

Texas Hold'em Meets Risk-Aware HTN Planning: A Modelling and Architectural Perspective

Ebaa Alnazer, Ilche Georgievski and Marco Aiello

Service Computing Department, IAAS, University of Stuttgart, Germany

Abstract

Games have long served as both a testbed and catalyst for advances in AI. Texas Hold'em stands out as one of the most strategically demanding variants of poker, characterised by high dynamism, uncertainty, and inherent risk, features commonly shared with real-world domains. At its core, every hand requires planning: deciding what action to take in the face of partial information and evolving strategy. This motivates us to propose a new direction for AI poker by framing Texas Hold'em as an AI planning problem. As a first step, we identify the key knowledge traits of the game using a conceptual framework for real-world planning domains, laying the foundation for a principled modelling approach. We then map gameplay elements into the framework of risk-aware HTN planning, which aligns well with modelling and playing poker, as it can effectively capture the game's strategic structure, uncertainty, and risk sensitivity. This approach allows player-specific risk attitudes to inform action choices based on hand strength and opponent behaviour. Building on this formulation, we propose a layered system architecture that enables adaptive decision-making during gameplay by interleaving risk-aware HTN planning, reactive refinement, learning, and execution. Taken together, our proposal represents a research agenda for modelling and solving strategic decision-making problems through risk-aware HTN planning.

Keywords

HTN planning, Poker, Risk-Awareness, Uncertainty, Domain Modelling, Texas Hold'em, System Architecture

1. Introduction

When Beyoncé sings, "*This ain't Texas, ain't no hold 'em, so lay your cards down...*", she likely is not thinking about poker algorithms. However, the lyrics capture something essential about Texas Hold'em: the game is a performance of intention and uncertainty, played out in layers of concealed information and strategy. Behind every hand lies a strategic plan of when to raise, bluff, or fold. This is based not only on card strength but also on a player's evolving beliefs about their opponents and the risks involved. In this sense, playing poker involves more than tactical moves; it requires constructing and adapting plans in real time, under uncertainty and risk.

Poker falls into the class of imperfect-information games, which serve as valuable domains for AI research because they reflect real-world decision-making challenges, such as uncertainty. In addition, a poker agent has to deal with hidden information, embrace stochasticity along with possibilities for deception, and adversarial dynamics. Over the years, the game of poker has attracted attention in AI as a challenging domain, e.g., [1, 2, 3]. This has led to the development of approaches that range from knowledge-based agents and simulation engines to theoretical equilibrium solutions and exploitative counter-strategies, culminating in AI-powered poker agents that are on par with professional players [4].

The notion of building a plan when playing poker resonates with AI planning, where the objective is to compute a course of action that achieves a desired outcome. AI planning has already been successfully applied in games, such as the bridge game [5] and video games [6]. Here, we postulate that decision-making in poker can likewise be viewed as solving a planning problem. Taking a planning perspective on poker allows us to frame decision-making not merely as reactive behaviour but as structured, goal-driven deliberation. This opens the door to applying established methods from AI planning.

Inspired by classical approaches that use game tree search for games of strategy (e.g., extensive-form games [7]), we consider Hierarchical Task Network (HTN) planning [8], a technique in AI planning,

ECAI Workshop on AI-based Planning for Complex Real-World Applications, October 25, 2025, Bologna, Italy

✉ ebaa.alnazer@iaas.uni-stuttgart.de (E. Alnazer); ilche.georgievski@iaas.uni-stuttgart.de (I. Georgievski); marco.aiello@iaas.uni-stuttgart.de (M. Aiello)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

particularly relevant, as its hierarchical structure mirrors the game tree enabling it to provide a natural model for the layered structure and decision-making in poker strategy. HTN planning has proven effective across many real-world domains, such as games, e.g., [6], robotics, e.g., [9], autonomous vehicles, e.g., [10], cloud computing, e.g., [11], and building automation, e.g., [12]. Our recently developed *risk-aware HTN planning framework* [13] enables explicit modelling of risk and uncertainty through probability distributions over action effects and costs, while preserving the expressiveness and computational strengths of HTN planning. These features make the framework a particularly suitable match for capturing the strategic reasoning required under risk and uncertainty in Texas Hold'em.

