Proceedings of the 13th European Conference on Combinatorics, Graph Theory and Applications
EUROCOMB'25
Budapest, August 25 - 29, 2025

A UNIFIED CONVENTION FOR ACHIEVEMENT POSITIONAL GAMES

(EXTENDED ABSTRACT)

Florian Galliot*   Jonas Sénizergues†

Abstract

We introduce achievement positional games, a convention for positional games which encompasses the Maker-Maker and Maker-Breaker conventions. We consider two hypergraphs, one red and one blue, on the same vertex set. Two players, Left and Right, take turns picking a previously unpicked vertex. Whoever first fills an edge of their color, blue for Left or red for Right, wins the game (draws are possible). We establish general properties of such games. In particular, we show that a lot of principles which hold for Maker-Maker games generalize to achievement positional games. We also study the algorithmic complexity of deciding whether Left has a winning strategy as first player when all blue edges have size at mot p and all red edges have size at most q. This problem is in P for $p, q \leq 2$, but it is NP-hard for $p \geq 3$ and $q = 2$, coNP-complete for $p = 2$ and $q \geq 3$, and PSPACE-complete for $p, q \geq 3$. A consequence of this last result is that, in the Maker-Maker convention, deciding whether the first player has a winning strategy on a hypergraph of rank 4 after one round of (non-optimal) play is PSPACE-complete. A full version of this paper is available at [6].

1 Introduction

Positional games. *Positional games* have been introduced by Hales and Jewett [7] and later popularized by Erdős and Selfridge [3]. The game board is a hypergraph $H = (V, E)$, where V is the vertex set and $E \subseteq 2^V$ is the edge set. Two players take turns picking a previously unpicked vertex of the hypergraph, and the result of the game is defined by some *convention.* The two most popular conventions are called *Maker-Maker* and *Maker-Breaker.* As they revolve around trying to fill an edge *i.e.* pick all the vertices of some edge, they are often referred to as "achievement games". In the Maker-Maker convention, whoever first fills an edge wins (draws are possible), whereas in the *Maker-Breaker* convention, Maker aims at filling an edge while Breaker aims at preventing him from doing so (no draw is possible). For

*Aix-Marseille Université, CNRS, Centrale Marseille, I2M, UMR 7373, 13453 Marseille, France. E-mail : florian.galliot@univ-amu.fr
†Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, F-33400 Talence, France. E-mail : jonas.senizergues@u-bordeaux.fr

all conventions, the main question is the result of the game with optimal play (who wins, or is it a draw), including the complexity of the associated algorithmic problems.

The Maker-Maker convention was the first one to be introduced, in 1963 by Hales and Jewett [7]. The game of *Tic-Tac-Toe* is a famous example. As a simple *strategy-stealing* argument [7] shows that the second player cannot have a winning strategy, the question is whether the given hypergraph H is a first player win or a draw with optimal play. This decision problem is trivially tractable for hypergraphs of rank 2 *i.e.* whose edges have size at most 2, but it is PSPACE-complete for hypergraphs that are 6-uniform *i.e.* whose edges have size exactly 6 (by a combination of [8] and [1]). Maker-Maker games are notoriously difficult to handle since both players must manage offense and defense at the same time.

The Maker-Breaker convention was introduced for that reason, in 1978 by Chvátal and Erdős [2]. It is by far the most studied, as it presents some convenient additional properties compared to the Maker-Maker convention thanks to the players having complementary goals, the most crucial one being subhypergraph monotonicity. The problem of deciding which player has a winning strategy when, say, Maker starts, is tractable for hypergraphs of rank 3 [5] but PSPACE-complete for 5-uniform hypergraphs [8] (a very recent improvement on the previously known result for 6-uniform hypergraphs [10]).

