Navigating in a Space of Game Views (extended abstract)

JAAMAS Track

Michael P. Wellman
University of Michigan
Ann Arbor, USA
wellman@umich.edu

Katherine Mayo
University of Michigan
Ann Arbor, USA
kamayo@umich.edu

ABSTRACT

We motivate and summarize a perspective on game-theoretic reasoning as navigating in a space of game views. The full journal version appears in *Autonomous Agents and Multi-Agent Systems* [5].

KEYWORDS

Game-theoretic reasoning

ACM Reference Format:
Michael P. Wellman and Katherine Mayo. 2025. Navigating in a Space of Game Views (extended abstract): JAAMAS Track. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 2 pages.

Advances in game-theoretic reasoning have greatly expanded the scale and scope of multiagent systems subject to the tools of game theory. The implicit perspective in game reasoning is that there is a central object: a *game*, to be solved. This is a convenient fiction. In reality, agents that we want to reason about live in a strategic environment extended in time, potentially influenced by actions of myriad other agents they or we might not even know about. Practically, to make decisions, we must draw some lines and bound attention—temporally, spatially, socially, topically—focusing on the most salient strategic interactions and trusting that the rest are sufficiently unimportant or independent to ignore in the operative context. We might also choose to cover broad swaths of interactions in an aggregated or abstracted way, trading fidelity for tractability.

Such modeling judgments are inescapable. For starters:

- *how far* in the future must an agent plan and reason about behaviors and reactions of others,
- *which others* should be considered, and
- *at what level* of detail?

An ambitious AI developer naturally wishes to make modeling choices explicit and place them under algorithmic control. In some contexts, it is possible to assess tradeoffs quantitatively, for example theoretical bounds on errors due to forms of game abstraction [4]. Gilpin and Sandholm [1] used integer programming to select an optimal abstraction scheme for poker game trees, given complexity constraints. More generally, evaluating tradeoffs to construct an optimal game-theoretic model is beyond current capabilities.

In contrast, generating candidate game models within a defined space is relatively straightforward. Systematic exploration in this model space can be informative, even if we cannot determine the "best" model. Different models make different tradeoffs, and the weaknesses of one model may line up with strengths of another. Considering an *ensemble* of models is one way to cover many perspectives, even when a single model for their union is intractable.

One might even aspire to control a strategic agent with game-theoretic reasoning, iteratively scoping decision problems and generating models with corresponding scopes. The model ensemble continually adapts based on experience and change of circumstance.

As a first step, we propose and start to develop a conceptual framework for sequential formulation of game-theoretic models. The key object is what we call a *game view*, which encapsulates a specific set of modeling choices about which aspects of a strategic situation to focus on in a particular analysis instance. Game-theoretic reasoning overall comprises a series of these instances, through a process we term ***game view navigation***.

Navigating among game views attempts to glean insights about a strategic scenario from multiple perspectives, varying scale and emphasis of different elements. Viewing everything all at once is not generally feasible, even with today's impressive game reasoning toolkit. This is especially true *a priori* when we lack confidence about which of the potentially relevant elements are actually salient.

The framework also serves to describe many existing methods in game reasoning. Some were expressly conceived in these terms, for example iterative abstraction of action spaces [3]. Others have not been explicitly described in this way, yet they fit the framework quite well.

Our key contribution, therefore, is explicating game view navigation as a unifying framework for iterative game reasoning. We show how numerous and diverse methods can be cast in this framework, drawing interesting connections in the process. We follow by applying the framework for new method design, based on modulating levels of aggregation to guide equilibrium search.

1 CONCEPTUAL FRAMEWORK

Fig. 1 presents a high-level diagram of the conceptual framework of game view navigation. ***Game knowledge*** is the underlying source of information about a game. This might include specifications in any well-defined representation language, as well as databases or simulators of various forms. The corpus of knowledge is considered static in this framework, though it may be elaborated or otherwise change form during navigation as a result of reasoning.

For example, in poker, game knowledge comprises rules for how the game is played, including how the cards are dealt out and how betting works. In the course of reasoning, all or part of the game tree could be worked out in a more explicit form, but that would not represent an actual change in game knowledge.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

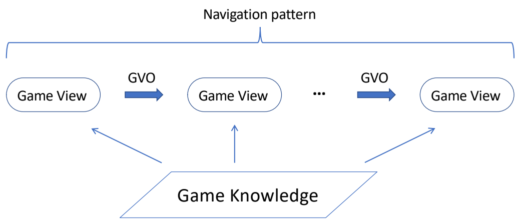

Figure 1: Game view navigation: A sequence of game views generated from game knowledge by game view operations.

A ***game view*** is a game model intended to capture key elements of a complex strategic situation. Models may be described in a variety of forms, for example standard formal descriptions such as normal-form or extensive-form, as well as special-purpose representations suitable for particular algorithms. Game views are built selectively from game knowledge, and serve as the central object of game-theoretic analysis. In the poker example, different game views might abstract elements of the state space (e.g., grouping cards into classes) or action space (reducing betting options), or might selectively consider only parts of the game tree.

What counts as a "key element" of a strategic situation is inevitably subjective or speculative, so technically any game model would qualify as a game view. We introduce the redundant terminology expressly to emphasize the fact that any game model represents *just* a particular view of a more complex situation, and there are other views of comparable validity which may provide complementary perspectives.