We propose a novel perspective: viewing Texas Hold'em, particularly No-Limit Texas Hold'em, the most strategically complex variant, as a risk-aware HTN planning problem. We explore this idea through two complementary aspects: modelling and system architecture. First, we examine how components of the game map onto constructs in risk-aware HTN planning, providing first steps toward formalising poker in this framework. Second, we propose a layered system architecture that interleaves long-term planning via risk-aware HTN planning, reactive refinement, learning, and execution monitoring.

Our contributions also include (1) identifying poker's key traits using an existing conceptual framework that emphasises realistic aspects in planning domains [14]; (2) mapping these traits to constructs in risk-aware HTN planning; and (3) designing a layered system architecture that enables adaptive planning for Texas Hold'em when conceptualised through the lens of uncertainty and risk attitudes.

The remainder of the paper covers related work (Section 2), background (Section 3), an analysis of knowledge traits (Section 4), the mapping to risk-aware HTN planning (Section 5), the layered architecture (Section 6), and conclusion (Section 7).

2. Related Work

Despite the research interest in applying AI planning to games, to the best of our knowledge, no prior work has specifically studied the use of AI planning for poker. Most existing literature either addresses general game planning or focuses on game types that differ significantly from poker in terms of structure, information availability, and objectives. Although poker shares certain features with other games, such as uncertainty, it also presents unique challenges, including hidden information, adversarial dynamics, and strategic deception, which remain largely unaddressed in planning research for games.

Moreover, the aspect of risk, particularly the incorporation of risk attitudes into decision-making, is absent in prior work. Existing approaches do not model or reason about varying attitudes toward risk when solving games as planning problems, which is a central concern here. We aim to close this gap by explicitly addressing how risk and risk sensitivity can be integrated into planning for playing poker.

The card game Bridge has served as a successful domain for AI planning [5], offering a perspective similar to ours in viewing gameplay as a planning task. This approach adapts HTN planning to handle uncertainty using belief functions to estimate unknown card positions and probabilistic reasoning for opponents' actions. However, it is domain-specific, meaning a specialised HTN planner was developed to play the game. In contrast, our proposal is general: it leverages domain-independent HTN planning engines, supports broader forms of non-determinism, and considers strategic elements like bluffing. Moreover, our proposal also accounts for adapting plans and learning opponent models.

Other studies on games propose to model uncertainty through expected effects [15]. Although their focus is on video games, we find this concept relevant and also explore its potential applicability in the context of poker. Likewise, research on combining multi-criteria evaluation and AI planning in computer games demonstrates how complex planning problems can be decomposed into smaller, manageable problems using techniques, such as partial order bounding and decomposition search [16]. While the modelling of these planning problems in conjunction with multi-criteria evaluation is not explored, this line of work presents a compelling approach for managing the strategic trade-off in poker and reducing the complexity by locally solving subgames within the larger game structure.

Some architectural work has laid the groundwork for integrating AI planning with game behaviour. For example, a three-layer planning architecture was proposed for highly dynamic and uncertain game

environments [17]. While this architecture inspired aspects of our proposed architecture, the overall work is tailored to a specific video game and does not capture poker complexities and characteristics.

Other relevant research integrates game-theoretic techniques with AI planning to reason about opponents' beliefs and their capacity to disrupt or defeat computed plans [18]. Although this work does not present game modelling specifically for AI planning, it offers valuable insights into planning under uncertainty. When combined with planning methods, presented techniques could enhance an agent's ability to handle uncertainty originating from opponent strategies, beliefs, and tactics, and adapt plans in dynamic game environments accordingly.

3. Background

3.1. No-Limit Texas Hold'em

No-Limit Texas Hold'em stands out as the most widely played, popular, and advanced poker variation from a strategy point of view, evidenced by the experts' divergent opinions on how to play a specific hand [19]. It is not only the most commonly associated with poker and the standard in casinos, but also the official ruleset in the World Series of Poker, the most prestigious poker tournament [20]. No-Limit Texas Hold'em is more aggressive and popular than its limit variant, as players can bet any amount, including their entire stack, and there is no limit on the number of raises per round.

Poker is typically played with a standard deck of 52 cards, consisting of combinations of 13 different cards (A, K, Q, J, T, 9, 8, 7, 6, 5, 4, 3, 2) and 4 different suits ($\spadesuit$, $\heartsuit$, $\diamondsuit$, $\clubsuit$), and can involve two to nine players. In most tournaments, each player starts with the same amount of chips, which is the currency used to buy into *rounds* (or *hands*) or bet with. If a player loses all his chips, the player is eliminated from the tournament. To win chips, multiple game rounds are played.

