varrho_i}(\sigma_{\varrho_i}(v)) + v - \mu_{\varrho_i}(v) + f_i \\ &\geq f_i. \end{aligned}$$

  Notice that the first and last equality are due to the definition of the measure stretch operator. The second one is derived from the fact that $\sigma_{\varrho_i}(v) \in \mathrm{E}_i$, while the third one from $v \in \mathrm{E}_{i+1}$, which implies $\mu_{\varrho_{i+1}}(v) = \mu_{\varrho_i}(v)$. Finally, the last inequality follows from Condition 1c applied to ϱ_i, *i.e.*, $\mu_{\varrho_i}(v) \leq \mu_{\varrho_i}(\sigma_{\varrho_i}(v)) + v$.

– $[v \in \mathrm{Ps}_{\boxminus}]$. Again by definition of the best-escape forfeit function, we have that $f_{i+1} = \mathsf{min}\left\{\mu_{\varrho_{i+1}}(u) + v - \mu_{\varrho_{i+1}}(v) \,\middle|\, u \in Mv(v) \setminus \mathrm{Q}_{i+1}\right\}$. In addition, $Mv(v) \setminus \mathrm{Q}_{i+1} \subseteq \mathrm{E}_i$ Therefore, the following equalities hold:

$$
\begin{aligned}
f_{i+1} &= \mathsf{min}\left\{\mu_{\varrho_{i+1}}(u) + v - \mu_{\varrho_{i+1}}(v) \,\middle|\, u \in Mv(v) \setminus \mathrm{Q}_{i+1}\right\} \\
&= \mathsf{min}\left\{\mu_{\varrho_{i+1}}(u) + \mathsf{wg}(v) - \mu_{\varrho_{i+1}}(v) \,\middle|\, u \in Mv(v) \setminus \mathrm{Q}_{i+1}\right\} \\
&= \mathsf{min}\left\{\mu_{\varrho_{i}}(u) + f_i + \mathsf{wg}(v) - \mu_{\varrho_{i+1}}(v) \,\middle|\, u \in Mv(v) \setminus \mathrm{Q}_{i+1}\right\} \\
&= \mathsf{min}\{\mu_{\varrho_{i}}(u) + f_i + \mathsf{wg}(v) - \mu_{\varrho_{i}}(v) \,|\, u \in Mv(v) \setminus \mathrm{Q}_{i+1}\} \\
&= \mathsf{min}\{\mu_{\varrho_{i}}(u) + v - \mu_{\varrho_{i}}(v) + f_i \,|\, u \in Mv(v) \setminus \mathrm{Q}_{i+1}\} \\
&\geq f_i.
\end{aligned}
$$

Notice that the second and last equality are due to the definition of the measure stretch operator. The third one is derived from the fact that $u \in Mv(v) \setminus \mathrm{Q}_{i+1} \subseteq \mathrm{E}_i$, while the fourth one from $v \in \mathrm{E}_{i+1}$, which implies $\mu_{\varrho_{i+1}}(v) = \mu_{\varrho_i}(v)$. Finally, the last inequality follows from Condition 1d applied to ϱ, *i.e.*, $\mu_{\varrho_i}(v) = \mu_{\varrho}(v) \leq \mu_{\varrho}(u) + v \leq \mu_{\varrho_i}(u) + v$, for all adjacents $u \in Mv(v)$.

Now suppose by contradiction that Condition 1d does not hold for $\varrho^\star$. Then, there exist a $\boxminus$-position $v \in \mathsf{qsi}(\varrho^\star) \cap \mathrm{Ps}_{\boxminus}$ and one of its adjacents $u \in Mv(v)$ such that $\mu_{\varrho^\star}(u) + v < \mu_{\varrho^\star}(v)$. Due to the process used to compute $\varrho^\star$, there are indexes $i, j \in [0, k]$ such that $\mu_{\varrho^\star}(u) = \mu_{\varrho_{i+1}}(u) = \mu_{\varrho}(u) + f_i$ and $\mu_{\varrho^\star}(v) = \mu_{\varrho_{j+1}}(v) = \mu_{\varrho}(v) + f_j$. Now, by Condition 1d applied to ϱ, we have $\mu_{\varrho}(v) \leq \mu_{\varrho}(u) + v$, which implies that $0 \leq \mu_{\varrho}(u) + v - \mu_{\varrho}(v) < f_j - f_i$ and, consequently, both $i < j$ and $u \notin \mathrm{Q}_j$. However,

