SEQUENTIAL DECISION MAKING IN GAMES WITH INCOMPLETE INFORMATION

By

ABHISHEK NINAD KULKARNI

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2023

To my family

ACKNOWLEDGEMENTS

I want to begin by expressing my deepest gratitude to my advisor, Prof. Jie Fu. Over the course of the past six years, she has been an exceptional guide and source of inspiration. Prof. Fu has consistently encouraged me to pursue my research interests with unwavering support, enabling me to delve into a diverse array of topics, ranging from theoretical concepts in game theory and formal methods to the practical realms of cybersecurity and robotics. Her dedication to helping me find my own unique voice as a researcher has been pivotal to my growth throughout my PhD journey. I am indebted to her for her meticulous feedback on my work and our thought-provoking discussions, which have been marked by originality, precision, and enlightening insights.

I would like to thank the members of my dissertation committee, Prof. Sean Meyn, Prof. Tuba Yavuz, and Prof. Yu Wang, and the members of my PhD qualifying exam committee, Prof. Carlo Pinciroli from Worcester Polytechnic Institute and Dr. Mitchell Colby from Scientific Systems Company Inc. Their insights and advice have been valuable in shaping the course of my academic journey.

I extend my heartfelt appreciation to all my coauthors, including Dr. Charles A. Kamhoua, Dr. Nandi O. Leslie, Prof. Shuo Han, Dr. Hazhar Rahmani, Dr. Lening Li, Haoxiang Ma, Sumukha Udupa, Matthew Cohen, Huan Luo, Yash Shukla, Dr. Robert Wright, Dr. Alvaro Velasquez, Dr. Jivko Sinapov, Dr. Siddharth Patki, Dr. Satish R. Inamdar, Prof. Madhuri Joshi, and Prof. Anita S. Joshi. Their invaluable contributions and the enlightening discussions we shared were instrumental in bringing my research to fruition. Collaborating with Dr. Kamhoua was a truly delightful experience, and I gained countless insights into the domain of cyber-physical systems security through our discussion. I am deeply grateful for his steadfast support of my research and his invaluable mentorship. I am deeply indebted to Prof. Inamdar, who played a pivotal role in acquainting me with the domain of research and introducing me to the concept of cyber-physical systems. This introduction served as the underpinning for the research that forms the core of this dissertation.

I express my gratitude to the members of the Control and Intelligent Robotics Lab (CIRL) for enriching my research journey with their engaging discussions on intriguing new challenges. Their close-knit collaboration has made our research dialogues not only productive but also enjoyable. I am also grateful to the wonderful conferences that I have had the privilege to attend, including the Conference on Decision and Control (CDC), American Control Conference (ACC), International Joint Conference on Artificial Intelligence (IJCAI), Conference on Decision and Game Theory for Security (GameSec), and International Conference on Robotics and Automation (ICRA). These conferences have provided me with invaluable opportunities to engage with leading researchers in my field.

My family has been an endless source of love, affection, support and motivation for me. My father, Ninad Kulkarni, has been instrumental in igniting my passion for mathematics and research, my mother, Snehal Kulkarni, gave me the most important lesson in life, that of handling failures, and my wife, Sanika Patki, has been a source of great emotional support and encouragement during the ups and downs of the PhD life. It is the blessings and belief of my grandparents in my abilities through the years that has given me the strength to pursue my dreams, even at times when they seemed unrealizable.

I extend my heartfelt gratitude to Prof. Prakash Mulbagal, my mathematics mentor at M. Prakash Academy in Pune, India. Under his guidance, my passion for mathematics was nurtured, and he inspired me to set forth on the trajectory towards a career in research and development. Perhaps the most profound lesson imparted by Prakash Sir was the philosophy of effective learning, serving as the bedrock upon which I embarked on my journey in the fields of mathematics and science. This journey ultimately culminated in the research presented in this dissertation.

I would also like to express my gratitude to Prof. Milind Patwardhan, Prof. Pushkar Joglekar, Prof. Milind Kamble, and Prof. Mrunal Shidore for their mentorship, encouragement, and unwavering support. Their guidance has played a pivotal role in my transformation from being solely an admirer of theoretical concepts to someone who also values the practical

implications of theory. Engaging in conversations with Prof. Patwardhan about the practical intricacies of robot functionality and concurrently discussing the theoretical facets of motion planning with Prof. Joglekar struck the perfect balance which motivated me to not only delve into theory or practice but also to bridge the gap between them. This approach laid the foundation for my thought process. I'd also like to extend my appreciation to Monica Patel, Aditya Joshi, and Shruti Phadke for their numerous insightful discussions and enjoyable moments, which not only contributed to my personal growth but also provided vital support during challenging personal trials.

In the course of the last seven years, I have been exceptionally fortunate to discover a close-knit community among the friends I made in the United States, including Krunal Chaudhari, Kritika Iyer, Ankur Agrawal, Anand Parwal, Aashima Parwal, Adhavan Jayabalan, Kenechukwu Mbanisi, Ishita Ankit, and Shubham Jain. From the philosophical discussions on the existence of God to contemplating the future of robotics and AI, our conversations have spanned a wide spectrum of topics that have profoundly impacted on my approach to research. This group has become like a second family to me, and it's thanks to them that my PhD journey has been an overwhelmingly positive experience.

Last but not the least, I want to thank my funding sources, DARPA, ARL, NSF, and Dr. Glenn Yee Scholarship, for supporting my PhD.

TABLE OF CONTENTS

page

LIST OF TABLES

Tables page

LIST OF FIGURES

Figures page

LIST OF ALGORITHMS

Algorithms page

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

SEQUENTIAL DECISION MAKING IN GAMES WITH INCOMPLETE INFORMATION

By

Abhishek Ninad Kulkarni

December 2023

Chair: Jie Fu
Major: Electrical and Computer Engineering

Sequential decision-making in non-cooperative games is an indispensable skill for autonomous agents to achieve complex temporal objectives in dynamic, uncontrolled environments in presence of other strategic agents. This dissertation studies the problem of synthesizing winning strategies in games with incomplete information played on graphs—a class of games that has received limited attention, yet holds significant implications in domains such as robotics, economics, and artificial intelligence.

We investigate the synthesis problem under two kinds of incomplete information. In two-player games, we consider situations where an adversary (P2) has incomplete information about the action capabilities or objectives of the agent (P1) or how P1 interprets the history of their interaction. We show that, in such situations, P1 may synthesize a deceptive strategy to satisfy its omega-regular objective that exploits P2's incomplete information to gain a strategic advantage. However, the effectiveness of a deceptive strategy depends on the level of awareness of P2 about its incomplete information.

We develop hypergame theory for games on graphs by introducing two models: static and dynamic hypergames on graphs, which model situations where P2's information remains constant or evolves during the interaction. These models capture interactions where both players play according to their subjective views of their interaction constructed using the information they know. We introduce new solution concepts to analyze the rational behavior of players within hypergames, based on which we identify the conditions for the use of deception to be

advantageous for P1 and design algorithms to synthesize the deceptive winning strategies under various assumptions on P2's incomplete information.

In single-player stochastic games, we study planning with incomplete preferences over omega-regular objectives, where the player may lack information about its own preferences. Decision-making is challenging in this setting because incomplete preferences do not always admit a utility representation, which renders classical decision theory inapplicable. We introduce a novel framework for planning with incomplete preferences over linear temporal logic objectives that include a preference language, an automata-theoretic computational model, and algorithms to synthesize preference-satisfying strategies under two new solution concepts.

CHAPTER 1
INTRODUCTION

The ability of sequential decision making is central to human cognition. It enables individuals to tackle intricate problems, adapt to dynamic environments, interact effectively with others, manage risks, and work toward long-term goals by making a series of interconnected choices. For instance, in a game of chess, a player determines their next move by anticipating multiple rounds of moves and potential counter-moves of their opponent. As autonomous agents become an integral part of human society, it is imperative for these agents to exhibit proficiency in making competent and strategic sequential decisions.

Game theory provides a theoretical framework to study sequential decision making problems. It focuses on the analysis of rational decision making of agents involved in strategic interactions within a stochastic environment in presence of other strategic and self-interested agents. Game theory offers a suite of mathematical tools, including *models* that can capture interactions between one or more players under various scenarios, and *solution concepts* that define the conditions on what constitutes rational behavior for players within the game. For instance, the simplest class of games called normal-form games consist of "one-step" games, in which players select their strategies simultaneously, and the outcomes are determined by a payoff matrix specifying the payoffs associated with all possible strategy combinations. The game ends once the players choose their actions. However, a considerable subset of games evolve over time and in a stateful manner, and the payoffs received by the players depend on the history of interactions. In such games, the ability of players to make strategic sequential decisions is crucial to achieve a desirable outcome.