Each hand follows a fixed sequence of five *stages* (or *streets*), *pre-flop*, *flop*, *turn*, *river*, and *showdown*, each with a betting round where players choose one of five rule-governed actions: (1) *Raise*: By adding chips, at least twice the current highest bet, with no upper limit, to the pot to increase the current bet, thereby forcing opponents to match the bet or fold. This action is valid only when an open bet exists and the minimum amount to raise is met. (2) *Check*: To stay in the game without betting, only if no open bet exists, and it is not the pre-flop stage. (3) *Call*: If a previous raise occurred, a player may call by matching the highest current bet. (4) *Fold*: To discard the hand and exit the current round at any time. (5) *Bet*: To place chips to initiate wagering and forcing others to call or fold, only when no open bet exists, and it is not the pre-flop stage.

Each betting round lasts until only one player has not folded their hand, or all players have matched the bet. If a player goes all-in, betting their entire *bankroll* (i.e., all their remaining chips), they cannot bet further but remain eligible to win the pot up to their invested amount.

A hand begins with the pre-flop stage, where each player is dealt two face-down *hole cards*. Two players must post mandatory bets, the *Small Blind* (SB) and *Big Blind* (BB), before any cards are dealt. BB is typically twice the size of SB. These blinds create an initial pot and encourage action. The blinds increase as the tournament progresses, and the players responsible for posting them, as well as the turn order, rotate after each round. In pre-flop rounds, the first to act is the player left of the Big Blind; in other rounds, the SB acts first. Betting then proceeds clockwise.

After the pre-flop betting round, the flop stage begins, which includes players who have matched the BB. The *dealer* places three *community cards* face-up on the table, and a new betting round starts. The dealer does not bet but manages dealing, announces all-ins, and calculates *hand ranks*. After the flop betting round, the turn stage begins with one community card dealt, followed by a betting round. Then the river stage deals the final community card, followed by the last betting round. Finally, the showdown occurs, where players reveal their hands, and the highest hand wins the hand (i.e., the pot) or it is split among ties. Hand rank is determined by the best five cards from the community cards and the player's hole cards. Thus, the goal of each player is to form the best five-card hand using the shared community cards in conjunction with their hole hand at the end of the final betting round.

3.2. Risk-Aware HTN Planning

HTN planning is based on enhancing planning domains with detailed knowledge about how tasks can be performed [8]. This rich domain knowledge and its intrinsic hierarchical structure allow HTN planning to naturally reflect human-like decision-making processes. Specifically, HTN planning begins with an initial state and an initial task network as an objective to be accomplished. It also requires domain knowledge structured as networks of primitive and compound tasks. In this hierarchy, primitive tasks can be executed directly by actions, while compound tasks must be decomposed into subtasks using methods. One way to search for solutions involves decomposing the initial task network and continuing to do so until only primitive tasks remain. If successful, the result is a plan, which is an ordered sequence of primitive tasks executable from the initial state.

We developed risk-aware HTN planning, a framework that extends classical HTN planning with constructs that consider the uncertainty of real-world environments [13]. It enables modelling of risk and uncertainty through probability distributions over actions' costs and effects, where costs are defined as unbounded negative functions. This probability distribution ranges from totally known to totally unknown, depending on the available knowledge about the application domain. When the probability distribution is totally known, planning decisions have to be made under risk, whereas when it is not totally known, they have to be made under uncertainty. In an environment of risk and uncertainty, planning decisions should be made by following some *risk-attitude*, which defines the decision makers' mindset towards taking risks. Agents can be *risk-seeking*, i.e., prefer risky choices and tolerate losses, *risk-averse*, i.e., avoid risky choices, or *risk-neutral*, i.e., indifferent to risk. Risk attitudes are expressed mathematically using *utility functions*, which evaluate the action outcomes according to the risk attitude.

4. Characterising Texas Hold'em as a Planning Domain

Treating Texas Hold'em as an AI planning problem requires capturing its relevant aspects as domain knowledge, which is then formalised into a planning domain. Incomplete or inaccurate knowledge can result in domain models that misrepresent the application domain [21], leading to plans that fail real-world execution. We address this by applying a conceptual framework that supports identifying and categorising aspects of real-world planning domains [14]. We discuss the game knowledge traits in terms of objectives, tasks, quantities, non-determinism, behaviour, constraints, and qualities.