$$
\begin{aligned}
f_j - f_i &= \mathsf{min}\left\{\mu_{\varrho_j}(z) + v - \mu_{\varrho_j}(v) \,\middle|\, z \in Mv(v) \setminus \mathrm{Q}_j\right\} - f_i \\
&\leq \mu_{\varrho_j}(u) + v - \mu_{\varrho_j}(v) - f_i \\
&= \mu_{\varrho_j}(u) + v - \mu_{\varrho}(v) - f_i \\
&= \mu_{\varrho_{i+1}}(u) + v - \mu_{\varrho}(v) - f_i \\
&= (\mu_{\varrho}(u) + f_i) + v - \mu_{\varrho}(v) - f_i \\
&= \mu_{\varrho}(u) + v - \mu_{\varrho}(v),
\end{aligned}
$$

leading to the contradiction $\mu_{\varrho}(u) + v - \mu_{\varrho}(v) < f_j - f_i \leq \mu_{\varrho}(u) + v - \mu_{\varrho}(v)$. Notice that the first equality is due to the definition of the best-escape forfeit function. The second and third ones, instead, follows from the fact that v and u changed their values at iterations $j + 1$ and $i + 1$, respectively. Finally, the fourth equality derives from the operation of lift and best-escape forfeit computed on u.

□

Theorem 3 (Termination)**.** *The solver operator* $\mathsf{sol} \triangleq \mathsf{ifp}\, \varrho \,.\, \mathsf{prg}_{+}(\mathsf{prg}_{0}(\varrho))$ *is a well-defined total function. Moreover, for every simple* QDR $\varrho \in \mathrm{R}$ *it holds that*

$\mathsf{sol}(\varrho) = (\mathsf{ifp}_k\, \varrho^\star.\, \mathsf{prg}_+(\mathsf{prg}_0(\varrho^\star)))(\varrho)$*, for some index* $k \leq n \cdot (Z+1)$*, where* n *is the number of positions in the* MPG *and* Z *is the number of positive weights of all its simple paths,* i.e., $Z \triangleq |\{\mathsf{wg}(\pi) \in \mathbb{N} \mid \pi \in \mathrm{SPth}(\mathrm{Ps})\}|$.

Proof. Consider the sequence $\varrho_0, \varrho_1, \ldots$ recursively defined as follows:

$$\varrho_0 \triangleq (\mathsf{ifp}_0\, \varrho^\star.\, \mathsf{prg}_+(\mathsf{prg}_0(\varrho^\star)))(\varrho) = \varrho$$

and $\varrho_{i+1} \triangleq (\mathsf{ifp}_{i+1}\, \varrho^\star.\, \mathsf{prg}_+(\mathsf{prg}_0(\varrho^\star)))(\varrho) = \mathsf{prg}_+(\mathsf{prg}_0(\varrho_i))$, for all $i \in \mathbb{N}$. By induction on the index i, thanks to the totality and inflationary properties of the progress operators prg_0 and prg_+ previously proved in Theorem 2, one can easily show that every ϱ_i is a QDR satisfying $\varrho_i \sqsubseteq \varrho_{i+1}$. Moreover, by Proposition 5, we have that $\varrho_i(v) \in \{\mathsf{wg}(\pi) \in \mathbb{N} \mid \pi \in \mathrm{SPth}(\mathrm{Ps})\}$, for all positions $v \in \mathsf{qsi}(\varrho_i)$ with $\varrho_i(v) \neq \infty$ and index $i > 0$. Now, there are at most $n \cdot (Z+1)$ such QDRs, thus, there necessarily exists an index $k \leq n \cdot (Z+1)$ such that $\varrho_{k+1} = \varrho_k$, which implies $\mathsf{sol}(\varrho) = (\mathsf{ifp}\, \varrho^\star.\, \mathsf{lift}(\varrho^\star))(\varrho) = \varrho_k$. Hence, the thesis immediately follows. □

Theorem 5 (Soundness)**.** $\|\mathsf{sol}(\varrho)\|_\boxminus \subseteq \mathrm{Wn}_\boxminus$*, for every* $\varrho \in \mathrm{R}$.