Games on graphs. A game played on a graph (for short, a game on graph) is a model used to study sequential interactions between one or more players that evolve over time in a stateful manner. They have garnered significant attention in various domains, including cybersecurity [1, 2, 3, 4], adversarial robot motion planning [5, 6], and discrete event systems [7, 8], among others. These games can represent non-terminating interactions that evolve indefinitely,

advancing through an unbounded number of rounds. Within this model, each game state corresponds to a node within the graph, and during each round, players make strategic choices that trigger transitions to successor nodes through edges. An outcome of these non-terminating games is represented by the infinite path in the graph defined by the strategies of the players. In this dissertation, we focus on reachability and safety ω-regular games on graphs [9], *i.e.*, the class of games in which players' objectives are characterized by an ω-regular language. A language containing infinite words is ω-regular if it can be expressed by a finite Büchi automaton. Specifically, we reachability consider objectives specified using a fragment of Linear Temporal Logic (LTL) called syntactically cosafe LTL (scLTL) [10], which can express ω-regular languages representable by a terminal Büchi automaton, and safety objectives specified using LTL, which can express ω-regular languages representable by a monitor.

Types of games. Games are categorized into four types based on two factors: whether all players have perfect information, and whether all players have complete information. In a game with *imperfect information* [11, 12], one or more players have partial or limited knowledge about the history of game states or the actions executed by other players. The models such as Partially Observable Markov Decision Processes (POMDPs) [13] and Partially Observable Stochastic Games (POSGs) [14] are noteworthy examples of games on graphs with imperfect information.

On the other hand, in a game with *incomplete information* [11, 15], one or more players have partial or limited knowledge about at least one of the following components of the game: (a) players' action capabilities, (b) players' objectives, (c) the game rules, (f) what one player knows about the other player, and what the other player knows about the information known to the first player, and so on .... When a player's knowledge is incomplete regarding their own capabilities or objectives, the incompleteness in the game is termed as *interoceptive.* In contrast, when a player's knowledge is incomplete concerning the external environment, encompassing aspects such as game rules, other players' capabilities, or objectives, it is referred to as *exteroceptive.*

The synthesis problem. A central problem about games on graphs is to synthesize winning strategies for a player. A strategy is said to be winning if following it guarantees that the player will satisfy its ω-regular objective regardless of the strategies employed by other players. However, the notion of "winning" in games on graphs varies depending on which solution concept is used to analyze the game. In this research, we focus on the qualitative solution concepts of sure, almost-sure, and positive winning [9]. A *sure winning* strategy guarantees the player that its objective will be accomplished in a finite number of steps. An *almost-sure winning* strategy provides the player with the assurance of achieving their objective with a probability one, while a *positive winning* strategy assures the player that its objective will be realized with a positive probability.

Literature on games on graphs with exteroceptive incomplete information. The synthesis problem has received significant attention in theoretical computer science and control systems for the class of games on graphs with perfect or imperfect, but complete, information. In analyzing these games, three questions are considered to be fundamental. First, whether the game is *determined*, *i.e.*, whether one of the players has a strategy to win (*i.e.*, satisfy its ω-regular objective) regardless of the strategy employed by the opponent? Determinacy is valuable in solving the synthesis problem because it allows for transitioning between the viewpoints of the two players: For example, if P1 does not have a winning strategy from a state, then the determinacy property guarantees that P2 has a winning strategy from that state. Interested readers may find a detailed discussion on determinacy in [16]. Second, does there exist an algorithm to *characterize state space*, *i.e.*, to identify which player wins from a given state? This assists in designing winning strategies; for example, a winning strategy to satisfy a safety objective must reject any action that leads to an opponent's winning state. Lastly, does there exists an algorithm to synthesize the winning strategy for a player from each of its winning states. If yes, then what is the computational complexity of such an algorithm?

Games on graphs with perfect and complete information. The answers to the above questions depend largely on the class of game on graph and the solution concept being considered. In case of games on graphs with perfect and complete information, it is known that the deterministic as well as stochastic turn-based variants are qualitatively determined [17], but their concurrent counterparts are not[1] [19, 20]. Regarding the characterization of state space, the solution concepts of sure and almost-sure winning are known to coincide for the deterministic turn-based games. This means that the set of winning states for either of the players remains the same regardless of which solution concept is employed to analyze the game. This is not the case with either stochastic turn-based games or concurrent games. In fact, for concurrent games, the strategies that use randomization are more powerful than the deterministic (i.e., pure) strategies [9]. Therefore, the number of winning states for one player may be greater under almost-sure winning concept when compared to that under sure winning. Lastly, the algorithms to synthesize winning strategies are known for most sub-classes of games on graphs with perfect and complete information. A few noteworthy algorithms include the linear-time algorithm for ω-regular reachability games [21], and polynomial-time algorithms for stochastic turn-based games and concurrent games with both qualitative reachability and more general parity objectives, as discussed in [9, 19].

Games on graphs with imperfect but complete information. Seminal works by Reif [22, 12] established the foundational framework for studying games on graphs with imperfect information. In these seminal works, Reif introduced a subset construction methodology to transform games with imperfect information into those characterized by perfect and complete information. This approach established the way for synthesizing winning strategies under the sure winning concept, specifically for the deterministic turn-based games on graphs. Subsequently, research demonstrated that all turn-based and concurrent games with imperfect information are determined when players employ randomized strategies, but are not determined when deterministic strategies

1 Note that stochastic concurrent games exhibit quantitative determinacy but lack qualitative determinacy. Quantitative determinacy involves computing the maximal probability with which a player can win in the limit from each state [18].

are employed. The subset construction, however, results in an exponential blowup of state space, resulting in the majority of algorithms for synthesizing winning strategies in these games to have at least exponential time and space complexity. The synthesis algorithms were first presented for partially observable Markov decision processes (POMDP), which represent a class of single-player games with imperfect information. Subsequently, a series of works [23, 24, 25] expanded these algorithms to address two-player games on graphs involving one-sided imperfect information. Recently, Bertrand et al. [16] introduced a doubly exponential algorithm for games featuring two-sided partial observation, while Gripon and Serre [26] extended these findings to encompass games where players may not observe the history of actions in addition to the history of states.

1.1 Aim of this Dissertation

In contrast to games on graphs with either perfect or imperfect information, sequential decision making in games on graphs with incomplete information has received little attention. The aim of this dissertation is to address this gap. Let us first understand the reason behind this gap by reviewing the literature on normal-form and extensive games with incomplete information.

Games on graphs with incomplete but perfect information. Two models are commonly used to represent games with incomplete information: Bayesian games [11] and hypergames [27]. Among these, Bayesian games are widely recognized as the standard model of games with incomplete information in game theory. This can be attributed to the foundational work by Harsanyi [28], in which he argued that any game with incomplete information can be equivalently transformed into a game with imperfect (but complete) information. The transformation entails assigning a type to each player in a game with incomplete information, where the type corresponds to their private information. Subsequently, assuming that all players know the set of potential types, the players maintain a subjective probability distribution over this set. During interaction, they update this distribution based on observations to infer the true type from the history of a player's decisions. To establish the initial distribution, Harsanyi's framework relies on the critical assumption that all players share a common prior distribution. In their seminal work,

Mertens and Zamir [29] relaxed this assumption by introducing the notion of a universal belief space. This work established that Harsanyi's model can indeed represent all kinds of games with incomplete information. The development of the Bayesian games model for repeated games, which represents a class of sequential interactions consisting of a number of repetitions of the same base game, was introduced by Aumann et al. [30]. There are three widely studied solution concepts for Bayesian games: The Bayesian Nash equilibrium [31] extends the concept of Nash equilibrium to normal-form games with incomplete information. The correlated equilibrium [30] represents the Nash equilibrium of a game extended by the inclusion of random events, about which players possess partial information. The perfect Bayesian equilibrium (PBE) [32] refines the Bayesian Nash equilibrium, particularly tailored for extensive-form games marked by incomplete information.

In contrast, hypergames [27] offer a framework capable of modeling games in which some players may be misinformed of some aspects of their interaction or remain unaware of their own and other players' misperceptions. Conceptually, a hypergame integrates the subjective views of all players about the game. Consequently, players can have distinct perceptions of the game without necessitating the assumption of common prior knowledge. Both normal-form and extensive-form versions of hypergames have been extensively explored in the literature [33, 34, 35], particularly within contexts of conflicts [36, 37, 38] and deception [39, 40, 35, 41, 42]. Several solution concepts have been proposed for hypergames including the Nash equilibrium, hyper-Nash equilibrium, Fraser-Hipel equilibrium, Stackelberg equilibrium, and subjective rationalizability [33, 43]. An in-depth discussion about various solution concepts and relations among them is studied in [33].