Objectives. At the highest level, a Texas Hold'em player's hard goal is to maximise long-term profit across many hands, with the subgoal of winning each individual hand. Both goals are quantitative.

Tasks. Texas Hold'em involves a hierarchy of tasks with varying levels of complexity. At the highest level, playing the game is a complex, long-term task composed of repeated iterations of playing individual rounds. Each round consists of sub-tasks corresponding to the different betting stages (pre-flop, flop, turn, river), which further decompose into finer-grained tasks. For example, before each round begins, a player must check their table position to determine whether they are required to post BB or SB. Within each stage of a round, a central task is selecting an action (folding, calling, betting, or raising), which depends on the current game state (e.g., position, stack size, community cards, opponents' actions) and the strategic framework the player adopts (e.g., exploitative versus equilibrium-based play, bluff frequency, risk tolerance). In addition, tasks in the game can often be accomplished in multiple ways. For example, the task of determining which action to take can vary depending on the player's strategy, their tendency to bluff, and their use of deception.

Quantities. Texas Hold'em contains several quantities that should be considered when representing the game knowledge. The quantities can be resources, such as the money and chips of each player, costs of the actions performed by the player or the opponents, i.e., the winnings and losses, the expected utility of each action computed based on the player risk attitude, and environmental inputs, such as the number of players, stack size, pot, amount to call, amount to raise, BB, and SB. The computation of some quantities is done using linear functions, such as the computation of the expected utility, number of players, and pot size, or non-linear function, such as the utility of a certain action outcome using the utility function of the player, probability of at least one player calling or raising, and evaluation metrics

to assess hand strength (e.g., Immediate Hand Strength, Hand Potential, and Effective Hand Strength), and to assess the pot odds for each bet (e.g., Expressed Pot Odds and Implied Pot Odds [22]).

Non-determinism. One of the key aspects of Texas Hold'em is the presence of uncertainty, which can lead to risky situations, where the probability distribution of the outcome is known or can be inferred from statistical inference [13], and/or uncertain situations, where the probability distribution is partially known, unknown, or unknowable. There are mainly two sources of uncertainty in this game. The first is related to the inherent randomness of the game, introduced by the shuffled deck and the stochastic distribution of cards. This is often modelled as a chance player, a concept used in game-theoretic formulations to represent non-deterministic events such as the dealing of cards [23]. The probabilities associated with these events are fully known and calculable, based on combinatorial analysis. For example, the probability of being dealt a specific pocket pair, such as two Aces, in Texas Hold'em is calculated using the formula: $\binom{4}{2} \cdot \binom{52}{2}^{-1} \approx 0.452\%$. This kind of probabilistic reasoning also applies to other scenarios, such as the odds of the opponent having a stronger hand, the probability of improving the hand after the flop, such as completing a straight draw, or the probability of specific cards appearing in the game's stages. The other source of uncertainty is related to the hidden or private information, such as the opponent's hand, their strategies and risk attitudes. Because poker is a game of imperfect information, players cannot directly observe opponents' hands. However, by interpreting betting patterns and timing, a skilled player can infer likely hand strength. For example, if an opponent consistently bets and raises aggressively, we might infer that they hold a strong hand. However, such inferences are strategically complex because opponents are aware that their actions convey information. Thus, they may bluff, intentionally acting strong (e.g., raising with a weak hand) to deceive other players into folding better hands. This interplay of belief modelling and deception introduces uncertainty. An additional and more technical source of uncertainty arises when considering poker-playing AI agents. Unlike deterministic systems that always choose the single best action, advanced agents typically compute a probability distribution over legal actions in a given game state. Even if raising has the highest expected value, the agent may occasionally choose to call or fold with a small probability. This form of action randomisation is intentional and serves to reduce exploitability by avoiding predictable patterns. These uncertainties in the game create a situation where there is always a degree of risk involved in every hand, as players must weigh the potential losses and gains and make decisions accordingly. As a result, these uncertainties make the consequences and costs of the player non-deterministic, ranging from known to unknown probability distributions of consequences, depending on player's knowledge.

