On the Complexity of Heterogeneous Multidimensional Quantitative Games

Véronique Bruyère[1], Quentin Hautem[*2], and Jean-François Raskin[†3]

1 **Département d'informatique, Université de Mons (UMONS), Mons, Belgium**
2 **Département d'informatique, Université de Mons (UMONS), Mons, Belgium**
3 **Département d'informatique, Université Libre de Bruxelles (U.L.B.), Brussels, Belgium**

Abstract

We study two-player zero-sum turn-based games played on multidimensional weighted graphs with heterogeneous quantitative objectives. Our objectives are defined starting from the measures Inf, Sup, LimInf, and LimSup of the weights seen along the play, as well as on the window mean-payoff (WMP) measure recently introduced in [6]. Whereas multidimensional games with Boolean combinations of classical mean-payoff objectives are undecidable [19], we show that CNF/DNF Boolean combinations for heterogeneous measures taken among {WMP, Inf, Sup, LimInf, LimSup} lead to EXPTIME-completeness with exponential memory strategies for both players. We also identify several interesting fragments with better complexities and memory requirements, and show that some of them are solvable in PTIME.

1998 ACM Subject Classification B.6.3 [design aids]: automatic synthesis; F.1.2 [Modes of computation]: interactive and reactive computation

Keywords and phrases two-player zero-sum games played on graphs, quantitative objectives, multidimensional heterogeneous objectives

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2016.11

1 Introduction

Two-player zero-sum turn-based games played on graphs are an adequate mathematical model to solve the *reactive synthesis problem* [18]. To model systems with resource constraints, like embedded systems, games with quantitative objectives have been studied, e.g. mean-payoff [22] and energy games [3]. In [5, 21, 6, 20], multidimensional games with conjunctions of several quantitative objectives have been investigated, such that all dimensions use the *same* measure. In this paper, we initiate the study of games played on multidimensional weighted graphs such that the objectives use *different* measures over the dimensions. As an example of conjunction of heterogeneous measures, you may want to design a system with (ϕ_1) a good window mean-response time (MP), that (ϕ_2) avoids too slow reaction after a finite prefix (LimInf), and that (ϕ_3) does not exceed some peak energy consumption in

* Author supported by FRIA fellowship
† Author supported by ERC Starting Grant (279499: inVEST) and partly by European project Cassting (FP7-ICT-601148).

the long run (LimSup). Now, assume that you want to ensure such a conjunction only under the hypothesis that (ψ) the frequency of requests from the environment is below some threshold (expressible as an MP). Such a property $\psi \to (\phi_1 \wedge \phi_2 \wedge \phi_3)$ is equivalent to the DNF Boolean combination of heterogeneous measures $\neg\psi \vee (\phi_1 \wedge \phi_2 \wedge \phi_3)$. We claim that such heterogeneous quantitative games provide a general, natural, and expressive model for the reactive synthesis problem and that the complexity of solving these games needs to be studied. Knowing that Boolean combination of MP objectives is undecidable [19], we have initiated this study with the omega-regular measures Inf, Sup, LimInf and LimSup, and the recent interesting window version WMP of MP introduced in [6].

While the MP measure considers the long-run average of the weights along the whole play, the WMP measure considers weights over a *local window of a given size* sliding along the play. A WMP objective asks now to ensure that the average weight satisfies a given constraint over every bounded window. This is a strengthening of the MP objective: winning for the WMP objective implies winning for the MP objective. Also, any finite-memory strategy that forces an MP measure larger than threshold $\nu + \epsilon$ (for any $\epsilon > 0$), also forces the WMP measure to be larger than ν provided that the window size is taken large enough. Aside from their naturalness, WMP objectives are algorithmically more tractable than classical MP objectives, see [6, 14]. First, unidimensional WMP games can be solved in polynomial time when working with polynomial windows [6] while only pseudo-polynomial time algorithms are known for MP games [22, 4]. Second, multidimensional games with Boolean combinations of MP objectives are undecidable [19], whereas we show here that games with Boolean combinations of WMP objectives and other classical objectives are decidable.

We show in this paper (see also Table 1) that the problem is EXPTIME-complete for CNF/DNF Boolean combinations of heterogeneous measures taken among {WMP, Inf, Sup, LimInf, LimSup}. We provide a detailed study of the complexity when the Boolean combination of the measures is replaced by an intersection, as it is often natural in practice to consider conjunction of constraints. EXPTIME-completeness of the problem still holds for the intersection of measures in {WMP, Inf, Sup, LimInf, LimSup}, and we get PSPACE-completeness when WMP measure is not considered. To avoid EXPTIME-hardness, we consider fragments where there is at most one occurrence of a WMP measure. In case of intersections of one WMP objective with any number of objectives of one kind among {Inf, Sup, LimInf, LimSup} (this number must be fixed in case of objectives Sup), we get P-completeness when dealing with polynomial windows, a reasonable hypothesis in practical applications. In case of no occurrence of WMP measure, we propose several refinements (on the number of occurrences of the other measures) for which we again get P-completeness. Some of our results are obtained by reductions to known qualitative games but most of them are obtained by new algorithms that require new ideas to handle in an optimal way one WMP objective together with qualitative objectives such as safety, reachability, Büchi and coBüchi objectives. In our results, we also provide a careful analysis of the memory requirements of winning strategies for both players.[1]

Let us mention some related work. Multidimensional mean-payoff games have been studied in [20]. Conjunction of lim inf mean-payoff ($\underline{\text{MP}}$) objectives are coNP-complete, conjunctions of lim sup MP ($\overline{\text{MP}}$) objectives are in $\text{NP} \cap \text{coNP}$. The general case of Boolean combinations of $\underline{\text{MP}}$ and $\overline{\text{MP}}$ is undecidable [19]. Multidimensional energy games with unfixed initial credit are coNP-complete [5, 8], and with fixed initial credit, they are 2EXPTIME-complete [16]. Generalization of these games with imperfect information have been studied

[1] All details of this paper can be found in the arXiv version arXiv:1511.08334v2.

Table 1 Overview - Our results are marked with (*).

<table>
<tr><th>Objectives</th><th>Complexity class</th><th>Player 1 memory</th><th>Player 2 memory</th></tr>
<tr><td>(CNF/DNF) Boolean combination of $\underline{\mathsf{MP}}$, $\overline{\mathsf{MP}}$ [19]</td><td>Undecidable</td><td>infinite</td><td>infinite</td></tr>
<tr><td>(CNF/DNF) Boolean combinaison of
WMP, Inf, Sup, LimInf, LimSup (*)</td><td rowspan="3">EXPTIME-complete</td><td colspan="2" rowspan="3">exponential</td></tr>
<tr><td>Intersection of WMP, Inf, Sup, LimInf, LimSup (*)</td></tr>
<tr><td>Intersection of WMP [6]</td></tr>
<tr><td rowspan="2">Intersection of Inf, Sup, LimInf, LimSup (*)
and refinements (*)</td><td>PSPACE-complete</td></tr>
<tr><td colspan="3">See Table 4</td></tr>
<tr><td>Intersection of $\underline{\mathsf{MP}}$ [20]</td><td>coNP-complete</td><td rowspan="2">infinite</td><td rowspan="3">memoryless</td></tr>
<tr><td>Intersection of $\overline{\mathsf{MP}}$ [20]</td><td rowspan="2">NP ∩ coNP</td></tr>
<tr><td>Unidimensional MP [22, 4]</td><td>memoryless</td></tr>
<tr><td>Unidimensional WMP [6]
WMP ∩ Ω with Ω ∈ {Inf, Sup, LimInf, LimSup} (*)</td><td>P-complete
(Polynomial windows)</td><td colspan="2">pseudo-polynomial</td></tr>
<tr><td>Unidimensional Inf, Sup, LimInf, LimSup [13]</td><td>P-complete</td><td colspan="2">memoryless</td></tr>
</table>

in [9] and shown undecidable. The WMP measure was first introduced in [6]. Unidimensional WMP games can be solved in polynomial time for polynomial windows, and multidimensional WMP games are EXPTIME-complete. In [6], the WMP measure is considered on all the dimensions, with no conjunction with other measures like Inf, Sup, LimInf, and LimSup, and the case of Boolean combinations of WMP objectives is not investigated. Games with objectives expressed in fragments of LTL have been studied in [1]. Our result that games with intersection of objectives in {Inf, Sup, LimInf, LimSup} are in PSPACE can be obtained by reduction to some of these fragments. But we here propose a simple proof adapted to our context, that allows to identify several polynomial fragments. Our other results cannot be obtained in this way and require new techniques and new algorithmic ideas.