Challenges to study games on graphs with incomplete information. Although, theoretically, Bayesian games can model every kind of incompleteness, we argue that they might not be the best choice to study the synthesis problem in games on graphs with incomplete information. We highlight three reasons for our hypothesis.

First, the assumption that the set of possible types is common knowledge for all players is unreasonable in many situations, especially those involving conflicts [41] or unawareness [44]. This assumption, in addition to the assumption of common prior, have been widely debated in economics and game theory communities [45, 46]. For instance, in cybersecurity, an attacker may not be aware about defender's use of honeypatches, which are patched vulnerabilities that appear like unpatched vulnerabilities to the attacker [47]. In this situation, the attacker cannot know the possible types of defenders.

Second, in the games where common prior assumption does not hold, the existing solution concepts for Bayesian games provide limited insight [48]. This is highlighted by the transformation from a hierarchical hypergame to a Bayesian game proposed by Sasaki and Kijima [48]. Through this transformation, the authors show that the solution concept of subjective rationalizability in hypergame coincides with that of Bayesian Nash equilibrium in its Bayesian representation, and the best response equilibrium in hypergame corresponds to Nash equilibrium in its Bayesian representation. However, the equivalent counterparts of the hypergame solution concepts such as Fraser-Hipel equilibrium [49, 50], or hyper-Nash equilibrium [51, 52] are not known.

Lastly, Harsanyi's approach to transform a game with incomplete information into one with imperfect information might not be effective for games on graphs due to the complexity of solving games on graphs with imperfect information. Last but not the least, the Bayesian games are inherently quantitative in nature and, therefore, are not best suited for solving the synthesis problem using qualitative solution concepts.

A large part of this dissertation is dedicated to developing the hypergame theory for two-player games on graphs, in which the ego player, P1, is aware that its adversary, P2, lacks knowledge about some component of the game. Formally, the first of the two research questions posed in this dissertation is stated as follows.

Research Question I

In a two-player game on graph with one-sided incomplete information, where P1 has complete and P2 has incomplete information, how to synthesize strategies for P1 that are provably-correct with respect to given ω-regular specifications and which leverage P1's knowledge about P2's incomplete information to gain strategic advantage over P2?

Literature on games on graphs with interoceptive incomplete information. In games with interoceptive incomplete information, players lack full knowledge of their own capabilities or objectives. This dissertation focuses on a specific category of single-player stochastic games on graph, also known as Markov Decision Process (MDP). In these games, the objective is to synthesize a strategy that achieves the most desirable goal, considering an incomplete preference over a set of ω-regular reachability objectives. A preference relation over a set of alternatives is said to be *complete* if the ordering between every possible pair is alternatives is well-defined. In other words, the preference relation can compare and rank any pair of alternatives and make a clear decision based on their preferences [53]. On the other hand, as preference relation is said to be *incomplete* if the relation is unable to rank or compare certain alternatives. The problem of making rational decisions to achieve most desirable goals given a preference relation is studied widely in the domain of preference-based planning [54].

Literature on preference-based planning. The literature on preference-based planning can be classified into four parts based on the whether the preference relation is complete or incomplete, and whether the environment is deterministic or stochastic. Planning with preferences over temporal goals in deterministic environment is a well-studied problem for both complete and incomplete preferences (see [54] for a survey). For preferences specified over temporal goals, the authors in [55] proposed a logical language for specifying preferences over the evolution of states and actions to synthesize a deterministic plan while the works [56, 57, 58] explored the minimum violation planning approaches that decide which low-priority constraints should be violated in a deterministic system, when not all objectives can be satisfied simultaneously. Mehdipour *et*

al. [59] associate weights with Boolean and temporal operators in signal temporal logic to specify the importance of satisfying the sub-formula and priority in the timing of satisfaction. This reduces the preference-based planning problem to maximizing the weighted satisfaction in deterministic dynamical systems.

However, the solutions to the preference-based planning problem for deterministic systems cannot be applied to stochastic systems. This is because, sequential decision making with preferences requires the agent to transform a preference over a set of high-level temporal goal into a preference over strategies. Thus, by following the most-preferred strategy, the player would be guaranteed to achieve the most desirable goal. Now, in stochastic environments, even a deterministic strategy yields a distribution over outcomes satisfied by the resulting paths. Therefore, to determine which strategy is better, we need a way to compare distributions over paths instead of comparing two paths[2], which is what the deterministic planners do.

Several works have studied the preference-based planning problem in stochastic environments. But almost all of them assume the preferences to be complete. Lahijanian and Kwiatkowska [60] considered the problem of revising a given specification to improve the probability of satisfaction of the specification. They formulated the problem as a multi-objective MDP problem that trades off minimizing the cost of revision and maximizing the probability of satisfying the revised formula. Li et al [61] solve a preference-based probabilistic planning problem by reducing it to a multi-objective model checking problem. The only work that studies the problem of probabilistic planning with incomplete preferences was presented by Fu [62], in which she introduces the notion of the value of preference satisfaction for planning within a predefined finite time duration and developed a mixed-integer linear program to maximize the satisfaction value for a subset of preference relations.

Challenges for sequential decision making with incomplete preferences. The assumption of completeness has long been recognized to be restrictive [63, 64, 65]. When studying decision making for autonomous agents, the incompleteness about preferences may arise mainly due to

2 A deterministic strategy in a deterministic environment results in a unique path.

two reasons [66]: (i) *Tentative incompleteness*, which arises from an agent's inescapability or urgency of making a decision. For example, an autonomous vehicle must make a decision every 100ms based on whatever knowledge is available at that time, even if it does not have all the necessary information. (ii) *Assertive incompleteness*, which arises when the outcomes are incommensurate in value. That is, the agent lacks a common value function to compare the two outcomes. For example, in the trolley problem [67], an autonomous agent must decide between sacrificing one person versus sacrificing 5 people.

Incomplete preferences pose a fundamental challenge to rational decision making. For complete preferences, any planner based on the classical decision theory determines the "best" alternative by first constructing a utility representation of the preference and then using optimization theory to identify the alternative that yields the highest utility. Nevertheless, the existence of such a utility representation is not assured for incomplete preferences, except in the special case where the preference relation is continuous [64].

Another unique challenge that arises when investigating sequential decision making with incomplete preferences is the need to operate with combinative preferences. Combinative preferences allow the agent to express preferences over alternatives that may not be mutually exclusive. For example, consider a user preference for a robot that "visiting A is strictly preferred over visiting B." The two alternatives, 'visiting A' and 'visiting B,' are not mutually exclusive because, for instance, a path that visits A may also visit B. Sequential decision making with combinative preferences remains relatively unexplored within the existing literature. Given these challenges, we state the second research question considered in this dissertation.

Research Question II

In a single-player stochastic game on graph, *i.e.*, a Markov decision process, given a set of outcomes represented as ω-regular objectives, how to synthesize a strategy for P1 to achieve an outcome that maximally satisfies an incomplete preference over the given set of outcomes.

1.2 Contributions of this Dissertation.

This section provides an overview of the key contributions of this dissertation and outlines its structure. The material within this dissertation is based upon my previously published papers: Ch. 3 is based on [68, 4, 69][3], Ch. 4 on [70], Ch. 5 on [71, 72], and Ch. 6 is based on [73].

Ch. 2 surveys the basic definitions of various classes of games, hypergames, objectives, strategies, and various solution concepts.

Chapters 3-5 are dedicated to addressing Research Question I (RQI) for three sub-classes of games on graphs with exteroceptive incomplete information. In these games, P1 is presumed to possess complete information, while P2 might misperceive one of the components of the game. We categorize these games into three sub-classes based upon the specific game component misperceived by P2 .

1. *Misperception of labeling function*: P2 lacks information about P1's labeling function. This means the same outcome (*i.e.*, an infinite path) could be interpreted differently by P1 and P2.

2. *Misperception of action set*: P2 has incomplete information about P1's action capabilities.

3. *Misperception of specification*: P2 has lacks information about the true objective of P1.

One of the key contributions of this dissertation is the development of hypergames theory for sequential decision making in games on graphs. We introduce two models: a static hypergame on graph and a dynamic hypergame on graph. The static hypergame represents interactions where players' perceptions remain constant throughout the interaction, while the dynamic hypergame accommodates evolving perceptions of players as private information is revealed. Depending on the kind of the misperception involved, we introduce four solution concepts.