Behaviour. Players in Texas Hold'em have risk attitudes (or risk tolerance), where some players are risk-loving, while others are risk-averse. Multiple works study the dynamics of the player's risk attitude during the game. For example, in [24], it is shown that within a reference point of wealth, players are risk-averse and the risk aversion decreases as wealth moves away from the reference point in either direction, while in [25], it is shown that players have lower marginal utility for additional wealth when they are wealthy than when they are poor. Although poker is not typically a cooperative game, the trust a player has in opponents can influence their decisions. For example, in [26], it is shown that players fold more often when their opponent's face appears trustworthy, showing how visual cues shape trust in decisions.

Constraints. Most tasks in the game have ordering constraints for their execution, i.e., are sequentially dependent. That is, they are totally ordered and certain decisions (e.g., posting blinds) must occur before others (e.g., choosing an action), and this order must be respected to maintain the game rules.

Qualities. When automating the planning process of a Texas Hold'em player, it is important that the behaviour of the planning system is explainable regarding three perspectives: the formation of domain knowledge, the formation of plans computed in the game, and the planning process itself.

5. Mapping Texas Hold'em to Risk-Aware HTN Planning

With the key components of Texas Hold'em identified as planning traits, we examine how gameplay can be modelled using risk-aware HTN planning, a framework well-suited to capture the game's uncertainty,

Table 1

Mapping between Texas Hold'em components and risk-aware HTN planning constructs.

Game Components	HTN Constructs	Examples
Tasks	Actions	Call, check, fold, all-in
Constraints, quantities	Preconditions	Game-rules (legal actions), strategies (bluff, deception), metrics (effective hand strength, expressed pot odds)
Tasks & ways to achieve them	Methods	Check-position, chance-tasks, determine-action, play-round, raise
Non-determinism	Variable costs	Probabilities of wins/losses
Non-determinism	Variable effects	Probabilities of game state changes
Quantities, context	Initial state	Position, pot, stack, SB, BB, current-street
Behaviour	Utility function	One-switch utility function

probabilistic outcomes, and strategies, while supporting decision-making under varying risk attitudes.

We begin by mapping the different components of the Texas Hold'em domain to corresponding constructs in risk-aware HTN planning. Table 1 summarises this mapping, including examples from gameplay. Generally, the tasks performed in the game can be modelled as compound tasks and operators in HTN planning. Game actions, such as call, fold, and check, can be modelled as operators, with preconditions governing the legality of game rules, i.e., allowing the execution of only valid actions in a specific game state (e.g., street). More complex game actions, such as raising, can be modelled as a compound task that can be decomposed by multiple methods, based on variables, such as the amount to raise. Here, we can apply abstraction, a common concept used in AI poker to minimise the number of possible actions that can be performed (e.g., [27, 28]). Consider the case of raise amount, where bet sizes are always relative, typically measured against either the current pot size or the size of the blinds. For instance, a bet of 100 chips may be considered large when the pot is only 10 chips, but the same amount would be negligible if the pot has already grown to 10,000 chips. Due to this relativity, we can categorise our raising options into multiple abstract levels, where the chip amounts are determined relative to the current pot size, allowing the agent to scale its raises appropriately with the game state.

Other decision-making processes in the game, such as determining which action to take, can also be modelled as compound tasks. These tasks may be decomposed using multiple methods, each governed by a set of preconditions that not only reflect the rules of the game but also incorporate strategic factors, such as bluffing and deception. For instance, the compound task for selecting an action can be decomposed via a specific method triggered when the agent intends to bluff. The decision of which method to apply can be dynamically selected during planning, depending on the agent's current strategy, potentially encoded in the game state, and its risk attitude. Additionally, strategic metrics (e.g., effective hand strength, expressed pot odds) that are computed to determine an action to be taken can be encoded in the preconditions of operators and methods. While these metrics may need to be computed dynamically via external functions, they play a key role in action selection.

Based on the logical or temporal structure of the poker game, ordering constraints between the tasks modelling the game knowledge can be modelled using totally, partially, or unordered task networks [8].

The non-deterministic aspects of Texas Hold'em can be represented in several ways, depending on the nature of the uncertainty involved. When the underlying probabilities are fully known, such as the likelihood of being dealt specific hole cards, they can be represented as chance events with probabilistic outcomes, often modelled via a chance player whose actions introduce stochastic effects into the environment. When the uncertainty is related to hidden information, such as the true value of the pot or the opponent's hand strength, modelling is more complex. Here, estimated metrics, such as expressed pot odds and implied pot odds, can inform probabilistic estimates and be modelled as preconditions of operators/methods. These estimates affect the operators' effects and costs (in terms of wins and losses) and make them variable. This modelling allows the agent to reason under uncertainty even when outcomes are not strictly random but instead depend on incomplete or inferred information.