2 Preliminaries

We consider turn-based two-player games on a finite multidimensional weighted directed graph. A *multi-weighted game structure* is a tuple $G = (V_1, V_2, E, w)$ where (i) (V, E) is a finite directed graph, with V the set of vertices and $E \subseteq V \times V$ the set of edges such that for each $v \in V$, there exists $(v, v') \in E$ for some $v' \in V$ (no deadlock), (ii) (V_1, V_2) forms a partition of V such that V_p is the set of vertices controlled by player $p \in \{1, 2\}$, and (iii) $w : E \to \mathbb{Z}^n$ is the n-dimensional weight function that associates a vector of n weights to each edge, for some $n \geq 1$. We also simply say that G is a (n-weighted) game structure.

The opponent of player $p \in \{1, 2\}$ is denoted by $\overline{p}$. A *play* of G is an infinite sequence $\rho = \rho_0\rho_1 \ldots \in V^\omega$ such that $(\rho_k, \rho_{k+1}) \in E$ for all $k \in \mathbb{N}$. *Histories* of G are finite sequences $\rho = \rho_0 \ldots \rho_i \in V^+$ defined in the same way. We denote by $\mathsf{Plays}(G)$ the set of plays in G and by $\mathsf{Hist}(G)$ the set of histories. Given a play $\rho = \rho_0\rho_1 \ldots$, the history $\rho_k \ldots \rho_{k+i}$ is denoted by $\rho_{[k,k+i]}$. We denote by w_m the projection of function w on the m^{th} dimension, and by W the maximum weight in absolute value on all dimensions.

Strategies, objectives and winning sets

A *strategy* σ for player $p \in \{1, 2\}$ is a function $\sigma : V^*V_p \to V$ assigning to each history $hv \in V^*V_p$ a vertex $v' = \sigma(hv)$ such that $(v, v') \in E$. It is *memoryless* if $\sigma(hv) = \sigma(h'v)$ for all histories $hv, h'v$ ending with the same vertex v, that is, σ is a function $\sigma : V_p \to V$. It

is *finite-memory* if $\sigma(hv)$ only needs finite memory of the history hv (recorded by a Moore machine). Given a strategy σ of player $p \in \{1,2\}$, we say that a play ρ of G is *consistent* with σ if $\rho_{k+1} = \sigma(\rho_0 \dots \rho_k)$ for all $k \in \mathbb{N}$ such that $\rho_k \in V_p$. A history consistent with a strategy is defined similarly. Given an initial vertex v_0, and a strategy σ_p of each player p, we have a unique play that is consistent with both strategies, called the *outcome* of (σ_1, σ_2) from v_0, and denoted by $\mathsf{Out}(v_0, \sigma_1, \sigma_2)$.

An *objective for player p* is a set of plays $\Omega \subseteq \mathsf{Plays}(G)$; it is *qualitative* (it only depends on the graph (V, E)), or *quantitative* (it also depends on the weight function w). A play ρ is *winning* for player p if $\rho \in \Omega$, and losing otherwise (i.e. winning for player $\overline{p}$). We thus consider zero-sum games such that the objective of player $\overline{p}$ is $\overline{\Omega} = \mathsf{Plays}(G) \setminus \Omega$, i.e., opposite to the objective Ω of player p. In the following, we always take the point of view of player 1 by supposing that Ω is his objective, and we denote by (G, Ω) the corresponding *game*. A strategy σ_p for player p is *winning* from an initial vertex v_0 if $\mathsf{Out}(v_0, \sigma_p, \sigma_{\overline{p}}) \in \Omega$ for all strategies $\sigma_{\overline{p}}$ of player $\overline{p}$. Vertex v_0 is also called *winning* for player p and the *winning set* Win_p^{Ω} is the set of all his winning vertices. Similarly the winning vertices of player $\overline{p}$ are those from which $\overline{p}$ can ensure his objective $\overline{\Omega}$ against all strategies of player p, and $\mathsf{Win}_{\overline{p}}^{\overline{\Omega}}$ is his winning set. The game is said *determined* when $\mathsf{Win}_p^{\Omega} \cup \mathsf{Win}_{\overline{p}}^{\overline{\Omega}} = V$. It is known that every game with Borel objectives is determined [17].

Qualitative objectives

Let $G = (V_1, V_2, E)$ be an *unweighted* game structure. Given a set $U \subseteq V$, classical qualitative objectives are the following ones. A *reachability* objective $\mathsf{Reach}(U)$ asks to visit a vertex of U at least once. A *safety* objective $\mathsf{Safe}(U)$ asks to visit no vertex of $V \setminus U$, that is, to avoid $V \setminus U$. A *Büchi* objective $\mathsf{Buchi}(U)$ asks to visit a vertex of U infinitely often. A *co-Büchi* objective $\mathsf{CoBuchi}(U)$ asks to visit no vertex of $V \setminus U$ infinitely often. Let $U_1, \dots, U_i$ be a family of subsets of V. A *generalized reachability* objective $\mathsf{GenReach}(U_1, \dots, U_i) = \cap_{k=1}^{i} \mathsf{Reach}(U_k)$ asks to visit a vertex of U_k at least once, for each $k \in \{1, \dots, i\}$. A *generalized Büchi* objective $\mathsf{GenBuchi}(U_1, \dots, U_i) = \cap_{k=1}^{i} \mathsf{Buchi}(U_k)$ asks to visit a vertex of U_k infinitely often, for each $k \in \{1, \dots, i\}$. All these objectives can be mixed by taking their intersection.

A game with an objective Ω is just called Ω *game*. As all the previous objectives Ω are ω-regular and thus Borel, the corresponding games (G, Ω) are determined.

Quantitative objectives

For a *1-weighted* game structure $G = (V_1, V_2, E, w)$ (with dimension $n = 1$), we now introduce quantitative objectives defined by some classical *measure functions* $f : \mathsf{Plays}(G) \to \mathbb{Q}$. Such a function f associates a rational number to each play $\rho = \rho_1 \rho_2 \dots$ according to the weights $w(\rho_k, \rho_{k+1})$, $k \geq 0$, and can be one among the next functions. The Inf measure $\mathsf{Inf}(\rho) = \inf_{k \geq 0}(w(\rho_k, \rho_{k+1}))$ (resp. the Sup measure $\mathsf{Sup}(\rho) = \sup_{k \geq 0}(w(\rho_k, \rho_{k+1}))$) defines the minimum (resp. maximum) weight seen along the play. The LimInf measure $\mathsf{LimInf}(\rho) = \liminf_{k \to \infty}(w(\rho_k, \rho_{k+1}))$ (resp. the LimSup measure $\mathsf{LimSup}(\rho) = \limsup_{k \to \infty}(w(\rho_k, \rho_{k+1}))$) defines the mininum (resp. maximum) weight seen infinitely often along the play.