Ch. 3 investigates RQI when P2 is misinformed about P1's labeling function. In particular, Ch. 3.1 studies the synthesis problem when the interaction between P1 and P2 is modeled as a deterministic two-player turn-based game. The key contributions in this chapter include (i)

3 The content of Ch. 3.3 is based on the paper [69], which is presently under review.

Modeling: We show how to model the interaction as hypergame on graph, (ii) Solution concepts: We extend the notion of *stealthy deception*, commonly studied for normal-form and extensive-form games [74], to hypergames on graphs by defining two solution concepts: stealthy deceptive sure winning and stealthy deceptive almost-sure winning. The strategies synthesized under these concepts guarantee P1 to satisfy its ω-regular objective within finite number of steps or with probability one, respectively, while ensuring that P2 does not become aware of the information asymmetry until P1 can ensure to satisfy the temporal logic specification irrespective of P2's actions. These solution concepts for hypergames on graphs not only provide the provably-correct deceptive strategies for P1 but also provide a way to assess the effectiveness of deception and its potential limitations. (iii) Synthesis algorithm: We show that synthesizing winning strategies in the interaction under these concepts is equivalent to solving for sure and almost-sure winning strategies in the hypergame on graph. Thus, reducing the problem of synthesizing winning strategies in a game with incomplete information to that in game with complete and perfect information. (iv) Comparison between the concepts: We establish that may benefit more from deception when the game is analyzed under stealthy deceptive almost-sure winning condition as compared to when it is analyzed under stealthy deceptive sure winning condition.

In Ch. 3.3, we study a joint mechanism design and deceptive strategy synthesis problem. In this problem, we aim to allocate two types of deception resources, namely, traps and fake targets to disinform P2 about P1's true labeling function. In principle, the traps alter the structure of the game but do not affect P2's perception, whereas the fake targets manipulate P2's perception of the goal states in the game. Thus, by deciding the location of decoys P1 can influence P2's perception and, therefore, its behavior. To this end, we first specialize the *hypergame on graph* introduced in Ch. 3.2 to model the consequence of P1 allocating a subset of states in the reachability game as either "traps" or "fake targets" on P2's perception. Second, we analyze the effect of traps and fake targets on P2's behavior when players follow either greedy deterministic strategies, or randomized strategies. With greedy deterministic strategies, we show that *fake targets could be*

more advantageous than traps. Whereas, with randomized strategies, we find that *neither the fake targets nor the traps provide a greater benefit over the other*. Moreover, we observe that the benefit of using deception is greater when players use greedy deterministic strategies than when they use randomized strategies. This is a surprising result since, for several classes of games on graphs, randomized strategies are either equally or more powerful than the deterministic ones [9, 75]. Finally, we note that the task of determining an optimal placement of decoys that maximizes the size of the stealthy deceptive sure/almost-sure winning region poses a challenging combinatorial problem. To address this challenge and develop an algorithm with practical feasibility, we establish three key properties: (i) We demonstrate that the placement of traps and fake targets can be treated independently, as fake targets offer at least the same advantages as traps, (ii) Drawing insights from concepts in compositional synthesis [76, 77], we establish sufficient conditions under which the objective function (i.e., the size of the stealthy deceptive sure/almost-sure winning region) exhibits submodularity or supermodularity property, (iii) Leveraging these findings, we propose a greedy algorithm to incrementally place the decoys. The algorithm is $(1 - 1/e)$-optimal when the objective function is sub- or super-modular. This approach alleviates the need to exhaustively solve a large number of hypergames for all possible configurations of decoys.

Ch. 4 investigates the class of deterministic two-player turn-based games on graphs where P2 has incomplete information about P1's action capabilities. In this chapter, we introduce a different hypergame model, called a *dynamic hypergame*, which allows the perception of players to evolve during the game. Specifically, when P1 reveals a private action (*i.e.*, an action previously unknown to P2), P2 updates his perception of P1's action set and, thereby, his counter-strategy. In this setting, we consider the synthesis of a deceptive sure-winning strategy, *i.e.*, the strategy using which P1 can enforce satisfaction of its ω-regular reachability objective in *finitely many* steps by strategically revealing the private actions, and deceptive almost-sure winning strategy, *i.e.*, the strategy using which P1 can enforce satisfaction of its ω-regular reachability objective with probability one and, possibly, an undetermined number of steps by strategically revealing the

private actions. Note that P1's deceptive strategy cannot be stealthy in this case because P2's perception is allowed to evolve. We obtain two important results: (i) P1 gains no advantage by using deception under deceptive sure winning condition. That is, no state which is losing for P1 in the game with complete, symmetric information that becomes winning for P1 with the use of deception under sure winning condition. (ii) On the contrary, we establish that deception could be advantageous for P1 under the deceptive almost-sure condition. That is, there may exist a state which is losing for P1 in the game with complete, symmetric information that becomes winning for P1 with the use of deception under this almost-sure winning condition. We present an algorithm to synthesize the deceptive almost-sure winning strategy for P1 in the interaction.

In Ch. 5, we investigate the class of games on graphs where P2 has incomplete information about P1's true objective. We study the problem in two settings. In Ch. 5.1, we consider the problem of synthesizing stealthy deceptive strategies in deterministic two-player turn-based games on graphs, when P1 leverages P2's misinformation to its own advantage but does not influence P2's perception. We show a reduction from the problem of synthesizing stealthy deceptive almost-sure winning strategies to that of computing almost-sure winning strategies in a hypergame MDP representing the second-level hypergame modeling the interaction between P1 and P2. The reduction relies upon the characterization of the state-space of the hypergame MDP, which we show to contain up to five regions.

Ch. 5.2 considers the problem of synthesizing (non-stealthy) deceptive strategies for P1 in a stochastic two-player concurrent game on graph when P2 misperceives P1's true objective and P2's perception may evolve during the interaction. In this setting, we model the interaction as a dynamic hypergame on graph, where P2 is assumed to maintain a probability distribution over P1's possible objectives (*i.e.*, a set of LTL objectives). Our solution consists of two key modules, namely, *opponent modeling* and *deceptive planning*. Under the hypothesis that P2 uses a sub-goal inference mechanism to update its probability distribution, the opponent modeling enables P1 to track P2's perception given the history of their interaction. Thus, P1 can predict how its strategy will influence P2's perception and strategy. Then, we integrate the opponent model into deceptive

planning to compute a strategy that maximizes the probability of satisfying P1's true temporal logic objective.

Finally, Ch. 6 investigates Research Question II by introducing a novel automata-theoretic approach to qualitative planning in MDPs with incomplete preferences over temporal logic objectives. Our approach consists of three steps. First, we express incomplete preferences over the satisfaction of temporal goals specified using a fragment of LTL. Unlike propositional preferences that are interpreted over states, preferences over temporal goals are interpreted over infinite words. Second, we define an *automata-theoretic model* to capture the preferences over infinite words induced by the given preference relation over temporal logic formulas. Thirdly, we present an algorithm to solve preference-satisfying strategies in a stochastic system modeled as a labeled MDP. We introduce two new concepts, namely, *Safe and Positively Improving (SPI)* and *Safe and Almost-surely Improving (SASI)* strategies, that identify and exploit opportunities with positive probability and probability one, respectively. To synthesize SPI and SASI strategies, we introduce the idea of *improvement MDP* that distinguishes between opportunistic and non-opportunistic states. We prove that synthesizing SPI and SASI in labeled MDP is equivalent to synthesizing positive and almost-sure winning strategies in improvement MDP. Finally, we show that the synthesized SPI, SASI strategies indeed yield the feasible, most-preferred outcomes.

CHAPTER 2
BACKGROUND ON GAME AND HYPERGAME THEORY

In this chapter, we discuss the basic definitions of various classes of games, hypergames, objectives, strategies, and various solution concepts appearing in this thesis.

2.1 Games on Graphs

We start by defining the games on graphs and providing an overview of the established results and algorithms in this domain. First, we outline some preliminary notation: Given a finite set X, the powerset of X is denoted as $\wp(X)$. The set of all finite (resp., infinite) ordered sequences of elements from X is denoted by X^* (resp., X^ω). The set of all finite ordered sequences of length greater than 0 is denoted by X^+. We write $\mathcal{D}(X)$ to denote the set of probability distributions over X. The support of a distribution $D \in \mathcal{D}(X)$ is denoted by $\mathsf{Supp}(D) = \{x \in X \mid D(x) > 0\}$. The indicator function is defined to be $\mathbf{1}_X(x) = 1$ if $x \in X$ and 0 otherwise.