Game view operations (GVOs) frame new game views based on previous game views and results from analyzing them. The new views are then constructed from the game knowledge. For example, GVOs might frame a new view as an abstraction or refinement of the previous, or as an elaboration or projection of other views. In poker, the domain of betting options could be adapted systematically based on results from prior views [2]. Formally, a GVO maps previous game views plus game knowledge into a new game view. This mapping typically includes explicit reasoning steps, such as solving previous game models according to specified solution concepts. Many specific GVOs are described below.

Finally, a ***navigation pattern*** is a generative sequence of game views, starting from an initial game view, and iterating through game view operations until a termination criterion is satisfied.

2 AN ILLUSTRATIVE EXAMPLE: IBR

We define a navigation pattern by specifying the game knowledge and initial game view Γ^0, along with reasoning steps, GVOs, and termination criteria. A typical process extracts a ***solution*** σ^{*k} for game view Γ^k at each iteration k, according to some solution concept as implemented by a specified ***solver***. The GVO then uses the solution σ^{*k} in producing Γ^{k+1} based on the game view history.

To illustrate the framework, we show how a simple game reasoning method—iterative best response (IBR)—can be cast in terms of game view navigation. The IBR method maintains a pure profile $s = (s_1, \ldots, s_n)$. On each iteration, we choose a player i and replace i's strategy with the best-response to the others, $s_i^* \in BR_i(s_{-i}) = \arg\max_{s_i \in S_i} u_i(s_i, s_{-i})$.

	Iterative Best Response (IBR)
GK	A base game Γ.
Init	$\Gamma^0 = \Gamma_{\downarrow(X_1,\ldots,X_{n-1},S_n)}$. Players $i \neq n$ are restricted to singleton sets: $X_i = \{s\}$, $s \in S_i$ arbitrarily chosen.
Solver	Derives $s^{*k} \in \mathrm{PSNE}(\Gamma^k)$.
GVO	$\Gamma^{k+1} = \Gamma_{\downarrow(X_1,\ldots,X_n)}$, with $X_i = \{s_i^{*k}\}$ for $i \not\equiv_n k+1$, and $X_i = S_i$ for $i \equiv_n k+1$.
Termination	On encountering a previously generated game view, that is, $\Gamma^k = \Gamma^{k'}$ for some $k' < k$.

Game knowledge (GK) in this instance takes on a simple form, that of a base game. Navigation operates on game views that are restrictions of this base game. Determining which restrictions to solve on each iteration is the essence of the navigation pattern.

(More flexible navigation patterns can make use of generative forms of game knowledge. For example, in approaches based on ***empirical game-theoretic analysis*** [6], GK is given in the form of an agent-based simulation model.)

For IBR, the key idea is that each game view fixes the strategies of all but one player. Note that when the strategy sets for all agents $j \neq i$ are singletons $\{s_j\}$, the PSNE computation in the *Solver* operation reduces to solving for $BR_i(s_{-i})$. Starting from an arbitrary assignment of strategies to $n-1$ players (*Init* step), IBR repeatedly determines a BR for the remaining player in this way, fixing that player's strategy to the BR, and moves on to the next player (*GVO*) who now has all their strategies available. The criterion for *Termination* is generation of a game view at iteration k equivalent to one seen at an earlier iteration k'. A PSNE of Γ^k represents a solution of the base game Γ if $k' = k - n$, which means that the singleton X_i are unchanged on the cycle for each i. Otherwise (i.e., $k' < k - n$) the strategy configurations are changing, and the procedure is cycling rather than converging.

3 MODULATING PLAYER AGGREGATION

The framework is also useful for developing new approaches. We specifically investigate navigation patterns that modulate levels of player aggregation, using an abstraction technique that approximates many-player games as games with fewer players [7]. As we demonstrate through computational experiments, navigating across levels of player reduction provides a flexible means to trade off computation and accuracy in identifying approximate equilibria.

REFERENCES

[1] Andrew Gilpin and Tuomas Sandholm. 2007. Better automated abstraction techniques for imperfect information games, with application to Texas Hold'em poker. In *6th Int'l Joint Conf. Autonomous Agents and Multi-Agent Systems*. 1168–1175.

[2] John Hawkin, Robert C. Holte, and Duane Szafron. 2012. Using sliding windows to generate action abstractions in extensive-form games. In *26th AAAI Conference on Artificial Intelligence*. 1924–1930.

[3] Tuomas Sandholm. 2015. Abstraction for Solving Large Incomplete-Information Games. In *29th AAAI Conference on Artificial Intelligence*. Austin, 4127–4131.

[4] Tuomas Sandholm and Satinder Singh. 2012. Lossy stochastic game abstraction with bounds. In *13th ACM Conference on Electronic Commerce*. 880–897.

[5] Michael P. Wellman and Katherine Mayo. 2024. Navigating in a space of game views. *Autonomous Agents and Multi-Agent Systems* 38, 31 (2024), 1–25.

[6] Michael P. Wellman, Karl Tuyls, and Amy Greenwald. 2025. Empirical game-theoretic analysis: A survey. *Journal of Artificial Intelligence Research* 82 (2025), 1017–1076.

[7] Bryce Wiedenbeck and Michael P. Wellman. 2012. Scaling simulation-based game analysis through deviation-preserving reduction. In *11th International Conference on Autonomous Agents and Multi-Agent Systems* (Valencia). 931–938.