Given such a measure function $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$, a bound $\nu \in \mathbb{Q}$, and a relation $\sim \in \{>, \geq, <, \leq\}$, we define the objective $\Omega = f(\sim \nu)$ such that

$$f(\sim \nu) \quad = \quad \{\rho \in \mathsf{Plays}(G) \mid f(\rho) \sim \nu\}. \tag{1}$$

We are also interested in the next two measure functions defined on histories instead of plays. Let $\rho = \rho_0 \dots \rho_i \in \mathsf{Hist}(G)$. The *total-payoff* ($\mathsf{TP}$) measure $\mathsf{TP}(\rho) = \Sigma_{k=0}^{i-1} w(\rho_k, \rho_{k+1})$

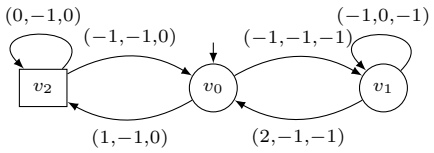

Figure 1 A multi-weighted two-player game.

defines the sum of the weights seen along ρ. The *mean-payoff* (MP) measure $\mathsf{MP}(\rho) = \frac{1}{i}\mathsf{TP}(\rho)$ defines the mean of the weights seen along ρ. The second measure can be extended to plays ρ as either $\underline{\mathsf{MP}}(\rho) = \liminf_{k\geq 0} \mathsf{MP}(\rho_{[0,k]})$ or $\overline{\mathsf{MP}}(\rho) = \limsup_{k\geq 0} \mathsf{MP}(\rho_{[0,k]})$. The MP measure on histories allows to define the *window mean-payoff objective*, a new ω-regular objective introduced in [6]: given a bound $\nu \in \mathbb{Q}$, a relation $\sim \in \{>, \geq, <, \leq\}$, and a window size $\lambda \in \mathbb{N}\setminus\{0\}$, the objective $\mathsf{WMP}(\lambda, \sim \nu)$[2] is equal to

$$\mathsf{WMP}(\lambda, \sim \nu) = \{\rho \in \mathsf{Plays}(G) \mid \forall k \geq 0, \exists l \in \{1, \ldots, \lambda\}, \mathsf{MP}(\rho_{[k,k+l]}) \sim \nu\}. \tag{2}$$

The window mean-payoff objective asks that the average weight becomes $\sim \nu$ inside a local bounded window for all positions of this window sliding along the play, instead of the classical mean-payoff objective asking that the long run-average $\underline{\mathsf{MP}}(\rho)$ (resp. $\overline{\mathsf{MP}}(\rho)$) is $\sim \nu$. This objective is a strengthening of the mean-payoff objective.

Given a n-weighted game structure G (with $n \geq 1$), we can mix objectives of (1) and (2) by fixing one such objective Ω_m for each dimension m, and taking the intersection $\cap_{m=1}^{n}\Omega_m$. More precisely, given a vector $(\sim_1 \nu_1, \ldots, \sim_n \nu_n)$, each objective Ω_m uses a measure function based on the weight function w_m; Ω_m is either of the form $f(\sim_m \nu_m)$ with $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$, or of the form $\mathsf{WMP}(\lambda, \sim_m \nu_m)$ for some window size λ (this size can change with m).

As for qualitative objectives, we use the shortcut Ω *game* for a game with quantitative objective Ω. For instance an $\mathsf{Inf}(\sim \nu)$ *game* is a 1-weighted game with objective $\mathsf{Inf}(\sim \nu)$, a $\mathsf{LimSup}(\sim_1 \nu_1)\cap\mathsf{WMP}(\lambda, \sim_2 \nu_2)\cap\mathsf{Inf}(\sim_3 \nu_3)$ *game* is a 3-weighted game with the intersection of a $\mathsf{LimSup}(\sim_1 \nu_1)$ objective on the first dimension, a $\mathsf{WMP}(\lambda, \sim_2 \nu_2)$ objective on the second one, and an $\mathsf{Inf}(\sim_3 \nu_3)$ objective on the third one. We sometimes abusively say that $\Omega = \cap_{m=1}^{n}\Omega_m$ with $\Omega_m \in \{\mathsf{WMP}, \mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$ without mentioning the used relations, bounds and window sizes. It is implicitly supposed that Ω_m deals with the m^{th} component of the weight function.

3 Problem

In this paper, we want to study the following problem.

▶ **Problem 1.** Let (G, Ω) be a multi-weighted game with dimension $n \geq 1$ and $\Omega = \cap_{m=1}^{n}\Omega_m$ such that each $\Omega_m \in \{\mathsf{WMP}, \mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$. Can we compute the winning sets Win_1^{Ω} and $\mathsf{Win}_2^{\overline{\Omega}}$? If yes what is the complexity of computing these sets and how (memoryless, finite-memory, general) are the winning strategies of both players? Given such a game (G, Ω) and an initial vertex v_0, the *synthesis problem* asks to decide whether player 1 has a winning strategy for Ω from v_0 and to build such a strategy when it exists.

[2] This objective is called "direct fixed window mean-payoff" in [6] among several other variants.

▶ **Remark 1.** (i) In this problem, we can assume that the bounds used in the vector $(\sim_1 \nu_1, \ldots, \sim_n \nu_n)$ are such that $(\nu_1, \ldots, \nu_n) = (0, \ldots, 0)$. Indeed, suppose that $\nu_m = \frac{a}{b}$ with $a \in \mathbb{Z}$ and $b \in \mathbb{N}\setminus\{0\}$, then replace the m^{th} component w_m of the weight function w by $b \cdot w_m - a$. (ii) Moreover notice that if $\nu_m = 0$ and $\Omega_m = \mathsf{WMP}$, then $\mathsf{MP}(\rho_{[k,k+l]})$ can be replaced by $\mathsf{TP}(\rho_{[k,k+l]})$ in (2). (iii) Finally, the vector $(\sim_1 0, \ldots, \sim_n 0)$ can be supposed to be equal to $(\geq 0, \ldots, \geq 0)$. Indeed strict inequality > 0 (resp. < 0) can be replaced by inequality ≥ 1 (resp. ≤ -1), and inequality ≤ 0 can be replaced by ≥ 0 by replacing the weight function by its negation and the measure Inf (resp. Sup, LimInf, LimSup, TP) by Sup (resp. Inf, LimSup, LimInf, TP).

From now on, we only work with vectors $(\geq 0, \ldots, \geq 0)$ and we no longer mention symbol $\geq$. Hence, as an example, $\mathsf{Inf}(\geq 0)$ and $\mathsf{WMP}(2, \geq 0)$ are replaced by $\mathsf{Inf}(0)$ and $\mathsf{WMP}(2, 0)$; and when the context is clear, we only mention Inf and WMP.

The games (G, Ω) of Problem 1 are determined since all the objectives Ω are ω-regular. Let us give an example where we mix quantitative objectives.

▶ **Example 1.** Consider the 3-weighted game structure of Figure 1. In all examples in this paper, we assume that circle (resp. square) vertices belong to player 1 (resp. player 2). Let $\Omega = \mathsf{WMP}(3, 0) \cap \mathsf{Sup}(0) \cap \mathsf{LimSup}(0)$ be the objective of player 1. We recall that by definition of Ω, we look at the WMP (resp. Sup, LimSup) objective on the first (resp. second, third) dimension. Let us show that v_0 is a winning vertex for player 1. Let σ_1 be the following strategy of player 1 from v_0: go to v_1, take the self loop once, go back to v_0 and then always go to v_2. Notice that $\rho \in v_0v_1v_1v_0\{v_2, v_0\}^\omega$. As player 1 forces ρ to begin with $v_0v_1v_1$, he ensures that $\mathsf{Sup}(\rho) \geq 0$ on the second component. Moreover as ρ visits infinitely often edge (v_2, v_2) or (v_2, v_0), player 1 also ensures to have $\mathsf{LimSup}(\rho) \geq 0$ on the third component. Finally, we have to check that $\rho \in \mathsf{WMP}(3, 0)$ with respect to the first component, that is (by Remark 1), for all k, there exists $l \in \{1, 2, 3\}$ such that $\mathsf{TP}(\rho_{[k,k+l]}) \geq 0$. For $k = 0$ (resp. $k = 1$, $k = 2$, $k = 3$) and $l = 3$ (resp. $l = 2$, $l = 1$, $l = 1$), we have $\mathsf{TP}(\rho_{[k,k+l]}) \geq 0$. Now, from position $k = 4$, the sum of weights is non-negative in at most 2 steps. Indeed, either player 2 takes the self loop (v_2, v_2) or he goes to v_0 where player 1 goes back to v_2. Therefore, for each $k \geq 4$, either $\rho_k = v_0$ and $\mathsf{TP}(\rho_{[k,k+l]}) \geq 0$ with $l = 1$, or $\rho_k = v_2$ and $\mathsf{TP}(\rho_{[k,k+l]}) \geq 0$ with $l = 1$ if $\rho_{k+1} = v_2$, and with $l = 2$ otherwise. It follows that $v_0 \in \mathsf{Win}_1^\Omega$. The strategy σ_1 needs memory: indeed, player 1 needs to remember if he has already visited the edge (v_1, v_1) as this is the only edge visiting a non-negative weight for the Sup objective. Finally, one can show that $\mathsf{Win}_1^\Omega = \{v_0, v_1\}$ and $\mathsf{Win}_2^{\overline{\Omega}} = \{v_2\}$.