We consider several classes of games on graphs, namely, MDP (one-player stochastic games), deterministic two-player turn-based games on graphs, and stochastic two-player concurrent games on graphs. All these classes can be represented in a unified manner as defined below.

Definition 1 (Game on Graph)**.** A game on graph is a transition system [78], represented by the tuple,

$$G = \langle S, Act, T, s_0, AP, L \rangle,$$

where S is the set of states; Act is the set of actions; T is a transition function; $s_0 \in S$ is an initial state; AP is a set of atomic propositions; $L : S \to \wp(AP)$ is a labeling function that maps every state to the set of atomic propositions that hold in that state.

Hereafter, we refer to a game on graph as simply a game. The class of the game is determined by the nature of its transition function and whether players select actions simultaneously or in a turn-based fashion.

In any game, the transition function of a game may be either deterministic or probabilistic. A deterministic transition function $T : S \times Act \to S$ maps a pair of a state and an action to a unique

next state. A probabilistic transition function $T : S \times Act \to \mathcal{D}(S)$ maps a pair of a state and an action to a distribution over possible next states.

A two-player game is said to be *concurrent* if, at every state, both players simultaneously decide their next action without the knowledge of the choice made by the other player. Let Act_1 be the set of actions available to P1 and Act_2 be the set of actions available to P2. Then, the set of actions in a concurrent game can be represented as $Act = Act_1 \times Act_2$ and the transition function may be represented as $T : S \times Act_1 \times Act_2 \to S$. On the other hand, a two-player game is said to be *turn-based* if one player (P1 or P2) decides the next action at every state. In a turn-based game, the set of states can be partitioned into two disjoint sets S_1 and S_2 such that $S = S_1 \cup S_2$, where S_1 is the set of states where P1 chooses the next action and S_2 is the set of states where P2 chooses the next action. The transition function can be written as $T : (S_1 \times Act_1) \cup (S_2 \times Act_2) \to S$. In turn-based games, each player observes the consequence of the action selected by its opponent in the previous round. It is noted that MDPs are single-player games whose transition function is probabilistic.

Plays. A *play* in a game G is an ordered sequence of state-action pairs $\tau = s_0 a_0 s_1 a_1 \ldots$ such that, for every any integer $i \geq 0$, we have $s_{i+1} = T(s_i, a_i)$. A *path* in a game G is the projection of a trace τ onto the state space of the game: $\tau \downarrow_S = \rho = s_0 s_1 \ldots$. The set of all paths in the game is denoted by $\mathsf{Paths}(G)$ and the set of all finite prefixes of plays is denoted by $\mathsf{PrefPaths} = \{\rho[0:n] \mid \rho \in \mathsf{Paths}(G), n \geq 0\}$. Given any path ρ, the set of all states appearing in ρ is denoted by $\mathsf{Occ}(\rho) := \{s \in S \mid \exists i \geq 0 : \rho[i] = s\}$. Similarly, an *action-history* is the projection of a trace τ onto the set of actions: $\tau \downarrow_{Act} = \alpha = a_0 a_1 \ldots$. Given the labeling function L, every run ρ in G can be mapped to a word over an alphabet $\Sigma = \wp(AP)$ as $w = L(\rho) = L(s_0)L(s_1)\ldots$.

Strategies. A strategy determines the next action to be chosen by a player given a history. In concurrent games, a P1 strategy is a function $\pi_1 : S^+ \to \mathcal{D}(Act_1)$ that maps every non-empty finite sequence of states in $\mathsf{PrefPaths}$ to a probability distribution over the set of P1's action set Act_1. Whereas, in turn-based games, a P1 strategy is defined only for non-empty finite sequence of

states ending in a P1 state. Thus, it is represented as a function $\pi_1 : S^* S_1 \to \mathcal{D}(Act_1)$. A P2 strategy in concurrent and turn-based game is defined analogously.

Strategies can either be deterministic or randomized. A P1 or P2 strategy is said to be *deterministic* if, for all non-empty finite sequence of states $\rho \in \mathsf{PrefPaths}$ such that $\pi_i(\rho)$, $i = 1, 2$, is defined, $\pi_i(\rho)$ is a Dirac delta distribution. Otherwise, it is said to be *randomized*. A strategy is said to be *memoryless* if it only depends on the last state. Therefore, a memoryless P1 strategy in a concurrent game is a function $\pi_1 : S \to \mathcal{D}(Act_1)$. Whereas, a memoryless P1 strategy in a turn-based game is a function $\pi_1 : S_1 \to \mathcal{D}(Act_1)$. Deterministic, randomized, memory-based, memoryless strategies of P2 are defined analogously.

Outcomes of strategies. Consider a starting state $s_0 \in S$, a P1 strategy π_1 and a P2 strategy π_2. Then, a path $\rho = s_0 s_1 \ldots$ is said to be (π_1, π_2)-possible from state s_0 if for every $i \geq 0$ the following two conditions hold: if $s_i \in S_1$ then there exists an action $a \in Act_1$ such that $\pi_1(s_0 \ldots s_i)(a) > 0$ and $T(s_i, a)(s_{i+1}) > 0$; and if $s_i \in S_2$ then there exists an action $a \in Act_2$ such that $\pi_2(s_0 \ldots s_i)(a) > 0$ and $T(s_i, a)(s_{i+1}) > 0$. The set of all paths that are (π_1, π_2)-possible from state s_0 is denoted by $\mathsf{Outcomes}(s_0, \pi_1, \pi_2)$.

Winning strategies. In several chapters, we consider P1's objective to be a reachability objective, and therefore, an adversarial P2 who wants to prevent P1 from satisfying her reachability objective has a safety objective. In this case, we use a compact representation of the game.

Definition 2 (Reachability Game)**.** A game on graph in which P1 has a reachability objective is a tuple,

$$G = \langle S, Act, T, s_0, F \rangle,$$

where $S, Act, T, s_0 \in S$ have the same meanings as Def. 1; $F \subseteq S$ is the set of states that P1 must reach in order to satisfy its objective.

A reachability objective defines the winning set for P1 as

$\mathsf{Reach}(F) := \{\rho \in \mathsf{Paths}(G) \mid \mathsf{Occ}(\rho) \cap F \neq \emptyset\}$. Similarly, a safety objective of P2, which means that P2 must prevent P1 from visiting a final state in F, defines the winning set for P2 as $\mathsf{Safe}(F) := \{\rho \in \mathsf{Paths}(G) \mid \mathsf{Occ}(\rho) \cap F = \emptyset\}$. Given P1's strategy π_1 and P2's strategy π_2, we say P 1 wins the game if the outcome $\rho \in \mathsf{Outcomes}(s_0, \pi_1, \pi_2)$ satisfies $\rho \in \mathsf{Reach}(F)$. Otherwise, P2 wins the game.

Definition 3 (Sure Winning Strategy)**.** A P1 strategy π_1 is said to be *sure winning* at a state $s \in S$ in a game with the winning set $\mathsf{Win} \subseteq \Sigma^\omega$ for P1 if, for every P2 strategy π_2, we have $\mathsf{Outcomes}(s_0, \pi_1, \pi_2) \subseteq \mathsf{Win}$.

Definition 4 (Almost-sure Winning Strategy)**.** A P1 strategy π_1 is said to be *almost-sure winning* at a state $s \in S$ in a game with the winning set $\mathsf{Win} \subseteq \Sigma^\omega$ for P1 if, for every P2 strategy π_2, we have $Pr(\mathsf{Outcomes}(s_0, \pi_1, \pi_2) \cap \mathsf{Win} \neq \emptyset) = 1$.

Definition 5 (Postive Winning Strategy)**.** A P1 strategy π_1 is said to be *positive winning* at a state $s \in S$ in a game with the winning set Win for P1 if, for every P2 strategy π_2, we have $Pr(\mathsf{Outcomes}(s_0, \pi_1, \pi_2) \cap \mathsf{Win} \neq \emptyset) > 0$.

The winning strategies for P2 under the three solution concepts are defined similarly.

The set of states in the game G from which P1 (resp. P2) has a sure winning strategy is called the *sure-winning region* for P1 (resp. P2), denoted as $\mathsf{SWin}_1(G,F)$ (resp. $\mathsf{SWin}_2(G,F)$). Analogously, the set of states in the game G from which P1 (resp. P2) has an almost sure winning strategy is called the *almost sure winning region* for P1 (resp. P2), denoted as $\mathsf{ASWin}_1(G,F)$ (resp. $\mathsf{ASWin}_2(G,F)$). Lastly, the set of states in the game G from which P1 (resp. P2) has a positive winning strategy is called the *positive winning region* for P1 (resp. P2), denoted as $\mathsf{PWin}_1(G,F)$ (resp. $\mathsf{PWin}_2(G,F)$). The parameters (G,F) are dropped when they are clear from context.