Player behaviour is captured via a utility function, which evaluates outcomes in terms of expected utility based on the agent's risk attitude. This utility function is a component of the risk-aware HTN planning formulation, enabling the computation of plans with the highest expected utility. Such plans

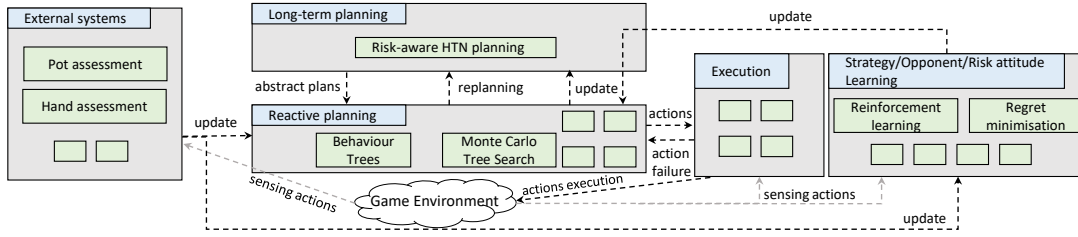


Figure 1: A layered planning architecture for Texas Hold'em. Grey boxes represent the layers, green boxes represent the components, and arrows represent the interactions between the layers.

reflect an agent's individual preferences toward risk when playing poker.

Finally, the initial task network contains the overarching gameplay objective, such as playing the game of multiple rounds or play a round, while the initial state includes contextual game information like pot size, stack, SB, BB, position of the player around the table, current street, and possibly the state of other opponents regarding their aggression levels and known strategies. Together, these elements define the planning problem instance that governs the agent's decision-making throughout the game.

6. Layered Planning Architecture for Adaptive Play

The primary challenge in developing approaches that model and solve Texas Hold'em as a planning problem lies in the highly dynamic nature of the game, which makes long-term planning non-trivial. Long-term planning has to be made on estimations of opponents' behaviour, hand strengths, pot assessments, hole card assessments, and the likely future flow of the game, all of which are inherently uncertain. As a result, preconditions assumed during planning may no longer hold during execution, making long-term plans fail to match the actual flow and game events. This mismatch between planned and actual (game) trajectories is a well-known challenge in dynamic environments.

This challenge is commonly addressed by interleaving planning and execution [29]. If execution fails due to unforeseen events, an execution module can trigger plan repair or replanning [30, 17, 31]. In rapidly changing environments like poker, such failures may occur frequently, leading to repeated replanning. This not only imposes a high computational cost but can also slow down agent's responsiveness. To mitigate this, some approaches do not replan starting from the initial task network, but opt for localised planning, that is, do plan repair or replan from the point of failure [31].

To address this primary challenge in Texas Hold'em, we propose a layered architecture, inspired by prior work in game environments [17]. Figure 1 provides an overview of the proposed architecture. At the top is the long-term planning layer represented by risk-aware HTN planning. This layer is responsible for generating abstract plans with the highest expected utility that may include compound tasks and benefit from knowledge abstraction. Below that is a reactive planning layer responsible for plan refinement at runtime, enabling adaptation to evolving game conditions. That is, the layer interprets abstract plans based on current context or failures and adjusts plans dynamically while still aiming for plans with the highest expected utility. Reactive techniques, such as Behaviour Trees [32] and Monte Carlo Tree Search [33] are applicable here. The architecture contains two other layers: the execution and learning layers. The execution layer interacts directly with the game environment. It executes actions from refined plans and continuously monitors the game state. If an action fails, let's say, due to invalid preconditions, the execution layer triggers the reactive planning layer to revise the action or the compound task it belongs to. For example, suppose `raise` to be a high-level task and the reactive planning layer to specify raising a certain number of chips. If the specified raise is no longer valid due to an intervening player action with a higher raise, the reactive planning layer must adapt and generate a valid raise. For more substantial deviations or failures, such as when raising is no longer a valid option, the player busts, or a fundamental strategy change is required, the reactive planning layer escalates the issue to the long-term planning layer for repair or recomputation of the abstract plan. Importantly, this process does not need to start from scratch; the system can retain and leverage

search space traversal history and resume from the point of failure.