▶ **Remark 2.** In Problem 1, the vector $(\sim_1 \nu_1, \ldots, \sim_n \nu_n)$ can be assumed equal to $(\geq 0, \ldots, \geq 0)$ by Remark 1. Thus an Inf (resp. Sup, LimInf, LimSup) objective is nothing else than a safety (resp. reachability, co-Büchi, Büchi) objective, and conversely. More precisely, every game $(G, \Omega = \cap_{m=1}^n \Omega_m)$ with each $\Omega_m \in \{\mathsf{WMP}, \mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$ can be polynomially reduced to a game $(G', \Omega' = \cap_{m=1}^n \Omega'_m)$ such that each Ω'_m belongs to $\{\mathsf{WMP}, \mathsf{Safe}, \mathsf{Reach}, \mathsf{CoBuchi}, \mathsf{Buchi}\}$, and with $|V| + |E|$ vertices and $2 \cdot |E|$ edges. This reduction is obtained by splitting each edge into two consecutive edges and decorating accordingly the new intermediate vertex to transfer the objectives. The WMP objectives in G are now WMP objectives in G' with a doubled window size. There also exists a polynomial reduction in the other direction, but without WMP objectives, and keeping the same game structure $G' = G$. These two game reductions will be used throughout this paper.

Table 2 Overview of some known results for qualitative objectives (i is the number of objectives in the intersection of reachability/Büchi objectives).

Objective	Complexity class	Algorithmic complexity	Player 1 memory	Player 2 memory
Reach/Safe [2, 13, 15]	P-complete	$O(\|V\| + \|E\|)$	memoryless	memoryless
Buchi/CoBuchi [7, 11, 15]	P-complete	$O(\|V\|^2)$	memoryless	memoryless
GenReach [12]	PSPACE-complete	$O(2^i \cdot (\|V\| + \|E\|))$	exponential memory	exponential memory
GenReach (i fixed) [12]	P-complete	$O(2^i \cdot (\|V\| + \|E\|))$	polynomial memory	polynomial memory
GenBuchi [10]	P-complete	$O(i \cdot \|V\| \cdot \|E\|)$	polynomial memory	memoryless
GenBuchi ∩ CoBuchi	P-complete	$O(i^2 \cdot \|V\| \cdot \|E\|)$	polynomial memory	memoryless

Some well-know properties

Table 2 gathers several well-known results about qualitative objectives in an unweighted game structure and Theorem 2 states the known results about the WMP objective. In this paper, the complexity of the algorithms is expressed in terms of the size $|V|$ and $|E|$ of the game structure G, the maximum weight W (in absolute value) and the dimension n of the weight function when G is weighted, the number i of objectives in an intersection of objectives[3], and the window size λ.

▶ **Theorem 2.** [6][4] *Let (G, Ω) be an n-weighted game such that $\Omega = \cap_{m=1}^{n} \Omega_m$ with $\Omega_m =$ WMP. The synthesis problem is* EXPTIME*-complete (with an algorithm in $O(\lambda^{4n} \cdot |V|^2 \cdot W^{2n})$ time), exponential memory strategies are sufficient and necessary for both players. This problem is already* EXPTIME*-hard when $n = 2$.*

When $n = 1$, the synthesis problem is decidable in $O(\lambda \cdot |V| \cdot (|V| + |E|) \cdot \lceil \log_2(\lambda \cdot W) \rceil)$ time, both players require finite-memory strategies, and memory in $O(\lambda^2 \cdot W)$ (resp. in $O(\lambda^2 \cdot W \cdot |V|)$) is sufficient for player 1 (resp. player 2). Moreover, if λ is polynomial in the size of the game, then the synthesis problem is P*-complete.*

Solution to Problem 1 and extensions

We here give the solution to Problem 1. We show that the synthesis problem is EXPTIME-complete and that exponential memory strategies are necessary and sufficient for both players:

▶ **Theorem 3.** *Let (G, Ω) be an n-weighted game such that $\Omega = \cap_{m=1}^{n} \Omega_m$ with $\Omega_m \in \{$*WMP*,* Inf*,* Sup*,* LimInf*,* LimSup*$\}$. The synthesis problem is* EXPTIME*-complete (with an algorithm in $O(|V| \cdot |E| \cdot (\lambda^2 \cdot W)^{2n})$ time), and exponential memory strategies are necessary and sufficient for both players.*

Proof. First, we show that the synthesis problem is in EXPTIME. To this end, we use an exponential reduction from [6] dealing with WMP objectives, that we adapt in a way to also deal with the Inf, Sup, LimInf and LimSup objectives. In this reduction, a game (G', Ω') with $\Omega' = \cap_{m=1}^{n} \Omega'_m$, is constructed such that for all m, $\Omega'_m \in \{$Buchi, CoBuchi$\}$, and the size of the game is exponential with $O(|V| \cdot (\lambda^2 \cdot W)^n)$ vertices and $O(|E| \cdot (\lambda^2 \cdot W)^n)$ edges. Recall that the intersection of co-Büchi objectives is a co-Büchi objective. It follows that G' is a generalized Büchi ∩ co-Büchi game. Then, by Table 2 (last item), we can compute the winning sets of both players in G in time $O(n^2 \cdot |V'| \cdot |E'|) = O(|V| \cdot |E| \cdot (\lambda^2 \cdot W)^{2n})$. Moreover,

[3] Notice that $i = n$ for the objectives considered in Problem 1.
[4] When $n = 1$, the time complexity and the memory requirements have been here correctly stated.

since polynomial memory strategies are sufficient in G', exponential memory strategies are sufficient in G. Finally, the EXPTIME-hardness and the necessity of exponential memory follow from Theorem 2. ◀

The previous theorem and proof can be generalized to Boolean combinations in DNF and CNF forms (instead of intersections) of objectives in {WMP, Inf, Sup, LimInf, LimSup}. One can assume w.l.o.g. that the dimension of the game is equal to the number of objectives that appear in the Boolean combination (by making copies of components of the weight function). The following theorem sums up the latter result, and then we discuss the general case of Boolean combinations.

▶ **Theorem 4.** *Let (G, Ω) be an nd-weighted game such that $\Omega = \cup_{k=1}^{d} \cap_{m=1}^{n} \Omega_{k,m}$ with $\Omega_{k,m} \in$ {WMP, Inf, Sup, LimInf, LimSup}. The synthesis problem is* EXPTIME*-complete (with an algorithm in $O(n^{d(d+2)} \cdot |V|^{d+1} \cdot |E| \cdot (\lambda^2 \cdot W)^{nd(d+2)})$ time), and exponential memory strategies are sufficient and necessary for both players. The same result holds when $\Omega = \cap_{k=1}^{d} \cup_{m=1}^{n} \Omega_{k,m}$.*

▶ **Remark 3.** (*i*) Whereas the synthesis problem for Boolean combinations of $\underline{\text{MP}}$ and $\overline{\text{MP}}$ is undecidable [19], it is here decidable for Boolean combinations of objectives in {WMP, Inf, Sup, LimInf, LimSup}. Furthermore, one can show that this problem is in EXPSPACE using a result of [1]. (*ii*) Note that when the number of dimensions is fixed (n in Theorem 3, nd in Theorem 4), the synthesis problem is still EXPTIME-hard by Theorem 2.