A P2 strategy π_2 is said to be a *permissive under sure winning condition* if for any state $s \in \mathsf{SWin}_2$ and any action $a \in Act_2$ such that $\pi_2(s)(a) > 0$, we have $s' \in \mathsf{SWin}_2$ for any state $s' \in S$ such that $T(s,a)(s') > 0$. That is, by following a permissive strategy P2 is guaranteed to stay

within his sure winning region. The permissive strategies for P1 and P2 under sure winning, almost-sure winning and positive winning conditions are defined analogously.

In the case of deterministic two-player turn-based games, the following results are known.

Proposition 1 (Determinacy)**.** *From every state $s \in S$ in a deterministic two-player turn-based game, either P1 or P2 has a memoryless sure winning strategy to satisfy their reachability or safety objective, respectively. That is, for any deterministic two-player turn-based game, G, and a subset of states F,* $\mathsf{SWin}_1(G,F) \cup \mathsf{SWin}_2(G,F) = S$ *and* $\mathsf{SWin}_1(G,F) \cap \mathsf{SWin}_2(G,F) = \emptyset$.

Proposition 2 (Equivalence of Sure and Almost-sure Winning)**.** *In a deterministic two-player turn-based game, P1's sure and almost-sure winning regions coincide. That is, for any deterministic two-player turn-based game, G, and a subset of states F, we have* $\mathsf{SWin}_1(G,F) = \mathsf{ASWin}_1(G,F)$.

It follows from Proposition 1 and Proposition 2 that P2's sure winning and almost sure winning regions also coincide [17].

Synthesis algorithm for deterministic two-player turn-based game. The sure/almost-sure winning region of P1 in a reachability G can be computed by using Alg. 2-1. The algorithm constructs a sequence of sets, called level-sets, $Z_0, Z_1, \ldots, Z_K$ such that, from any state in $Z_k \setminus Z_{k-1}$, $k > 0$, P2 has a strategy to visit $Z_0 := F$ in no more than k steps. For any state $s \in Z_K$, we define its *rank* to be the minimum number of steps in which P2 can ensure a visit to F regardless of P1's strategy, denoted by $\mathsf{rank}_G(s)$. Thus, $\mathsf{rank}_G(s) = 0$ when $s \in F$, $\mathsf{rank}_G(s) = \min\{k \mid s \in Z_k\}$ when $s \in Z_K \setminus F$, and $\mathsf{rank}_G(s) = \infty$ when $s \notin Z_K$. The following properties of the level-sets constructed by Alg. 2-1 are known [9].

Proposition 3. *The following statements are true about the level-sets $Z_0, Z_1, \ldots, Z_K$ constructed by Alg. 2-1.*

1. $Z_0 \subseteq Z_1 \subseteq Z_2 \ldots \subseteq Z_K$.
2. *For any sets $F_1 \subseteq F_2 \subseteq S$, we have* $\mathsf{SWin}_1(G,F_1) \subseteq \mathsf{SWin}_1(G,F_2)$.

3. *For any sets $F_1, F_2 \subseteq S$, we have*

$$\mathsf{SWin}_1(G, F_1 \cup F_2) = \mathsf{SWin}_1(G, \mathsf{SWin}_1(G, F_1) \cup \mathsf{SWin}_1(G, F_2)).$$

Given the level-sets constructed by Alg. 2-1, a memoryless sure winning strategy of P2 can be constructed as follows: Given a P2 state $s \in Z_K \setminus F$, let $D_s = \{a \in A_2 \mid s' = T(s,a) \wedge \mathsf{rank}_G(s') < \mathsf{rank}_G(s)\}$ be the set of actions $a \in A_2$ for which the next state $s' = T(s,a)$ has a strictly smaller rank than s. Then, any deterministic strategy $\pi_2 : S \to A$ such that $\pi_2(s) \in D_s$ is a memoryless sure winning strategy for P2. Due to Proposition 1, the winning region of P1 is $S \setminus Z_K$. A deterministic memoryless strategy $\pi_1 : S \to A_1$ is sure winning for P1 at a P1 state $s \in S_1$ if $\pi_1(s) \in \{a \in A_1 \mid s' = T(s,a) \wedge s' \in S \setminus Z_K\}$.

Given the level-sets constructed by Alg. 2-1, a memoryless *almost-sure winning strategy* of P2 can be constructed as follows [9]: Given a P2 state $s \in Z_K \setminus F$, let $D_s = \{a \in A_2 \mid s' = T(s,a) \wedge s' \in Z_K\}$ be the set of actions $a \in A_2$ for which the next state $s' = T(s,a)$ is within the set Z_K. Then, any strategy $\pi_2 \in \Pi_2$ such that $\mathsf{Supp}(\pi_2(s)) = D_s$ is a memoryless almost-sure winning strategy for P2. Similarly, given any P1 state $s \in S \setminus Z_K$, any strategy $\pi_1 \in \Pi_1$ such that $\mathsf{Supp}(\pi_1(s)) = \{a \in A_1 \mid s' = T(s,a) \wedge s' \in S \setminus Z_K\}$ is almost-sure winning for P2.

Algorithm 2-1 Zielonka's recursive algorithm to compute sure winning region in a reachability game.

function $\mathsf{SWin}_1(G, F)$
  $Z_0 \leftarrow \mathcal{F}, k \leftarrow 0$
  repeat
    $\mathsf{Pre}_1(Z_k) \leftarrow \{v \in V_1 \mid \forall a \in A_1 : \Delta(v,a) \in Z_k\}$
    $\mathsf{Pre}_2(Z_k) \leftarrow \{v \in V_2 \mid \exists a \in A_2 : \Delta(v,a) \in Z_k\}$
    $Z_{k+1} = Z_k \cup \mathsf{Pre}_1(Z_k) \cup \mathsf{Pre}_2(Z_k)$
    $k \leftarrow k+1$
  until $Z_k \neq Z_{k-1}$
  return Z_k
end function

2.2 Temporal Logic and Automata

Linear Temporal Logic (LTL). Since we are interested in infinite-duration games, we focus on ω-regular objectives. Specifically, in some chapters, we use LTL formulas [79] to define the objectives of P1 and P2. Formally, an LTL formula is defined inductively as

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \bigcirc\varphi \mid \varphi \,\mathsf{U}\, \varphi,$$

where $p \in AP$ is an atomic proposition, $\neg$ (negation), $\wedge$ (and), and $\vee$ (or) are Boolean operators, and $\bigcirc$ (next), U (strong until) and W (weak until) are temporal operators. A formula $\bigcirc\varphi$ means that the formula φ will be true in the next state. A formula $\varphi_1 \,\mathsf{U}\, \varphi_2$ means that φ_2 will be true in some future time step, and before that φ_1 holds true for every time step. We define two additional temporal operators: $\Diamond$ (eventually) and $\Box$ (always) as follows: $\Diamond\varphi = \top \,\mathsf{U}\, \varphi$ and $\Box\varphi = \neg\Diamond\neg\varphi$.

Syntactically co-safe LTL formulas. Sometimes, we restrict the specifications of the players to the class of scLTL [10]. An scLTL formula contains only $\Diamond$, $\bigcirc$, and U temporal operators when written in a positive normal form (*i.e.*, the negation operator $\neg$ appears only in front of atomic propositions). A unique property of scLTL formulas is that a word satisfying an scLTL formula φ only needs to have a *good prefix*. That is, given a good prefix $w \in \Sigma^*$, the word $ww' \models \varphi$ satisfies the scLTL formula φ for any $w' \in \Sigma^\omega$. The set of good prefixes can be compactly represented as the language accepted by a deterministic finite automaton (DFA) defined as follows.

Definition 6 (Deterministic Finite Automaton)**.** A deterministic finite automaton (DFA) is a tuple,

$$\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F \rangle,$$

where Q is the set of states; $\Sigma := \wp(AP)$ is the alphabet; $\delta : Q \times \Sigma \to Q$ is a deterministic transition function; $q_0 \in Q$ is the initial state; and $F \subseteq Q$ is the set of final states.