Proof. Let $\varrho^\star \triangleq \mathsf{sol}(\varrho)$ be the result of the solver operator applied to $\varrho \in \mathrm{R}$. By Lemma 5, it holds that $\varrho^\star = \mathsf{prg}_0(\varrho^\star) = \mathsf{prg}_+(\varrho^\star)$. As a consequence, $\mu_{\varrho^\star}$ is a progress measure, due to Lemmas 1 and 2. At this point, by recalling that $\|\varrho^\star\|_\boxminus = \|\mu_{\varrho^\star}\|_\boxminus$, as reported in Definition 4, the thesis is immediately derived by applying Theorem 1 to $\mu_{\varrho^\star}$. □

Theorem 6 (Completeness)**.** $\|\mathsf{sol}(\varrho)\|_\oplus \subseteq \mathrm{Wn}_\oplus$*, for every* $\varrho \in \mathrm{R}$.

Proof. The thesis immediately follows by considering Theorem 3 and Condition 1b of Definition 4. Indeed, by the statement of the recalled theorem, $\mathsf{sol}(\varrho)$ is a QDR, independently of the element $\varrho \in \mathrm{R}$ given as input to the solver operator. Thus, thanks to the above condition, it holds the $\|\mathsf{sol}(\varrho)\|_\oplus \subseteq \mathrm{Wn}_\oplus$. □

Lemma 3. *Let* $\varrho^\star \triangleq \mathsf{prg}_+(\varrho)$*. Then,* $\mu_{\varrho^\star}(v) > \mu_\varrho(v)$*, for all positions* $v \in \Delta(\varrho)$.

Proof. Consider the set $\mathrm{E} \triangleq \mathsf{bep}(\varrho, \Delta(\varrho)) \subseteq \mathsf{esc}(\varrho, \Delta(\varrho))$ and let $\widehat{\varrho} \triangleq \mathsf{lift}(\varrho, \mathrm{E}, \overline{\Delta(\varrho)})$. First observe that $\mu_{\widehat{\varrho}}(v) = \mu_{\varrho^\star}(v)$, for all escape positions $v \in \mathrm{E}$. We now show that $\mu_{\varrho^\star}(v) > \mu_\varrho(v)$, via a case analysis on the owner of the position v itself.

- $[v \in \mathrm{Ps}_\oplus]$**.** By definition of the function esc, it holds that $\sigma_\varrho(v) \notin \Delta(\varrho)$ and $\mu_\varrho(v) \geq \mu_\varrho(u)+v$, for all adjacents $u \in Mv(v) \cap \Delta(\varrho)$. Since $v \in \Delta(\varrho)$, due to the way this specific weak quasi dominion is constructed, $v \in \mathsf{npp}(\varrho)$. Thus, there exists a successor $u^\star \in Mv(v)$ with $\mu_\varrho(v) < \mu_\varrho(u^\star) + v$, from which it follows that $u^\star \notin \Delta(\varrho)$, *i.e.*, $u^\star \in \overline{\Delta(\varrho)}$. As a consequence, we obtain that $\mu_{\widehat{\varrho}}(v) \geq \mu_\varrho(u^\star) + v > \mu_\varrho(v)$. Hence, $\mu_{\varrho^\star}(v) > \mu_\varrho(v)$.

- $[v \in \mathrm{Ps}_\boxminus]$**.** Since $v \in \Delta(\varrho)$, we have that $\mu_\varrho(v) < \mu_\varrho(u) + v$, for all adjacents $u \in Mv(v) \setminus \Delta(\varrho)$. Thus, $\mu_{\widehat{\varrho}}(v) = \min\{\mu_\varrho(u) + v \mid u \in Mv(v) \setminus \Delta(\varrho)\} > \mu_\varrho(v)$. Hence, $\mu_{\varrho^\star}(v) > \mu_\varrho(v)$ in this case as well.