4 Efficient fragment with one WMP objective

In the previous section, we considered games (G, Ω) with $\Omega = \cap_{m=1}^{n} \Omega_m$ being any intersection of objectives in {WMP, Inf, Sup, LimInf, LimSup}. We here focus on a particular class of games in a way to achieve a lower complexity for the synthesis problem. We do not consider the case where at least two Ω_m are WMP objectives since the synthesis problem is already EXPTIME-hard in this case (by Theorem 2). We thus focus on the intersections of exactly one[5] objective WMP and any number of objectives of one kind in {Inf, Sup, LimInf, LimSup}. Note that this number must be fixed in the case of objectives Sup to avoid PSPACE-hardness in this case (see Table 2, third row). For the considered fragment, we show that the synthesis problem is P-complete for polynomial windows. The latter assumption is reasonable in practical applications where one expects a positive mean-payoff in any "short" window sliding along the play.

▶ **Theorem 5.** *Let (G, Ω) be an n-weighted game with objective $\Omega = \Omega_1 \cap \Gamma$ for player 1 such that $\Omega_1 =$ WMP and $\Gamma = \cap_{m=2}^{n} \Omega_m$ such that $\forall m\, \Omega_m =$ Inf (resp. $\forall m\, \Omega_m =$ LimInf, $\forall m\, \Omega_m =$ LimSup, $\{\forall m\, \Omega_m =$ Sup and n is fixed$\}$). Then the synthesis problem is decidable (in time polynomial in the size of the game, λ and $\lceil \log(W) \rceil$). In general, both players require finite-memory strategies, and pseudo-polynomial memory is sufficient for both players. Moreover, when λ is polynomial in the size of the game then the synthesis problem is* P*-complete.*

To prove this theorem, we use the first reduction of Remark 2 to obtain a game $(G', \Omega'_1 \cap \Gamma')$ (with $\Gamma' = \cap_{m=2}^{n} \Omega'_m$) such that $\Omega'_1 =$ WMP and $\forall m\, \Omega'_m \in$ {Reach, Safe, Buchi, CoBuchi}. Recall that the intersection of safety (resp. co-Büchi) objectives is a safety (resp. co-Büchi)

[5] The case with no WMP objective will be treated in the next section.

Table 3 Overview of the fragment (i is the number of objectives in the intersection of Reach/Büchi objectives).

Objective	Algorithmic complexity	Player 1 memory	Player 2 memory
WMP ∩ Safe	$O(\lambda \cdot \lvert V\rvert \cdot (\lvert V\rvert + \lvert E\rvert) \cdot \lceil \log(\lambda \cdot W)\rceil)$	$O(\lambda^2 \cdot W)$	$O(\lambda^2 \cdot W \cdot \lvert V\rvert)$
WMP ∩ Reach	$O(\lambda \cdot \lvert V\rvert \cdot (\lvert V\rvert + \lvert E\rvert) \cdot \lceil \log(\lambda \cdot W)\rceil)$	$O(\lambda^2 \cdot W \cdot \lvert V\rvert)$	$O(\lambda^2 \cdot W \cdot \lvert V\rvert)$
WMP ∩ GenReach (i fixed)	$O(\lambda \cdot 2^{2^i} \cdot \lvert V\rvert \cdot (\lvert V\rvert + \lvert E\rvert) \cdot \lceil \log(\lambda \cdot W)\rceil)$	$O(\lambda^2 \cdot 2^i \cdot W \cdot \lvert V\rvert)$	$O(\lambda^2 \cdot 2^i \cdot W \cdot \lvert V\rvert)$
WMP ∩ Buchi	$O(\lambda \cdot \lvert V\rvert^2 \cdot (\lvert V\rvert + \lvert E\rvert) \cdot \lceil \log(\lambda \cdot W)\rceil)$	$O(\lambda^2 \cdot W \cdot \lvert V\rvert)$	$O(\lambda^2 \cdot W \cdot \lvert V\rvert^2)$
WMP ∩ GenBuchi	$O(\lambda \cdot i^3 \cdot \lvert V\rvert^2 \cdot (\lvert V\rvert + \lvert E\rvert) \cdot \lceil \log(\lambda \cdot W)\rceil)$	$O(\lambda^2 \cdot W \cdot i \cdot \lvert V\rvert)$	$O(\lambda^2 \cdot W \cdot i^2 \cdot \lvert V\rvert^2)$
WMP ∩ CoBuchi	$O(\lambda \cdot \lvert V\rvert^2 \cdot (\lvert V\rvert + \lvert E\rvert) \cdot \lceil \log(\lambda \cdot W)\rceil)$	$O(\lambda^2 \cdot W \cdot \lvert V\rvert^2)$	$O(\lambda^2 \cdot W \cdot \lvert V\rvert)$

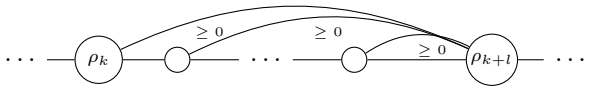

Figure 2 A λ-window at position k that is inductively-closed in $k+l$.

objective. We thus have to study WMP ∩ Safe (resp. WMP ∩ Reach, WMP ∩ GenReach (with n fixed), WMP ∩ Buchi, WMP ∩ GenBuchi, WMP ∩ CoBuchi) games. Table 3 gives an overview of the obtained properties; it indicates time polynomial in the size of the game, λ and $\lceil \log(W)\rceil$, and pseudo-polynomial[6] memories for both players. We then get Theorem 5 such that the algorithmic complexity and the memory requirements are those of Table 3 with $|V|$ replaced by $|V|+|E|$. Notice that finite memory is necessary for both players by Theorem 2. When λ is polynomial in size of the game, the complexity becomes polynomial, and the synthesis problem is P-hard (see Table 2).

Apart the objective $\Omega = \mathsf{WMP}(\lambda, 0) \cap \mathsf{Safe}(U)$, solving the games of Table 3 is difficult and requires to develop some new tools generalizing the classical concept of p-attractor while dealing with good windows (the p-attractor of a set U is the set of vertices from which player p has a strategy to reach U against any strategy of player $\overline{p}$). To this end, let us introduce some properties of windows.

Properties of windows

Let us focus on the $\mathsf{WMP}(\lambda, 0)$ objective (with TP in (2)) and introduce some terminology. Let $\rho = \rho_0\rho_1\dots$ be a play. A *λ-window at position k* is a window of size λ placed along ρ from k to $k+\lambda$. If there exists $l \in \{1, \dots, \lambda\}$ such that $\mathsf{TP}(\rho_{[k,k+l]}) \geq 0$, such a λ-window at position k is called *good* or *closed in $k+l$* (to specify index l), otherwise it is called *bad*. Moreover if l is the smallest index such that $\mathsf{TP}(\rho_{[k,k+l]}) \geq 0$, we say it is *first-closed in $k+l$*. An interesting property is the next one: a λ-window at position k is *inductively-closed in $k+l$* if it is closed in $k+l$ and for all $k' \in \{k+1, \dots, k+l-1\}$, the λ-window at position k' is also closed in $k+l$ (see Figure 2). A λ-window at position k that is first-closed in $k+l$ is inductively-closed in $k+l$. The next lemma will be useful:

▶ **Lemma 6.** *A play ρ is winning for $\mathsf{WMP}(\lambda, 0)$ iff there exists a sequence $(k_i)_{i\geq 0}$ with $k_0 = 0$ such that $\forall i, k_{i+1} - k_i \in \{1, \dots, \lambda\}$, and the λ-window at position k_i is inductively-closed in k_{i+1}.*

[6] Having pseudo-polynomial memories is not really a problem since the proofs show that the strategies can be efficiently encoded by programs using two counters, as in the case of Theorem 2.