The learning layer is concerned with opponent modelling. Integrating it into the planning loop enables the agent to adapt its strategies based on observed opponent behaviour. Learning from patterns, such as fold frequency or bluffing tendencies, can improve AI agents playing poker [1, 34]. Building opponent models through learning allows agents to act exploitatively [1]. Embedding these models in the planning layers lets the agent adjust task and selection based on inferred strategies. Moreover, repeated interaction can inform not only strategy but also risk attitude: if high-risk play consistently fails against certain opponents, the agent can adapt its utility function to favour lower-risk decisions. Such changes in the risk attitude are propagated to the reactive planning layer, which propagates it to long-term planning whenever replanning is required. Thus, integrating learning, such as reinforcement learning, enables both convergence to equilibrium strategies and dynamic adjustment of risk sensitivity.

We further separate concerns of estimation tasks (e.g., pot assessment, hand strength estimation) as distinct components, which can also be seen as systems external to the architecture. These systems may feed context updates into the reactive planning and learning layers. If that new context significantly alters the game dynamics, these layers may prompt the long-term planning layer to initiate replanning.

7. Conclusion

We propose a new direction for AI poker research by framing No-Limit Texas Hold'em as a risk-aware HTN planning problem. This perspective leverages the game's strategic depth and inherent uncertainty, offering a structured approach to decision-making that accounts for win/loss outcomes and risk attitudes. Our work introduces this framing and lays the groundwork for modelling and solving poker through risk-aware HTN planning. We began by identifying the key knowledge characteristics of No-Limit Texas Hold'em using a conceptual framework for real-world planning domains. These features informed our mapping of the game's components into constructs within the risk-aware HTN planning framework. Building on this, we proposed a layered architecture that supports adaptive decision-making during play by interleaving long-term planning, reactive refinement, learning, and execution monitoring.

Our proposal opens several avenues for further investigation. One question is which game aspects can and cannot be effectively modelled within risk-aware HTN planning. For example, opponent modelling and strategy could be learned, but those outputs should still be integrated into planning through modelling in the initial state and preconditions of actions/methods. Another area of interest is the practical realisation of the proposed framework. We are working on encoding a domain model for No-Limit Texas Hold'em and implementing the layered planning architecture.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

- [1] M. B. Johanson, Robust strategies and counter-strategies: Building a champion level computer poker player, Ph.D. thesis, University of Alberta, 2007.
- [2] B. Beattie, G. Nicolai, D. Gerhard, R. J. Hilderman, Pattern classification in no-limit poker: A head-start evolutionary approach, in: CCSCSI on Advances in AI, 2007, pp. 204–215.
- [3] A. Gilpin, T. Sandholm, T. B. Sørensen, A heads-up no-limit Texas Hold'em poker player: Discretized betting models and automatically generated equilibrium-finding programs, in: International Joint Conference on Autonomous Agents and Multiagent Systems, 2008, pp. 911–918.
- [4] J. Rubin, I. Watson, Computer poker: A review, *Artificial Intelligence* 175 (2011) 958–987.
- [5] S. Smith, D. Nau, T. Throop, Computer bridge: A big win for AI planning, *AI Mag.* 19 (1998) 93–93.
- [6] J.-P. Kelly, A. Botea, S. Koenig, Offline planning with hierarchical task networks in video games, in: *AAAI Conference on AI and Interactive Digital Entertainment*, volume 4, 2008, pp. 60–65.