Now, consider a position $v \in \Delta(\varrho) \setminus \mathrm{E}$. Obviously, $\mu_{\varrho}(v) < \infty$. If $\mu_{\varrho^\star}(v) = \infty$, the thesis immediately follows. Otherwise, it will be considered as an escape of some weak quasi dominion $\mathrm{Q} \subset \Delta(\varrho)$, after the removal of the first escape positions in E. Due to the non-decreasing property of the sequence of best-escape forfeit shown in the proof of Theorem 2, v exits from Q with a forfeit $f^\star$ at least as great as the one f of E that we just proved to be strictly positive. Indeed, $f = \mu_{\varrho^\star}(z) - \mu_{\varrho}(z) > 0$, for all $z \in \mathrm{E}$. Therefore, $\mu_{\varrho^\star}(v) - \mu_{\varrho}(v) = f^\star \geq f > 0$, which implies $\mu_{\varrho^\star}(v) > \mu_{\varrho}(v)$. □

Lemma 4. *Let $\varrho \in \mathrm{R}$ and $\sigma^\star \in \mathrm{Str}_{\oplus}(\mathsf{qsi}(\varrho))$ a $\oplus$-strategy such that, if $\sigma^\star(v) \neq \sigma_{\varrho}(v)$, then $\mu_{\varrho}(v) < \mu_{\varrho}(\sigma^\star(v)) + v$, for all positions $v \in \mathsf{qsi}(\varrho) \cap \mathrm{Ps}_{\oplus}$. Then, $\sigma^\star$ is a $\oplus$-witness for $\mathsf{qsi}(\varrho)$.*

Proof. The proof proceed by induction on the number $i \triangleq |\mathrm{D}|$ of the positions in

$$\mathrm{D} \triangleq \{v \in \mathrm{Ps}_{\oplus} \mid \sigma^\star(v) \neq \sigma_{\varrho}(v)\}$$