For a finite word $w = \sigma_0\sigma_1 \ldots \sigma_n \in \Sigma^*$, the DFA generates a sequence of states $q_0 q_1 \ldots q_{n+1}$

such that $q_0 = \iota$ and $q_{i+1} = \delta(q_i, \sigma_i)$ for any $0 \leq i \leq n$. The word w is accepted by the DFA if and only if $q_{n+1} \in F$. The set of words accepted by the DFA $\mathcal{A}$ is called *its language*. Given P1's objective expressed as an scLTL formula φ, the set of good prefixes of words corresponding to φ is accepted by a DFA, which has a special property that all final states are sink states. Thereby, if a finite prefix of an infinite run reaches a final state, it is ensured that the "last" state will be a final state and the word, corresponding to this run, is accepted. We assume that the DFA is complete—that is, for every state-action pair (q, σ), $\delta(q, \sigma)$ is defined. An incomplete DFA can be made complete by adding a sink state q_{sink} such that $\forall \sigma \in \Sigma, \delta(q_{\mathsf{sink}}, \sigma) = q_{\mathsf{sink}}$, and directing all undefined transitions to the sink state q_{sink}.

A path ρ in a game G is said to satisfy an LTL formula φ, if the labeling sequence $L(\rho)$ satisfies the formula φ, *i.e.*, $L(\rho) \models \varphi$. Given this relation, a game in which P1 has an scLTL objective can be equivalently represented by a reachability game constructed in the following way.

Definition 7 (Product Game)**.** Given a game on graph G, let φ be an scLTL formula representing P1's objective, and $\mathcal{A}$ be the DFA representing the language of φ. Then, the product of the game G with the DFA $\mathcal{A}$, is the a reachability game,

$$\widehat{G} = \langle \widehat{S}, Act, \widehat{T}, \widehat{s_0}, \widehat{F} \rangle,$$

where $\widehat{S} = S \times Q$ is the set of states; Act is the set of actions; $\widehat{s_0} = (s_0, L(s_0))$ is the initial state; and $\widehat{F} = S \times F$ is the set of final states. The transition function is defined as follows: If the transition function of G is deterministic, then $\widehat{T}((s,q),a) = (s',q')$ if and only if $T(s,a) = s'$ and $\delta(q, L(s')) = q'$. If the transition function of G is probabilistic, then $\widehat{T}((s,q),a)(s',q') = T(s,a)(s')$ if $\delta(q, L(s')) = q'$, and $\widehat{T}((s,q),a)(s',q') = 0$, otherwise.

2.3 Hypergame Theory

A hypergame [27] is a model used to capture strategic interactions when players have incomplete information. Intuitively, a hypergame is a game of games, and each game is associated with a player's subjective view of its interaction with other players based on its own information

and information about others' subjective views. Hypergames are defined inductively based on the level of perception of individual players. A level-0 (L0) hypergame is a game with complete, symmetric information, where the perceptual games of both players are identical to the true game. In a level-1 (L1) hypergame, at least one of the players, say P2, misperceives the true game, but neither is aware of it. In this case, both players believe their perceptual game to be the true game and play according to their perceptual games, which are level-0 hypergames. In a level-2 (L2) hypergame, one of the players becomes aware of the misperception and is able to reason about its opponent's perceptual game.

Definition 8 (Level-1 and Level-2 Hypergame)**.** Given the true game known to P1 G_1 and P2's perceptual game G_2, the level-1 (L1) hypergame is defined as a tuple $H^1 := \langle G_1, G_2 \rangle$. The level-2 (L2) hypergame between P1 and P2 is the tuple,

$$H^2 = \langle H^1, G_2 \rangle.$$

In L2-hypergame, P1 is aware of P2's misperception, but P2 remains unaware that it lacks information. Consequently, P2 computes its strategy by solving its perceptual game G_2. P1 decides its strategy by solving the L1-hypergame H^1, which allows P1 to incorporate P2's strategy as computed in G_2 into its decision-making.

We now discuss the solution concepts of hypergames. Given that different players may have different perceptions (*i.e.*, subjective views) of the utility functions in a hypergame, let u_i^j denote the utility function of player i perceived by player j.

Definition 9 (Subjective Rationalizability [43])**.** Given a L2 hypergame $H^2 = \langle H^1, G_2 \rangle$, strategy $\pi_i^{*,2}$ is subjective rationalizable for P2 if and only if it satisfies, for all $\pi_i \in \Pi_i$,

$$u_i^2(h, \pi_i^{*,2}, \pi_j^{*,2}) \geq u_i^2(h, \pi_i, \pi_j^{*,2}),$$

where $(i, j) \in \{(1,2),(2,1)\}$. The strategy $\pi_1^{*,1}$ is subjective rationalizable for P1 if and only if it

satisfies, for all $\pi_1 \in \Pi_1$,

$$u_1^1(h, \pi_1^{*,1}, \pi_2^{*,2}) \geq u_1^1(h, \pi_1, \pi_2^{*,2}),$$

where $\pi_2^{*,2}$ is subjective rationalizable for P2.

In words, a strategy $\pi_i^{*,i}$ is called subjective rationalizable for player i if in player i's subjective view, it is the best response to player j's best response $\pi_j^{*,i}$, which is computed from player i's perceptual game. A pair of subjective rationalizability (SR) strategies $\langle \pi_1^{*,1}, \pi_2^{*,2} \rangle$ is called the best-response equilibrium of the hypergame H^2. In L2 hypergame, P2's strategy is subjective rationalizable if it is rationalizable in P2's perceptual game G_2. P1's strategy is subjective rationalizable if it is the best response to P2's subjective rationalizable strategy.

CHAPTER 3
SYNTHESIS WITH MISPERCEPTION OF LABELING FUNCTION

This chapter investigates the synthesis of deceptive winning strategies for the sub-class of games with incomplete information where P2 misperceives P1's labeling function. The labeling function enables a player to interpret an outcome, *i.e.*, an infinite sequence of game states, to evaluate whether it satisfies its ω-regular objective. Therefore, when P2 misperceives P1's labeling function, P2 may not correctly distinguish between a winning and a losing outcome. For instance, imagine the case where P2 mislabels an unsafe state as a target state. This provides P1 an opportunity to leverage P2's misperception and enforce a winning outcome from an otherwise P1's losing state, *i.e.*, a state from which P2 has a winning strategy if it had complete information.

The chapter contains two sections. The first section develops the theoretical foundations of analyzing the aforementioned class of games. It introduces a static hypergame on graph model and the solution concepts of stealthy deceptive sure winning and stealthy deceptive almost-sure winning to analyze the rational behavior of players in the hypergame. The second section applies the developed theory to solve the decoy placement problem (also known as the honeypot allocation problem [80, 81]) in cybersecurity, which asks to place deception resources in a network to disinform P2 about P1's labeling function and leverage it to synthesize a deceptive strategy for P1 to maximize its winning region.

3.1 Effect of Labeling Misperception

Consider an interaction between P1 and P2 characterized by a deterministic two-player turn-based zero-sum game, $G = \langle S, Act, T, s_0, AP, L \rangle$, as defined in Def. 1. In this interaction, P1's objective is to satisfy an scLTL formula φ while P2's objective is to prevent P1 from satisfying her objective. However, we assume that P1 and P2 play with different labeling functions. Specifically, the information structure is as follows:

Assumption 1 (Information Structure)**.** The components S, Act, T and AP of the game G, and P1's objective φ are known to both players P1 and P2. P1 has complete information about the labeling function, that is, she knows the true label $L(s)$ of every state $s \in S$. P2 has incomplete

information about P1's labeling function: There exists at least one state $s \in S$ such that $L_2(s) \subseteq L(s)$. P1 knows P2's perceived labeling function L_2.

Perceptual games. As a result of Assumption 1, P1 and P2 have different perceptions of the game arena. P1 knows the true game arena G whereas P2 knows the arena with a different labeling function, say $G_2 = \langle S, Act, T, AP, L_2 \rangle$. Hence, P1 and P2 play different games in their minds. Since P1 knows her labeling function, she constructs a perceptual game as the product $G \otimes \mathcal{A}$, where $\mathcal{A}$ is the DFA representing the language of scLTL formula φ. On the other hand, P2 constructs his perceptual game as the product $G_2 \otimes \mathcal{A}$.

Notation 1. Given a labeling function L, let $G(L)$ denote the deterministic two-player turn-based game on a graph in which the labeling function is L.

Abusing the notation, we will write P1's perceptual game $G(L)$ to represent the product game $G \otimes \mathcal{A}$. Similarly, P2's perceptual game is denoted by $G(L_2)$, which represents the product game $G_2 \otimes \mathcal{A}$.