Unified achievement games. We introduce *achievement positional games*. Such a game is a triple $\mathcal{G} = (V, E_L, E_R)$, where (V, E_L) and (V, E_R) are hypergraphs which we see as having *blue edges* and *red edges* respectively. There are two players, taking turns picking a previously unpicked vertex: Left aims at filling a blue edge, while Right aims at filling a red edge. Whoever reaches their goal first wins the game, or we get a draw if this never happens. Achievement positional games include all Maker-Maker and Maker-Breaker games. Indeed, Maker-Maker games correspond to the case $E_L = E_R$, while Maker-Breaker games correspond to the case $E_R = \varnothing$ (or $E_L = \varnothing$) or equivalently to the case where each of E_L and E_R is the set of all minimal transversals of the other (a *transversal* of a set of edges E is a set of vertices that intersects each element of E). That last interpretation of Maker-Breaker games puts their asymmetrical nature into question, which is another motivation for the introduction of a unifying convention.

Objectives and results. We first establish elementary properties of achievement positional games in general. In particular, we look at some general principles which hold in the Maker-Maker convention to see if they generalize to achievement positional games. For all those that we consider, we show that this is indeed the case, emphasizing the fact that most properties of Maker-Maker games come from their "achievement" nature rather than symmetry. Our second objective is the study of the algorithmic complexity of the game. We get results for almost all edge sizes, which are summed up in Table 1. As a corollary, we also show that deciding whether the next player has a winning strategy for the Maker-Maker game on a hypergraph of rank 4 after one round of (non-optimal) play is PSPACE-complete.

2 Preliminaries

In this paper, a *hypergraph* is a pair (V, E) where V is a finite *vertex set* and $E \subseteq 2^V \setminus \{\varnothing\}$ is the *edge set*. An *achievement positional game* is a triple $\mathcal{G} = (V, E_L, E_R)$ where (V, E_L) and (V, E_R) are hypergraphs. The elements of E_L and E_R are called *blue edges* and *red edges* respectively. Two players, Left and Right, take turns picking a vertex in V that has not been picked before. We say a player *fills* an edge if that player has picked all the vertices of that

q \ p	0 , 1	2	3	4	5+
0 , 1	LSPACE [trivial]	LSPACE [9]	P [5]	open	PSPACE-c [8]
2	LSPACE [trivial]	P [Th. 4.1]	NP-hard [Th. 4.3]	NP-hard [Th. 4.3]	PSPACE-c [8]
3+	LSPACE [trivial]	coNP-c [Th. 4.2]	PSPACE-c [Th. 4.4]	PSPACE-c [Th. 4.4]	PSPACE-c [8]

Table 1: Algorithmic complexity of deciding whether Left has a winning strategy as first player, for blue edges of size at most p and red edges of size at most q.

edge. The blue and red edges can be seen as the winning sets of Left and Right respectively, so that the result of the game is determined as follows:

- If Left fills a blue edge before Right fills a red edge, then Left wins.
- If Right fills a red edge before Left fills a blue edge, then Right wins.
- If none of the above happens before all vertices are picked, then the game is a draw.

For algorithmic considerations, we introduce the problem AchievementPos(p,q) which consists in deciding, given an achievement positional game $\mathcal{G}$ such that all blue edges have size at most p and all red edges have size at most q, whether Left has a winning strategy on $\mathcal{G}$ as first player.

Like all positional games, AchievementPos(p,q) is in PSPACE as the game cannot last more than $|V|$ moves. We can also notice that, for all k, AchievementPos(0,k) and AchievementPos(1,k) are trivial problems. Meanwhile, for all k, AchievementPos(k,0) and AchievementPos(k,1) are equivalent to the Maker-Breaker game played on hypergraphs of rank k, so the literature provides some results.

Proposition 2.1. *AchievementPos(k,0) and AchievementPos(k,1) are in* LSPACE *for $k \geq 2$, in* P *for $k = 3$, but are* PSPACE*-complete for $k \geq 5$. Moreover, AchievementPos(0,k) and AchievementPos(1,k) are in* LSPACE *for all k.*

3 General results

A lot of convenient properties of Maker-Maker games generalize to achievement positional games. For instance, this is the case for the well-known *strategy-stealing* argument [7] which ensures that the second player can never have a winning strategy in the Maker-Maker convention.