on which the two strategies $\sigma^\star$ and σ_{ϱ} differ. The base case $i = 0$ is immediate, since ϱ is a QDR. Therefore, assume $i > 0$, let $v \in \mathrm{D}$, and consider the strategy $\widehat{\sigma} \in \mathrm{Str}_{\oplus}(\mathsf{qsi}(\varrho))$ such that $\widehat{\sigma}(v) = \sigma_{\varrho}(v)$ and $\widehat{\sigma}(u) = \sigma^\star(u)$, for all positions $u \in \mathsf{qsi}(\varrho) \cap \mathrm{Ps}_{\oplus}$ with $u \neq v$. By the inductive hypothesis, we have that $\widehat{\sigma}$ is a $\oplus$-witness for the quasi dominion $\mathsf{qsi}(\varrho)$. Now, consider an arbitrary path π compatible with the $\oplus$-strategy $\sigma^\star$. If π does not meet v, it is necessarily compatible with the $\oplus$-strategy $\widehat{\sigma}$, thus, $\mathsf{wg}(\pi) > 0$. If π meets v once, then it can be decomposed as $\pi' v \pi''$, where π' and π'' are paths not meeting v, where only the first can be possibly empty. On the one hand, if π'' is infinite, by Proposition 1, we have $\mathsf{wg}(\pi'') = \infty$ and, so, $\mathsf{wg}(\pi) = \mathsf{wg}(\pi' v \pi'') = \infty$. On the other hand, if π'' is finite, then, by Propositions 2 and 4, we have that $0 < \mu_{\varrho}(\mathsf{fst}(\pi' v)) \leq \mathsf{wg}(\pi') + \mu_{\varrho}(\mathsf{lst}(\pi' v)) = \mathsf{wg}(\pi') + \mu_{\varrho}(v)$ and $\mu_{\varrho}(\mathsf{fst}(\pi'')) \leq \mathsf{wg}(\pi'')$, since both π' and π'' are compatible with $\widehat{\sigma}$. Moreover, $\mu_{\varrho}(v) < \mu_{\varrho}(\sigma^\star(v)) + v = \mu_{\varrho}(\mathsf{fst}(\pi'')) + v = \mu_{\varrho}(\mathsf{fst}(\pi'')) + \mathsf{wg}(v)$. Now, by putting all things together, we have $0 < \mu_{\varrho}(\mathsf{fst}(\pi' v)) \leq \mathsf{wg}(\pi') + \mu_{\varrho}(v) < \mathsf{wg}(\pi') + \mu_{\varrho}(\mathsf{fst}(\pi'')) + \mathsf{wg}(v) \leq \mathsf{wg}(\pi') + \mathsf{wg}(v) + \mathsf{wg}(\pi'') = \mathsf{wg}(\pi' v \pi'')$, *i.e.*, $\mathsf{wg}(\pi) > 0$. Finally, consider the case where π meets v more than once and, so, infinitely many times, due to the regularity of the path, which is in its turn due to the memoryless strategies. Then, π can be written as $\pi'(v\pi'')^\omega = \pi' v (\pi'' v)^\omega$, where π' and π'' are possibly empty paths not meeting v. First observe that the finite path $\pi'' v$ is compatible with $\widehat{\sigma}$, thus, by Proposition 2, we have that $\mu_{\varrho}(\mathsf{fst}(\pi'' v)) \leq \mathsf{wg}(\pi'') + \mu_{\varrho}(v)$. Moreover, $\mu_{\varrho}(v) < \mu_{\varrho}(\mathsf{fst}(\pi'' v)) + \mathsf{wg}(v)$, as already shown above. Hence, $\mu_{\varrho}(v) < \mu_{\varrho}(\mathsf{fst}(\pi'')) + \mathsf{wg}(v) \leq \mathsf{wg}(\pi'') + \mathsf{wg}(v) + \mu_{\varrho}(v) = \mathsf{wg}(\pi'' v) + \mu_{\varrho}(v)$, which implies $\mathsf{wg}(\pi'' v) > 0$. As a consequence, $\mathsf{wg}((\pi'' v)^\omega) = \infty$ and, so, $\mathsf{wg}(\pi) = \mathsf{wg}(\pi' v (\pi'' v)^\omega) > 0$. Summing up, $\sigma^\star$ is a $\oplus$-witness for the quasi dominion $\mathsf{qsi}(\varrho)$ as required by the lemma statement. □

Lemma 5. *Let $\varrho^\star \triangleq \mathsf{sol}(\varrho)$ be the result of the solver operator applied to an arbitrary $\varrho \in \mathrm{R}$. Then, $\varrho^\star$ is a fixpoint of the progress operators,* i.e., *$\varrho^\star = \mathsf{prg}_0(\varrho^\star) = \mathsf{prg}_+(\varrho^\star)$.*

Proof. By definition of inflationary fixpoint, $\varrho^\star$ is a fixpoint of the composition of the two progress operators, *i.e.*, $\varrho^\star = \mathsf{prg}_+(\mathsf{prg}_0(\varrho^\star))$, which are inflationary functions, due to Theorem 2. As a consequence, we have that $\varrho^\star = \mathsf{prg}_+(\mathsf{prg}_0(\varrho^\star)) \sqsupseteq \mathsf{prg}_0(\varrho^\star) \sqsupseteq \varrho^\star$. Thus, $\mathsf{prg}_0(\varrho^\star) = \varrho^\star$ and, so, $\mathsf{prg}_+(\varrho^\star) = \varrho^\star$. □

References

Agrawal, M., Kayal, N., Saxena, N., 2004. PRIMES is in P. Annals of Mathematics 160, 781–793.

Allamigeon, X., Benchimol, P., Gaubert, S., 2014a. Combinatorial Simplex Algorithms Can Solve Mean-Payoff Games. SIAM Journal on Optimization 24, 2096–2117.

Allamigeon, X., Benchimol, P., Gaubert, S., 2014b. The Tropical Shadow-Vertex Algorithm Solves Mean-Payoff Games in Polynomial Time on Average, in: International Colloquium on Automata, Languages, and Programming'14, Springer. pp. 89–100.