Algorithm 1 ICWEnd

Require: 1-weighted game structure $G = (V_1, V_2, E, w)$, set $U \subseteq V$, window size $\lambda \in \mathbb{N}\setminus\{0\}$
Ensure: $\mathsf{Win}_1^{\mathsf{ICWEnd}^\lambda(U)}$

```
for all v ∈ V do
  if v ∈ U then
    C_0(v) ← 0
  else
    C_0(v) ← −∞
for all l ∈ {1, ..., λ} do
  for all v ∈ V_1 do
    C_l(v) ← max_{(v,v')∈E} {w(v, v') ⊕ max{C_0(v'), C_{l−1}(v')}}
  for all v ∈ V_2 do
    C_l(v) ← min_{(v,v')∈E} {w(v, v') ⊕ max{C_0(v'), C_{l−1}(v')}}
return {v ∈ V | C_λ(v) ≥ 0}
```

When such a sequence $(k_i)_{i\geq 0}$ exists for a play ρ, it is called a λ-*good decomposition* of ρ. We extend this notion to histories $\rho = \rho_0\rho_1\dots\rho_k$ as follows. A finite sequence $(k_i)_{i=0}^{j}$ is a λ-good decomposition of ρ if $k_0 = 0$, $k_j = k$, for each $i \in \{0,\dots j-1\}$ we have $k_{i+1} - k_i \in \{1,\dots,\lambda\}$, and the λ-window at position k_i is inductively-closed in k_{i+1}. From now on, a λ-window is simply called a window.

Two new objectives

Let us now introduce our new tools. They are based on the following new objectives.

▶ **Definition 7.** Let $G = (V_1, V_2, E, w)$ be a 1-weighted game structure, $U \subseteq V$ be a set of vertices, and $\lambda \in \mathbb{N} \setminus \{0\}$ be a window size. We consider the next two sets of plays:

- $\mathsf{ICWEnd}^\lambda(U) = \{\ \rho \in \mathsf{Plays}(G) \mid \exists l \in \{1,\dots,\lambda\}, \rho_l \in U$, and the window at position 0 is inductively-closed in $l\ \}$,
- $\mathsf{GDEnd}^\lambda(U) = \{\ \rho \in \mathsf{Plays}(G) \mid \exists l \geq 0, \rho_l \in U$ and $\rho_{[0,l]}$, has a λ-good decomposition $\}$.

Notice that the plays of $\mathsf{ICWEnd}^\lambda(U)$ are particular plays of $\mathsf{GDEnd}^\lambda(U)$. Hence we have $\mathsf{ICWEnd}^\lambda(U) \subseteq \mathsf{GDEnd}^\lambda(U)$. We propose two algorithms, Algorithms 1 and 2, one for computing the winning set of player 1 for the objective $\mathsf{ICWEnd}^\lambda(U)$, and the other for the objective $\mathsf{GDEnd}^\lambda(U)$.

Algorithm 1 uses the operator $\oplus$ defined as follows. Let $a, b \in \mathbb{Z}\cup\{-\infty\}$, then $a\oplus b = a+b$ if $a + b \geq 0$, and $-\infty$ otherwise. With this definition, either $a \oplus b \geq 0$ or $a \oplus b = -\infty$. Algorithm 1 intuitively works as follows. Given a vertex v and a number i of steps, the value $C_i(v)$ is computed iteratively (from $C_{i-1}(v)$) and represents the best total payoff that player 1 can ensure in *at most i steps* while closing the window from v in a vertex of U. Value $-\infty$ indicates that the window starting in v cannot be inductively-closed in U. The winning set of player 1 is thus the set of vertices v for which $C_\lambda(v) \geq 0$. This algorithm is inspired from Algorithm GoodWin in [6] computing $\mathsf{Win}_1^{\mathsf{ICWEnd}^\lambda(U)}$ with $U = V$.[7]

[7] We have detected a flaw in this algorithm that has been corrected in Algorithm 1. The algorithm in [6] wrongly computes the set of vertices from which player 1 can force to close the window in exactly l steps (instead of at most l steps) for some $l \in \{1,\dots,\lambda\}$.

Algorithm 2 GDEnd

Require: 1-weighted game structure $G = (V_1, V_2, E, w)$, subset $U \subseteq V$, window size $\lambda \in \mathbb{N} \setminus \{0\}$

Ensure: $\mathsf{Win}_1^{\mathsf{GDEnd}^\lambda(U)}$

1: $k \leftarrow 0$
2: $X_0 \leftarrow U$
3: **repeat**
4: $\quad X_{k+1} \leftarrow X_k \cup \mathsf{ICWEnd}(G, X_k, \lambda)$
5: $\quad k \leftarrow k + 1$
6: **until** $X_k = X_{k-1}$
7: **return** X_k

▶ **Lemma 8.** *Let G be a 1-weighted game structure, U be a subset of V, and $\lambda \in \mathbb{N} \setminus \{0\}$ be a window size. Then Algorithm* ICWEnd *computes the set $\mathsf{Win}_1^{\mathsf{ICWEnd}^\lambda(U)}$ in $O(\lambda \cdot (|V|+|E|) \cdot \lceil \log_2(\lambda \cdot W) \rceil)$ time, and finite-memory strategies with memory linear in λ are sufficient for both players.*

We now turn to Algorithm 2 for computing the winning set of player 1 when the objective is $\mathsf{GDEnd}^\lambda(U)$. It shares similarities with the classical algorithm computing the p-attractor of U while requiring to use previous Algorithm $\mathsf{ICWEnd}^\lambda(U)$.

▶ **Lemma 9.** *Let G be a 1-weighted game structure, U be a subset of V, and $\lambda \in \mathbb{N} \setminus \{0\}$ be a window size. Then Algorithm* GDEnd *computes the set $\mathsf{Win}_1^{\mathsf{GDEnd}^\lambda(U)}$ of winning vertices of player 1 for the objective $\mathsf{GDEnd}^\lambda(U)$ in $O(\lambda \cdot |V| \cdot (|V| + |E|) \cdot \lceil \log_2(\lambda \cdot W) \rceil)$ time, and finite-memory strategies with memory in $O(\lambda^2 \cdot W \cdot |V|)$ (resp. in $O(\lambda^2 \cdot W)$) are sufficient for player 1 (resp. player 2).*

The proof of Lemma 9 works as follows. Let $X^* = \cup_{k \geq 0} X_k$ be the set computed by Algorithm GDEnd. We explain why $X^* = \mathsf{Win}_1^{\mathsf{GDEnd}^\lambda(U)}$. From $v_0 \in X^*$, player 1 plays a winning strategy for the objective $\mathsf{ICWEnd}^\lambda(X_k)$, and as soon as this objective is realized he repeats such a strategy for decreasing values of k until reaching U. This strategy is winning for the objective $\mathsf{GDEnd}^\lambda(U)$ and it is finite-memory. Let us now consider $v_0 \in V \setminus X^*$. As player 2 has a winning strategy σ_2^* for the objective $\overline{\mathsf{ICWEnd}^\lambda(X^*)}$, then for $\rho = \mathsf{Out}(v_0, \sigma_1, \sigma_2^*)$ with σ_1 being any strategy of player 1, if the window at position 0 is inductively-closed in ρ_l for $l \in \{1, \ldots, \lambda\}$, then necessarily $\rho_l \in V \setminus X^*$. Therefore we propose the following strategy of player 2 from v_0: (1) If the current vertex v belongs to $V \setminus X^*$, play the winning strategy σ_2^*. (2) As soon as the window starting from v is first-closed in v' in l steps with $l \in \{1, \ldots, \lambda\}$, as $v' \in V \setminus X^*$, go back to step (1). (3) As soon as the window starting from v is bad, play whatever. This strategy is finite-memory and it is winning for player 2 since it ensures that the play cannot have a prefix with a λ-good decomposition ending in U.

▶ **Example 10.** Consider the game (G, Ω) depicted on Figure 3, where $\Omega = \mathsf{GDEnd}^2(U)$ with $U = \{v_1, v_3, v_4\}$. Let us execute Algorithm 2: $X_0 = U$, $X_1 = \mathsf{Win}_1^{\mathsf{ICWEnd}^2(X_0)} = \{v_1, v_2, v_3, v_4\}$ and then, $X_2 = \mathsf{Win}_1^{\mathsf{ICWEnd}^2(X_1)} = V$. Thus all vertices in G are winning for player 1 for the objective Ω. A winning strategy for player 1 consists in looping once in v_2 and then going to v_3. Indeed for any strategy of player 2, the outcome is either $v_0 v_1^\omega$ or $v_0 v_2 v_2 v_3 v_4^\omega$, and both outcomes admit a prefix which has a 2-good decomposition and ends with a vertex of U.