Lemma 3.1 (Strategy Stealing)**.** *Let $\mathcal{G} = (V, E_L, E_R)$ be an achievement positional game. If there exists a bijection $\sigma : V \to V$ such that $\sigma(e) \in E_L$ and $\sigma^{-1}(e) \in E_L$ for all $e \in E_R$, then Left has a non-losing strategy on $\mathcal{G}$ as first player.*

Pairing strategies are an important tool in both Maker-Breaker and Maker-Maker conventions [7]. A *complete pairing* of a hypergraph H is a set Π of pairwise disjoint pairs of vertices such that every edge of H contains some element of Π. If H admits a complete pairing, then the outcome is a Breaker win for the Maker-Breaker game or a draw for the Maker-Maker game, as picking one vertex from each pair prevents the other player from filling an edge. We observe that, in general achievement positional games, pairing strategies may still be used as non-losing strategies which block the opponent.

Let us also mention the following monotonicity property: adding or shrinking blue edges cannot harm Left, and adding or shrinking red edges cannot harm Right.

4 Complexity results

Theorem 4.1. AchievementPos(2,2) *is in* P.

Sketch of the proof. After a series of forced moves, we get a situation where all edges have size exactly 2. At this point, the player who is next to play can be assumed not to have a P_3 (path on 3 vertices) of their color, as they would have a winning strategy in two moves. In particular, assume that Right is next to play, otherwise Right has a non-losing pairing strategy. Right must force all of Left's moves until she has broken every blue P_3. Any move u by Right triggers a sequence of forced moves, corresponding to an alternating red-blue path $P(u)$ which is easy to compute. If $P(u)$ ends with a red edge for some u, then we can assume that Right picks u and all forced moves along $P(u)$ are played, as Right keeps the initiative. However, if $P(u)$ ends with a blue edge for all u, then Right avoids a loss if and only if she can trigger one last sequence of forced moves after which every blue P_3 is broken. □

Theorem 4.2. AchievementPos(2,3) *is* coNP*-complete.*

Proof. We consider the complement of this problem, or rather an equivalent version of it. We show that it is NP-complete to decide whether Left has a non-losing strategy as first player on an achievement positional game where all blue (resp. red) edges have size at most 3 (resp. at most 2).

Let us first show membership in NP. Consider the following strategy $\mathcal{S}$ for Right: pick some u that wins in one move if such u exists, otherwise pick some v that prevents Left from winning in one move if such v exists, otherwise pick some w at the center of an intact red P_3 is one exists, otherwise (Left has a non-losing pairing strategy) pick an arbitrary vertex. Clearly, if Right has a winning strategy on $\mathcal{G}$ as second player, then $\mathcal{S}$ is one. Moreover, the move prescribed by $\mathcal{S}$ in any given situation is easily computed in polynomial time. Therefore, a polynomial certificate for Left's non-losing strategy is simply the sequence of all of Left's moves, assuming that Right plays according to $\mathcal{S}$.

We now reduce 3-SAT to our problem. Consider an instance ϕ of 3-SAT, with a set of variables V and a set of clauses C. We build a game $\mathcal{G} = (V, E_L, E_R)$ as follows (see Figure 1 for a visual example):

- For all $x \in V$, we define $V_x = \{x, \neg x\}$.
- For all $c = \ell_1 \vee \ell_2 \vee \ell_3 \in C$, with literals ℓ_1, ℓ_2, ℓ_3, we define $V_c = \left\{c_{\ell_1}, c_{\ell_2}, c_{\ell_3}, c'_{\ell_1}, c'_{\ell_2}, c'_{\ell_3}\right\}$.
- $V = \bigcup_{x \in V} V_x \cup \bigcup_{c \in C} V_c \cup \{\omega, \check{\omega}, \hat{\omega}\}$.
- $E_L = \bigcup_{x \in V} \{\{x, \neg x\}\} \cup \bigcup_{c=\ell_1 \vee \ell_2 \vee \ell_3 \in C} \left\{\left\{\ell_1, c_{\ell_1}, c'_{\ell_1}\right\}, \left\{\ell_2, c_{\ell_2}, c'_{\ell_2}\right\}, \left\{\ell_3, c_{\ell_3}, c'_{\ell_3}\right\}\right\}$.
- $E_R = \bigcup_{c=\ell_1 \vee \ell_2 \vee \ell_3 \in C} \{\{c_{\ell_1}, c_{\ell_2}\}, \{c_{\ell_2}, c_{\ell_3}\}, \{c_{\ell_3}, c_{\ell_1}\}\} \cup \{\{\omega, \check{\omega}\}, \{\omega, \hat{\omega}\}\}$.