Allamigeon, X., Benchimol, P., Gaubert, S., Joswig, M., 2015. Tropicalizing the Simplex Algorithm. SIAM Journal on Discrete Mathematics 29, 751–795.

Austin, P., Dell'Erba, D., 2023. Errata to: 'Faster Deterministic Exponential Time Algorithm for Energy Games and Mean Payoff Games'. Technical Report. arXiv.

Benerecetti, M., Dell'Erba, D., Faella, M., Mogavero, F., 2023. From Quasi-Dominions to Progress Measures, in: Automata Theory and Applications: Games, Learning and Structures, September 20-24, 2021, Virtual, World Scientific. pp. 159–199.

Benerecetti, M., Dell'Erba, D., Mogavero, F., 2016a. A Delayed Promotion Policy for Parity Games, in: Games, Automata, Logics, and Formal Verification'16, pp. 30–45.

Benerecetti, M., Dell'Erba, D., Mogavero, F., 2016b. Improving Priority Promotion for Parity Games, in: Haifa Verification Conference16, Springer. pp. 1–17.

Benerecetti, M., Dell'Erba, D., Mogavero, F., 2016c. Solving Parity Games via Priority Promotion, in: Computer Aided Verification'16, Springer. pp. 270–290.

Benerecetti, M., Dell'Erba, D., Mogavero, F., 2018a. A Delayed Promotion Policy for Parity Games. Information and Computation 262, 221–240.

Benerecetti, M., Dell'Erba, D., Mogavero, F., 2018b. Solving Parity Games via Priority Promotion. Formal Methods in System Design 52, 193–226.

Benerecetti, M., Dell'Erba, D., Mogavero, F., 2020. Solving Mean-Payoff Games via Quasi Dominions, in: Tools and Algorithms for the Construction and Analysis of Systems'20, Springer. pp. 289–306.

Björklund, H., Sandberg, S., Vorobyov, S., 2004. A Combinatorial Strongly Subexponential Strategy Improvement Algorithm for Mean-Payoff Games, in: Mathematical Foundations of Computer Science'04, Springer. pp. 673–685.

Björklund, H., Vorobyov, S., 2007. A Combinatorial Strongly Subexponential Strategy Improvement Algorithm for Mean-Payoff Games. Discrete Applied Mathematics 155, 210–229.

Bloem, R., Chatterjee, K., Henzinger, T., Jobstmannand, B., 2009. Better Quality in Synthesis Through Quantitative Objectives, in: Computer Aided Verification'09, pp. 140–156.

Bohy, A., Bruyère, V., Filiot, E., Raskin, J.F., 2013. Synthesis from LTL Specifications with Mean-Payoff Objectives, in: Tools and Algorithms for the Construction and Analysis of Systems'13, pp. 169–184.

Boker, U., Chatterjee, K., Henzinger, T., Kupferman, O., 2011. Temporal Specifications with Accumulative Values, in: Logic in Computer Science'11, IEEE Computer Society. pp. 43–52.

Bourbaki, N., 1949. Sur le Théorème de Zorn. Archiv der Mathematik 2, 434–437.

Bouyer, P., Fahrenberg, U., Larsen, K., Markey, N., Srba, J., 2008. Infinite Runs in Weighted Timed Automata with Energy Constraints, in: Formal Modeling and Analysis of Timed Systems'2008, Springer. pp. 33–47.

Brim, L., Chaloupka, J., 2012. Using Strategy Improvement to Stay Alive. International Journal of Foundations of Computer Science 23, 585–608.

Brim, L., Chaloupka, J., Doyen, L., Gentilini, R., Raskin, J.F., 2011. Faster Algorithms for Mean-Payoff Games. Formal Methods in System Design 38, 97–118.

Comin, C., Posenato, R., Rizzi, R., 2017. Hyper Temporal Networks - A Tractable Generalization of Simple Temporal Networks and its Relation to Mean-Payoff Games. Constraints 22.

Comin, C., Rizzi, R., 2015. Dynamic Consistency of Conditional Simple Temporal Networks via Mean-Payoff Games: A Singly-Exponential Time DC-checking, in: Temporal Representation and Reasoning'15, IEEE Computer Society. pp. 19–28.