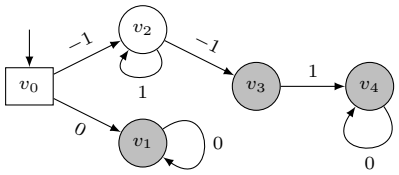

Figure 3 Objective $\mathsf{GDEnd}^2(U)$.

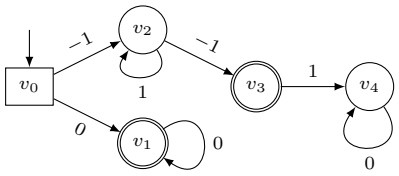

Figure 4 Objective $\mathsf{WMP}(2,0) \cap \mathsf{Reach}(U)$.

Objective WMP ∩ Reach

Now that we have introduced our new tools, let us come back to the games of Table 3, second row. Notice that the objectives $\mathsf{GDEnd}^\lambda(U)$ and $\mathsf{WMP}(\lambda,0) \cap \mathsf{Reach}(U)$ are close to each other: a play ρ belongs to $\mathsf{GDEnd}^\lambda(U)$ if it has a prefix which has a λ-good decomposition and ends with a vertex in U, while ρ belongs to $\mathsf{WMP}(\lambda,0) \cap \mathsf{Reach}(U)$ if it has a λ-good decomposition and one of its vertices belongs to U. Therefore solving games for the objective $\mathsf{WMP}(\lambda,0) \cap \mathsf{Reach}(U)$ requires to win for a modified objective $\mathsf{GDEnd}^\lambda(U)$ such that the λ-good decomposition visits U and ends in a winning vertex for the objective WMP. The following example illustrates this modified objective.

▶ **Example 11.** Consider the game (G,Ω) depicted on Figure 4, where $\Omega = \mathsf{WMP}(2,0) \cap \mathsf{Reach}(U)$ with $U = \{v_1, v_3\}$. To compute the winning set Win_1^Ω, we need to modify G in a new game structure G' where we add a bit to each vertex indicating whether U has been visited (bit equals 1) or not (bit equals 0). This game structure is the one of previous Figure 3, where the set $U' = \{v_1, v_3, v_4\}$ of gray nodes are those with bit 1. We already know that every vertex of G' is winning for objective $\mathsf{GDEnd}^2(U')$, and we note that they are also all winning for objective $\mathsf{WMP}(2,0)$. Therefore, in G', player 1 can ensure a finite 2-good decomposition ending in a vertex of U' from which he can ensure an infinite 2-good decomposition. Thanks to the added bit and coming back to G, we get that all vertices of G are winning for Ω.

Now, it is easy to solve games for the objective $\mathsf{WMP}(\lambda,0) \cap \mathsf{GenReach}(U_1,\dots,U_{i-1})$ when i is fixed (see Table 3, third row). From such a game, we construct a new game structure where we add i bits to each vertex: the j^{th} bit equals 1 if U_j has been visited, and 0 otherwise. Note that i has to be fixed to get a polynomial construction, otherwise the problem is already PSPACE-hard by Table 2 (third row). Finally, we solve the new game where the objective is $\mathsf{WMP}(\lambda,0) \cap \mathsf{GenReach}(U')$ where U' is the set of vertices with all bits equal to 1.

Objective WMP ∩ Buchi

Solving games for the objective $\mathsf{WMP} \cap \mathsf{Buchi}$ needs an algorithm that repeatedly uses the algorithm for the objective $\mathsf{WMP} \cap \mathsf{Reach}$.

▶ **Proposition 1.** *Let $G = (V_1, V_2, E, w)$ be a 1-weighted game structure, and Ω be the objective $\mathsf{WMP}(\lambda, 0) \cap \mathsf{Buchi}(U)$. Then Win_1^{Ω} can be computed in $O(\lambda \cdot |V|^2 \cdot (|V| + |E|) \cdot \lceil \log_2(\lambda \cdot W) \rceil)$ time and finite-memory strategies with memory in $O(\lambda^2 \cdot W \cdot |V|)$ (resp. in $O(\lambda^2 \cdot W \cdot |V|^2)$) are sufficient for player 1 (resp. player 2).*

The algorithm for the objective $\mathsf{WMP} \cap \mathsf{Buchi}$ works as follows: *(1)* Compute the winning set X for player 1 for the objective $\mathsf{WMP}(\lambda, 0) \cap \mathsf{Reach}(U)$ in G; *(2)* compute the 2-attractor Y of the set $V \setminus X$ in G; *(3)* repeat step (1) in the game $G[V \setminus Y]$ and step (2) in the game G, until X is empty or X is the set of all vertices in $G[V \setminus Y]$. The final set X is the winning set of player 1 for the objective $\mathsf{WMP}(\lambda, 0) \cap \mathsf{Buchi}(U)$.

We first explain why $X \subseteq \mathsf{Win}_1^{\mathsf{WMP}(\lambda,0) \cap \mathsf{Buchi}(U)}$. Notice that this algorithm exactly computes the set X of vertices such that player 1 wins for the objective $\mathsf{WMP}(\lambda, 0) \cap \mathsf{Reach}(U)$ while staying in X (player 2 cannot leave this set). Therefore, from $v_0 \in X$, player 1 plays a winning strategy for the objective $\mathsf{WMP}(\lambda, 0) \cap \mathsf{Reach}(U)$, and as soon as a vertex of U has been visited and the current history has a λ-good decomposition ending in a vertex $v \in X$, he repeats such a strategy, ad infinitum. This strategy is winning for the objective $\mathsf{WMP}(\lambda, 0) \cap \mathsf{Buchi}(U)$ and it is finite-memory. Now, we show that $\mathsf{Win}_1^{\mathsf{WMP}(\lambda,0) \cap \mathsf{Buchi}(U)} \subseteq X$. If player 1 can win for the objective $\mathsf{WMP}(\lambda, 0) \cap \mathsf{Buchi}(U)$, then a winning strategy ensures that he only visits vertices of $\mathsf{Win}_1^{\mathsf{WMP}(\lambda,0) \cap \mathsf{Reach}(U)}$. Then, player 1 has a strategy to win the $\mathsf{WMP}(\lambda, 0) \cap \mathsf{Reach}(U)$ objective using only vertices from which this property is ensured, which shows that $v_0 \in X$.

Concerning the two last rows of Table 3, one can derive an algorithm for the objective $\mathsf{WMP} \cap \mathsf{GenBuchi}$ from the algorithm for the objective $\mathsf{WMP} \cap \mathsf{Buchi}$; the objective $\mathsf{WMP} \cap \mathsf{CoBuchi}$ is the most difficult to solve and requires elaborated arguments to manage correctly two nested fixpoints together with the good windows.

5 Intersection of objectives in {Inf,Sup,LimInf,LimSup}

The aim of this section is to provide a refinement of Theorem 3 for games $(G, \Omega = \cap_{m=1}^{n} \Omega_m)$ when no objective Ω_m is a WMP objective. In this case, we get the better complexity of PSPACE-completeness (instead of EXPTIME-completeness) for the synthesis problem; nevertheless the two players still need exponential memory strategies to win (Theorem 12). We also study with precision (in Table 4) the complexity and the memory requirements in terms of the objectives of $\{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$ that appear in the intersection $\Omega = \cap_{m=1}^{n} \Omega_m$. When there is at most one Sup, we get a polynomial fragment and in certain cases, players can play memoryless. Notice that the membership to PSPACE in Theorem 12 could have been obtained from one result proved in [1] (in this paper, the objective is defined by an LTL formula, and it is proved that deciding the winner is in PSPACE for Boolean combinations of formulas of the form "eventually p" and "infinitely often p"). We here propose a simple proof adapted to our context, that allows an easy study of the winning strategies as well as the identification of polynomial fragments.