Since there are multiple pairwise vertex-disjoint red P_3's, every move from Left must threaten to win on the next move until the last red P_3 is broken.

As such, Left must start by picking $\ell \in \{x, \neg x\}$ for some $x \in V$, which forces Right to pick the other one since $\{x, \neg x\} \in E_R$. After that, it can easily be shown that it is optimal for Left to pick c_ℓ for each clause c which contains the literal ℓ, as it forces Right to pick c'_ℓ in response since $\{\ell, c_\ell, c'_\ell\} \in E_R$. This breaks every red P_3 in the clause gadgets corresponding to clauses containing ℓ.

Left must repeat this process of picking a literal and then breaking all clause gadgets of clauses containing that literal, until he has picked at least one of c_{ℓ_1}, c_{ℓ_2} or c_{ℓ_3} for each clause $c = \ell_1 \vee \ell_2 \vee \ell_3 \in C$.

If there exists a valuation μ which satisfies ϕ, then Left succeeds in doing so, by picking x if $\mu(x) = \mathsf{T}$ or $\neg x$ if $\mu(x) = \mathsf{F}$, for all $x \in V$. After that, he can simply pick ω, thus ensuring not to lose the game. If such a valuation does not exist, then Left will have to play a move that does not force Right's answer while leaving at least one red P_3 intact, thus losing the game. All in all, Left has a non-losing strategy on $\mathcal{G}$ as first player if and only if ϕ is satisfiable, which concludes the proof. □

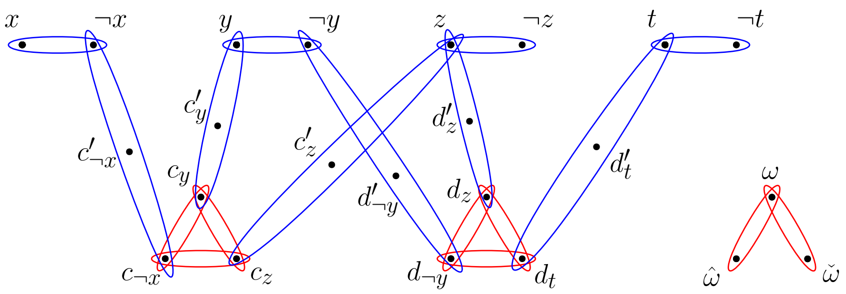

Figure 1: The full gadget from the proof of Theorem 4.2 for a set of two clauses $c = \neg x \vee y \vee z$ and $d = \neg y \vee z \vee t$.

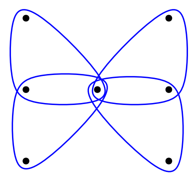

Figure 2: A blue butterfly.

Theorem 4.3. AchievementPos(3,2) *is* NP*-hard.*

Proof. We use the same construction as in the previous proof (see Figure 1), except we add two copies of the blue butterfly gadget (see Figure 2) to transform draws into wins for Left. Indeed, the game unfolds in the same way until/if Left breaks the last red P_3, after which Left will be able to play the first move in one of the butterflies to win the game. □

Theorem 4.4. AchievementPos(3,3) *is* PSPACE*-complete.*

Sketch of the proof. We perform a reduction from 3-QBF[11]. The input is a logic formula ϕ in CNF form, with clauses of size exactly 3 and variables $x_1, \ldots, x_{2n}$. Two players, Satisfier and Falsifier, take turns setting the variables $x_1, \ldots, x_{2n}$ (in that order) to T or F. Satisfier starts the game, and wins if ϕ is satisfied, otherwise Falsifier wins. Given ϕ, we build an instance $\mathcal{G}$ of AchievementPos(3,3) such that Left has a winning strategy on $\mathcal{G}$ as second player if and only if Falsifier has a winning strategy on ϕ.