Comin, C., Rizzi, R., 2017. Improved Pseudo-Polynomial Bound for the Value Problem and Optimal Strategy Synthesis in Mean-Payoff Games. Algorithmica 77.

Condon, A., 1992. The Complexity of Stochastic Games. Information and Computation 96, 203–224.

Dhingra, V., Gaubert, S., 2006. How to Solve Large Scale Deterministic Games with Mean Payoff by Policy Iteration, in: Performance Evaluation Methodologies and Tools'06, Association for Computing Machinery. p. 12.

van Dijk, T., 2018. Oink: an Implementation and Evaluation of Modern Parity Game Solvers, in: Tools and Algorithms for the Construction and Analysis of Systems'18, Springer. pp. 291–308.

Dorfman, D., Kaplan, H., Zwick, U., 2019. A Faster Deterministic Exponential Time Algorithm for Energy Games and Mean Payoff Games, in: International Colloquium on Automata, Languages, and Programming'19, Leibniz-Zentrum fuer Informatik. pp. 1–14.

Ehrenfeucht, A., Mycielski, J., 1979. Positional Strategies for Mean Payoff Games. International Journal of Game Theory 8, 109–113.

Fearnley, J., 2010. Non-Oblivious Strategy Improvement, in: Logic for Programming Artificial Intelligence and Reasoning'10, Springer. pp. 212–230.

Fearnley, J., Jain, S., Schewe, S., Stephan, F., Wojtczak, D., 2017. An Ordered Approach to Solving Parity Games in Quasi Polynomial Time and Quasi Linear Space, in: SPIN Symposium on Model Checking of Software'17, Association for Computing Machinery. pp. 112–121.

Fellows, M., Koblitz, N., 1992. Self-Witnessing Polynomial-Time Complexity and Prime Factorization, in: Conference on Structure in Complexity Theory'92, IEEE Computer Society. pp. 107–110.

Friedmann, O., Lange, M., 2009. Solving Parity Games in Practice, in: Automated Technology for Verification and Analysis'09, Springer. pp. 182–196.

Gabow, H., Tarjan, R., 1991. Faster Scaling Algorithms for General Graph Matching Problems. Journal of the ACM 38, 815–853.

Goldberg, A., 1995. Scaling Algorithms for the Shortest Paths Problem. SIAM Journal on Computing 24, 494–504.

Goldberg, A., Rao, S., 1998. Beyond the Flow Decomposition Barrier. Journal of the ACM 45, 783–797.

Gurvich, V., Karzanov, A., Khachivan, L., 1988. Cyclic Games and an Algorithm to Find Minimax Cycle Means in Directed Graphs. USSR Computational Mathematics and Mathematical Physics 28, 85–91.

Jurdziński, M., 1998. Deciding the Winner in Parity Games is in UP $\cap$ co-UP. Information Processing Letters 68, 119–124.

Jurdziński, M., 2000. Small Progress Measures for Solving Parity Games, in: Symposium on Theoretical Aspects of Computer Science'00, Springer. pp. 290–301.

Jurdziński, M., Lazic, R., 2017. Succinct Progress Measures for Solving Parity Games, in: Logic in Computer Science'17, Association for Computing Machinery. pp. 1–9.

Klarlund, N., 1991. Liminf Progress Measures, in: Mathematical Foundations of Programming Semantics'91, Springer. pp. 477–491.

Lifshits, Y., Pavlov, D., 2007. Potential theory for mean payoff games. Journal of Mathematical Sciences 145, 4967–4974.

Pisaruk, N., 1999. Mean-Cost Cyclical Games. Mathematics of Operations Research 24, 817–828.

Schewe, S., 2008. An Optimal Strategy Improvement Algorithm for Solving Parity and Payoff Games, in: Computer Science Logic'08, Springer. pp. 369–384.

Witt, E., 1950. Beweisstudien zum Satz von M. Zorn. Mathematische Nachrichten 4, 434–438.

Zwick, U., Paterson, M., 1996. The Complexity of Mean Payoff Games on Graphs. Theoretical Computer Science 158, 343–359.