▶ **Theorem 12.** *Let (G, Ω) be an n-weighted game such that $\Omega = \cap_{m=1}^{n} \Omega_m$ with $\Omega_m \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$. The synthesis problem is PSPACE-complete (with an algorithm in $O(2^n \cdot (|V| + |E|))$ time) and exponential memory strategies are necessary and sufficient for both players.*

In Table 4, we recall Theorem 12 (first row) and exhibit several polynomial refinements (next rows). As shown by Example 13, these additional results are optimal with respect to the required memory (no/finite memory) for the winning strategies.

Table 4 Overview of properties for the intersection of objectives in {Inf, Sup, LimInf, LimSup}.

Inf	Sup	LimInf	LimSup	Complexity class	Algorithmic complexity	Player 1 memory	Player 2 memory
any	any	any	any	PSPACE-complete	$O(2^n \cdot (\|V\| + \|E\|))$	exponential memory	exponential memory
any	≤ 1	any	any	P-complete	$O(n^2 \cdot (\|V\| + \|E\|) \cdot \|E\|)$	polynomial memory	memoryless
any	0	any	≤ 1	P-complete	$O((\|V\| + \|E\|) \cdot \|E\|)$	memoryless	memoryless
any	1	0	0	P-complete	$O(\|V\| + \|E\|)$	memoryless	memoryless

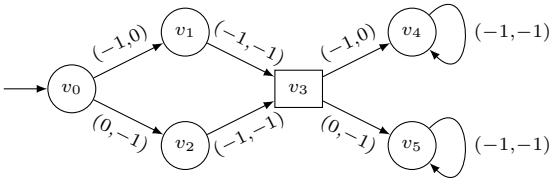

Figure 5 Example where player 2 needs memory.

▶ **Example 13.** First, we come back to the game structure G depicted on Figure 1, where we only keep the second and the third dimensions. Assume $\Omega = \mathsf{Sup}(0) \cap \mathsf{LimSup}(0)$. Then, v_0 is winning for player 1 but memory is required to remember if player 1 has visited the edge (v_1, v_1). The same argument holds for $\Omega = \mathsf{Sup}(0) \cap \mathsf{LimInf}(0)$ and $\Omega = \mathsf{Sup}(0) \cap \mathsf{Sup}(0)$. This example with objective $\Omega = \mathsf{Sup}(0) \cap \mathsf{LimSup}(0)$ indicates that player 1 cannot win memoryless in a game as in the second row of Table 4. This example with objective $\Omega = \mathsf{Sup}(0) \cap \mathsf{LimSup}(0)$ (resp. $\Omega = \mathsf{Sup}(0) \cap \mathsf{LimInf}(0)$, $\Omega = \mathsf{Sup}(0) \cap \mathsf{Sup}(0)$) also shows that player 1 needs memory to win if $[1, 0, 0]$ (referring to the second, third and fourth columns of Table 4) in the last row of Table 4 is replaced by $[1, 0, 1]$ (resp. $[1, 1, 0]$, $[2, 0, 0]$). Now, assume that $v_2 \in V_1$, that is, G is a one-player game and let $\Omega = \mathsf{LimSup}(0) \cap \mathsf{LimSup}(0)$. Again, v_0 is winning but player 1 needs memory since he has to alternate between v_1 (and take the self loop) and v_2. This shows that in the third row of Table 4, if $[\leq 1]$ is replaced by $[2]$ then player 1 needs memory to win. Finally, consider the game depicted on Figure 5. Let $\Omega = \mathsf{Sup}(0) \cap \mathsf{Sup}(0)$. Vertex v_0 is losing for player 1 (i.e. winning for player 2), but player 2 needs memory since he has to know which edge player 1 took from v_0 to counter him by taking the edge with the same vector of weights from v_3. This shows that in the second row of Table 4, if $[\leq 1]$ is replaced by $[2]$ then player 2 needs memory.

▶ **Remark 4.** When n is fixed, the synthesis problem becomes P-complete for games (G, Ω) such that $\Omega = \cap_{m=1}^{n} \Omega_m$ with $\Omega_m \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$. The P-easyness follows from Theorem 12 and the P-hardness follows from Table 2 and the second reduction of Remark 2.

Acknowledgments. We would like to thank Mickael Randour for his availability and help.

References

1 R. Alur, S. La Torre, and P. Madhusudan. Playing games with boxes and diamonds. In *CONCUR 2003*, volume 2761 of *LNCS*, pages 127–141. Springer, 2003.

2 C. Beeri. On the membership problem for functional and multivalued dependencies in relational databases. *ACM Trans. Database Syst.*, 5(3):241–259, 1980.

3 P. Bouyer, U. Fahrenberg, K. G. Larsen, N. Markey, and J. Srba. Infinite runs in weighted timed automata with energy constraints. In *FORMATS 2008*, volume 5215 of *LNCS*, pages 33–47. Springer, 2008.

4 L. Brim, J. Chaloupka, L. Doyen, R. Gentilini, and J.-F. Raskin. Faster algorithms for mean-payoff games. *Formal Methods in System Design*, 38(2):97–118, 2011.

5 K. Chatterjee, L. Doyen, T. A. Henzinger, and J.-F. Raskin. Generalized mean-payoff and energy games. In *FSTTCS 2010*, volume 8 of *LIPIcs*, pages 505–516, 2010.

6 K. Chatterjee, L. Doyen, M. Randour, and J.-F. Raskin. Looking at mean-payoff and total-payoff through windows. *Inf. Comput.*, 242:25–52, 2015.

7 K. Chatterjee and M. Henzinger. Efficient and dynamic algorithms for alternating büchi games and maximal end-component decomposition. *J. ACM*, 61(3):15:1–15:40, 2014.

8 K. Chatterjee, M. Randour, and J.-F. Raskin. Strategy synthesis for multi-dimensional quantitative objectives. *Acta Inf.*, 51(3-4):129–163, 2014.

9 A. Degorre, L. Doyen, R. Gentilini, J.-F. Raskin, and S. Toruńczyk. Energy and mean-payoff games with imperfect information. In *CSL 2010*, volume 6247 of *LNCS*, pages 260–274. Springer, 2010.

10 S. Dziembowski, M. Jurdzinski, and I. Walukiewicz. How much memory is needed to win infinite games? In *LICS 1997*, pages 99–110. IEEE Computer Society, 1997.

11 E. Allen Emerson and C. S. Jutla. Tree automata, mu-calculus and determinacy (extended abstract). In *FOCS 1991*, pages 368–377. IEEE Computer Society, 1991.

12 N. Fijalkow and F. Horn. Les jeux d'accessibilité généralisée. *Technique et Science Informatiques*, 32(9-10):931–949, 2013.

13 E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *LNCS*. Springer, 2002.

14 P. Hunter, G. A. Pérez, and J.-F. Raskin. Looking at mean-payoff through foggy windows. In *ATVA 2015*, volume 9364 of *LNCS*, pages 429–445. Springer, 2015.

15 N. Immerman. Number of quantifiers is better than number of tape cells. *J. Comput. Syst. Sci.*, 22(3):384–406, 1981.

16 M. Jurdzinski, R. Lazic, and S. Schmitz. Fixed-dimensional energy games are in pseudo-polynomial time. In *ICALP 2015*, volume 9135 of *LNCS*, pages 260–272. Springer, 2015.

17 D. A. Martin. Borel determinacy. *Annals of Mathematics*, 102(2):363–371, 1975.

18 A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *ACM 1989*, pages 179–190. ACM Press, 1989.

19 Y. Velner. Robust multidimensional mean-payoff games are undecidable. In *FoSSaCS 2015*, volume 9034 of *LNCS*, pages 312–327. Springer, 2015.

20 Y. Velner, K. Chatterjee, L. Doyen, T. A. Henzinger, A. M. Rabinovich, and J.-F. Raskin. The complexity of multi-mean-payoff and multi-energy games. *Inf. Comput.*, 241:177–196, 2015.

21 Y. Velner and A. Rabinovich. Church synthesis problem for noisy input. In *FOSSACS 2011*, volume 6604 of *LNCS*, pages 275–289. Springer, 2011.

22 U. Zwick and M. Paterson. The complexity of mean payoff games on graphs. *Theor. Comput. Sci.*, 158(1&2):343–359, 1996.