- [7] T. Davis, K. Waugh, M. Bowling, Solving large extensive-form games with strategy constraints, in: AAAI Conference on AI, volume 33, 2019, pp. 1861–1868.
- [8] I. Georgievski, M. Aiello, HTN planning: Overview, comparison, and beyond, *AI* 222 (2015) 124–156.
- [9] M. Weser, D. Off, J. Zhang, HTN robot planning in partially observable dynamic environments, in: IEEE International Conference on Robotics and Automation, 2010, pp. 1505–1510.
- [10] E. Alnazer, I. Georgievski, N. Prakash, M. Aiello, A role for HTN planning in increasing trust in autonomous driving, in: IEEE International Smart Cities Conference, 2022, pp. 1–7.
- [11] I. Georgievski, F. Nizamic, A. Lazovik, M. Aiello, Cloud Ready Applications Composed via HTN Planning, in: IEEE Int’l Conf. on Service-Oriented Computing and Applications, 2017, pp. 23–33.
- [12] I. Georgievski, T. A. Nguyen, F. Nizamic, B. Setz, A. Lazovik, M. Aiello, Planning meets activity recognition: Service coordination for intelligent buildings, *Perv. and Mob. Comp.* 38 (2017) 110–139.
- [13] E. Alnazer, I. Georgievski, M. Aiello, Risk Awareness in HTN Planning, N. 2204.10669, arXiv, 2022.
- [14] E. Alnazer, I. Georgievski, Understanding real-world AI planning domains: A conceptual framework, in: SummerSOC, Communications in Computer and Information Science, 2023, pp. 3–23.
- [15] T. Humphreys, Exploring HTN planners through example, in: *Game AI Pro 360: Guide to Architecture*, CRC Press, 2019, pp. 103–122.
- [16] M. Muller, Multicriteria Evaluation in Computer Game-Playing, and its Relation to AI Planning, in: International Conference on AI Planning & Scheduling, 2002, p. 1.
- [17] X. Neufeld, Long-term planning and reactive execution in highly dynamic environments, Ph.D. thesis, Otto von Guericke University Magdeburg, 2020.
- [18] R. Vane, P. E. Lehner, Hypergames and AI in automated adversarial planning, in: DARPA Planning Workshop, 1990, pp. 198–206.
- [19] D. Sklansky, M. Malmuth, *Hold’em Poker for Advanced Players*, Two Plus Two Publishing (1999).
- [20] S. Braids, *The Intelligent Guide to Texas Hold’em Poker*, Chartley Publishing LLC, 2010.
- [21] T. S. Vaquero, J. R. Silva, F. Tonidandel, J. C. Beck, itSIMPLE: towards an integrated design system for real planning applications, *The Knowledge Engineering Review* 28 (2013) 215–230.
- [22] D. Billings, A. Davidson, J. Schaeffer, D. Szafron, The challenge of poker, *AI* 134 (2002) 201–240.
- [23] M. Bowling, N. Burch, M. Johanson, O. Tammelin, Heads-up limit hold’em poker is solved, *Science* 347 (2015) 145–149.
- [24] G. Smith, M. Levere, R. Kurtzman, Poker player behavior after big wins and big losses, *Management Science* 55 (2009) 1547–1555.
- [25] M. Rabin, Risk aversion and expected-utility theory: A calibration theorem, in: *Handbook of the fundamentals of financial decision making: Part I*, World Scientific, 2013, pp. 241–252.
- [26] E. J. Schlicht, S. Shimojo, C. F. Camerer, P. Battaglia, K. Nakayama, Human wagering behavior depends on opponents’ faces, *PloS one* 5 (2010) e11663.
- [27] M. Johanson, N. Burch, R. Valenzano, M. Bowling, Evaluating state-space abstractions in extensive-form games, in: Int’l Conf. on Autonomous Agents and Multi-Agent Systems, 2013, pp. 271–278.
- [28] M. Moravčík, M. Schmid, N. e. a. Burch, Deepstack: Expert-level artificial intelligence in heads-up no-limit poker, *Science* 356 (2017) 508–513.
- [29] I. Georgievski, M. Aiello, Automated planning for ubiquitous computing, *ACM CSUR* 49 (2016) 63:1–63:46.
- [30] D. J. Soemers, M. H. Winands, Hierarchical task network plan reuse for video games, in: IEEE Conference on Computational Intelligence and Games, 2016, pp. 1–8.
- [31] Y. Bansod, D. Nau, S. Patra, M. Roberts, Integrating Planning and Acting With a Re-Entrant HTN Planner, in: ICAPS Workshop on Hierarchical Planning, 2021, pp. 22–54.
- [32] G. Robertson, I. Watson, Building behavior trees from observations in real-time strategy games, in: IEEE Int’l Symposium on Innovations in Intelligent Systems and Applications, 2015, pp. 1–7.
- [33] M. Chung, Monte Carlo planning in RTS games (2005).
- [34] S. Ganzfried, T. Sandholm, Potential-aware imperfect-recall abstraction with earth mover’s distance in imperfect-information games, in: AAAI Conference on Artificial Intelligence, volume 28, 2014.