The variable gadget associated to x_1 is pictured in Figure 3. Its edges of size 2 are the only ones in the entire game, so Right is forced to pick either $t_{1,R}$ (interpreted as $\mu(x_1) = \mathsf{T}$) or $f_{1,R}$ ($\mu(x_1) = \mathsf{F}$). This triggers a forced sequence of moves on the four vertices at the top. These moves update the variable gadget associated to x_2, which becomes as pictured in Figure 4. Left is forced to pick either $t_{2,L}$ ($\mu(x_2) = \mathsf{T}$) or $f_{2,L}$ ($\mu(x_2) = \mathsf{F}$), etc. until all variable gadgets have been played in. This marks the end of Phase 1, with half the vertices in $U = \bigcup_{1 \leq i \leq 2n} \{t_{i,R}, f_{i,R}\}$ having been picked by Right (by genuine choice for odd i, by forced choice for even i). The (blue) clause-edges are defined using U and parities, *e.g.* a clause $c = x_1 \vee x_2 \vee \neg x_3$ yields a blue edge $\{t_{1,R}, f_{2,R}, f_{3,R}\}$. During Phase 2, we use blue butterflies to allow Left to pick the remaining half of U, thus filling a clause-edge if and only if μ does not satisfy ϕ. □

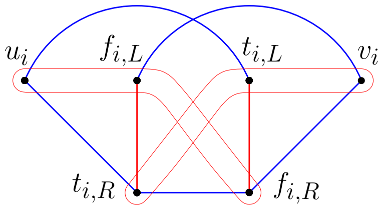

Figure 3: The variable gadget for odd i.

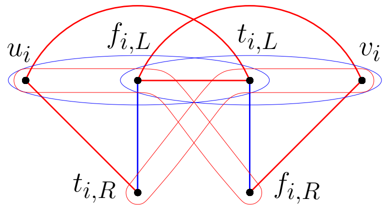

Figure 4: The variable gadget for even i.

Corollary 4.5. *Deciding whether the first player has a winning strategy for the Maker-Maker game on a hypergraph of rank 4 with one round having already been played is* PSPACE*-complete.*

Proof. Let $\mathcal{G}$ be an instance of AchievementPos(3,3). We add a vertex u_L to each $e \in E_L$ and a vertex u_R to each $e \in E_R$, then we forget about the colors. We get a hypergraph H such that, after one round of the Maker-Maker game on H where the players pick u_L and u_R respectively, we get precisely $\mathcal{G}$. □

5 Conclusion

We have introduced achievement positional games, a new convention for positional games where the players try to fill different edges. We have established some of their general properties (see [6] for all the results), which are not any weaker compared to the subfamily of Maker-Maker games, and obtained complexity results for almost all edge sizes. A corollary is that the Maker-Maker convention is PSPACE-complete for positions that can be obtained from a hypergraph of rank 4 after just one round of play, which is the first known complexity result on this convention for edges of size between 3 and 5.

We have not determined the exact complexity of the cases $(p, q) \in \{(3, 2), (4, 2)\}$, even though we know they are NP-hard. The commonplace intuition within the community is that Maker-Breaker games on hypergraphs of rank 4 are PSPACE-complete, which would imply that AchievementPos(4,2) also is. As for AchievementPos(3,2), a proof of either membership in NP or PSPACE-hardness would be compelling.

A natural prospect would be to define avoidance positional games, where whoever first fills an edge of their color loses. Since the case $E_L = E_R$ (*Avoider-Avoider convention*) is already PSPACE-complete for edges of size 2 [4], the complexity aspects would not be as interesting. However, an analogous study to that of Section 3 could be performed to better understand general properties of avoidance games.

Acknowledgments

We thank Sylvain Gravier for inspiring the topic of this paper and providing useful feedback. This research was partly supported by the ANR project P-GASE (ANR-21-CE48-0001-01).