As a consequence of misperception, there exist paths $\rho = s_0 s_1 \ldots$ in the game arena that are interpreted differently by P1 and P2. Specifically, P1's interpretation is $L(\rho) = L(s_0)L(s_1)\ldots$ whereas that of P2 is $L_2(\rho) = L_2(s_0)L_2(s_1)\ldots$. Because of this the paths induced by ρ in the DFA representing the language of scLTL formula φ are different. Thus, P1 may *mislead* or *deceive* P2 on how much progress has been made towards satisfying φ by strategically visiting those states $s \in S$ where $L(s) \neq L_2(s)$.

A necessary condition for P1 to succeed at deception is to ensure that P2 does not learn about his misperception. Assuming that both players expect their opponent to be rational, P2 would learn about his misperception if P1 acts in a way that P2 considers irrational. A P1's deceptive strategy that prevents P2 from beecoming aware of his misperception is called a *stealthy deceptive strategy* (formalized in Def. 11). We now state our problem statement.

Problem 1. Consider an interaction between P1 and P2 under Assumption 1 where the true game arena is G, P2's perceived game arena is G_2, and P1's objective is to satisfy φ. Then, determine

the stealthy deceptive strategy using which P1 can satisfy φ under the qualitative solution concepts of sure and almost-sure winning.

3.2 Static Hypergame on Graph

Given that P1 and P2 play different perceptual games, their interaction can be modeled as a hypergame. Following the discussion in Sec. 2.3, the first-level hypergame representing the interaction between P1 and P2 is given by $H^1 = \langle G(L), G(L_2) \rangle$. Since P1 is aware that φ_2 is her private information, she is also aware that P2 misperceives her true objective. Therefore, their interaction is, in fact, a second-level hypergame.

$$H^2 = \langle H^1, G(L_2) \rangle. \tag{3-1}$$

We now define a graphical model of the hypergame H^2 that incorporates the superior knowledge of P1. Using this model, we can compute P2's subjectively rationalizable strategy and use it to synthesize a *stealthy deceptive* strategy for P1.

Definition 10. Given the perceptual games $G(L)$ and $G(L_2)$, a *hypergame on a graph* is a deterministic two-player turn-based game on a graph,

$$\mathcal{H} = \langle V, Act, \Delta, v_0, \mathcal{F} \rangle,$$

where

- $V := S \times Q \times Q$ is the set of states;
- $\Delta : V \times Act \to V$ is a deterministic transition function such that given two states $(s,q,p), (s',q',p') \in V$ and an action $a \in Act$, we have $(s',q',p') = \Delta((s,q,p),a)$ if and only if $s' = T(s,a)$ and $q' = \delta(q, L(s'))$ and $p' = \delta(p, L_2(s'))$;
- $v_0 \in V$ is an initial state.
- $\mathcal{F} = \mathsf{ASWin}(G(L), S \times F) \times Q$ is the set of final states.

In a hypergame on a graph, a state $v = (s, q, p) \in V$ allows P1 to track the progress q that is *truly* made towards satisfying the objective as well as the progress p that P2 thinks has been made towards satisfying the objective. This is because the component q evolves according to P1's perceptual game $G(L)$ whereas the component p evolves according to P2's perceptual game $G(L_2)$. The set of final states is defined, intuitively, to signify that P1's objective in $\mathcal{H}$ is to visit a winning state in her perceptual game regardless of what P2's perception is.

We now formalize the notion of stealthy deceptive strategy that leverages P2's misperception but ensures that P2 remains unaware of her misperception.

Definition 11 (Stealthy Deceptive Winning Strategy)**.** A memoryless, randomized strategy $\pi : V \to \mathcal{D}(Act_1)$ is said to be *stealthy deceptive sure (resp., almost-sure) winning* in the hypergame $\mathcal{H}$ if the following two conditions hold: (a) *Stealthy*: For any $v \in V \setminus \mathsf{ASWin}_1(G(L), F) \times Q$, $\pi(v, a) > 0$ only if action a is subjectively rationalizable for P1 in $G(L_2)$; (b) *Winning*: Given any state $v \in V$ and any subjectively rationalizable strategy μ of P2, for every run $\rho \in \mathsf{Outcomes}(v, \pi, \mu)$ we have $\mathsf{Occ}(\rho) \cap \mathcal{F} \neq \emptyset$ (resp., $Pr(\mathsf{Occ}(\rho) \cap \mathcal{F} \neq \emptyset) = 1$).

A state $v \in \mathcal{H}$ is said to be *stealthy deceptive sure (resp., almost-sure) winning* if P1 has a stealthy deceptive sure (resp., almost-sure) winning strategy from that state. The set of all stealthy deceptive sure (resp., almost-sure) states is called the stealthy deceptive sure (resp., almost-sure) winning region.

3.2.1 Stealthy Deceptive Sure Winning Strategy

In this section, we reduce the problem of synthesizing a stealthy deceptive sure winning strategy to that of synthesizing a sure winning strategy in a deterministic two-player turn-based game on a graph. Our idea is to construct a game on a graph that includes only those actions that are subjectively rationalizable from P2's perspective. The following result provides a way to characterize the subjectively rationalizable actions.

Lemma 3-1. *Given a state $v = (s, q, p) \in V$ and an action $a \in Act$, let $v' = (s', q', p') = \Delta(v, a)$. Then, the action a is subjectively rationalizable at the state v if one of the following conditions hold:*

(a) The states (s,p) and (s',p') are both P1's sure winning states in P2's perceptual game $G(L_2)$.

(b) The states (s,p) and (s',p') are both P2's sure winning states in P2's perceptual game $G(L_2)$.

(c) If neither (a) nor (b) holds, then the action a is subjectively rationalizable at the state v.

Intuitively, the conditions (a) and (b) assert that P2 thinks that a rational player, from a winning state, will select a winning action which allows the player to stay within their winning region. From a losing state, any action is considered subjectively rationalizable because the player knows that they have lost the game.

Notation. We denote P1 and P2's subjectively rationalizable strategies in P2's game by $\pi_i^2 : V \to 2^{Act}$, for $i = 1,2$. For a P1 state $v = (s,q,p)$,

$$\pi_1^2(v) = \{a \in Act_1 \mid \Delta_2((s,p),a) \in \mathsf{SWin}_1(G(L_2),F)\}, \tag{3-2}$$

where T_2 is the transition function of P2's perceptual game $G(L_2)$ and $\mathsf{SWin}_1(G(L_2),F)$ is P1's sure winning region in $G(L_2)$. P2's subjectively rationalizable strategy π_2^2 is defined analogously.

Next, we define the game on a graph that excludes non-subjectively rationalizable P1 actions.

Definition 12. Given the hypergame on a graph $\mathcal{H}$ and the subjectively rationalizable strategies π_1^2, π_2^2 of P1 and P2 in P2's perceptual game $G(L_2)$, the game on a graph excluding non-subjectively rationalizable P1 actions is the tuple,

$$\widehat{\mathcal{H}} = (V, Act, \widehat{\Delta}, v_0, \mathcal{F})$$

where the transition function $\widehat{\Delta}$ is obtained from Δ by restricting both players' actions as follows: For any state $v = (s,q,p) \in V$ and any action $a \in Act$,

- If $(s,q) \in \mathsf{SWin}_1(G(L),F)$, then $\widehat{\Delta}(v,a) = \Delta(v,a)$.

- If $(s,q) \notin \mathsf{SWin}_1(G(L),F)$ and $(s,p) \in \mathsf{SWin}_2(G(L_2),F)$, then $\widehat{\Delta}(v,a) = \uparrow$ if $s \in S_2$ and $\pi_2^2(v)(a) = 0$. Otherwise, $\widehat{\Delta}(v,a) = \Delta(v,a)$.

- If $(s,q) \notin \mathsf{SWin}_1(G(L),F)$ and $(s,p) \in \mathsf{SWin}_1(G(L_2),F)$, then $\widehat{\Delta}(v,a) = \uparrow$ if $s \in S_1$ and $\pi_1^2(v)(a) = 0$. Otherwise, $\widehat{\Delta}(v,a) = \Delta(v,a)$.

The set of final states $\mathcal{F} = \{(s,q,p) \in V \mid (s,q) \in \mathsf{SWin}_1(G(L),F) \text{ and } p \in Q\}$–that is, P1 satisfied her objective in $G(L)$ by visiting any state in $\mathcal{F}$.

Theorem 3-1. *Given a state $v \in V$, P1 has a* stealthy deceptive sure winning *strategy at the state v in hypergame $\mathcal{H}$ if and only if she has a* sure winning *strategy at the state v in the game $\widehat{\mathcal{H}}$.*

Proof. Before reaching the set $\mathsf{SWin}_1(G(L),F) \times Q$, at any state (s,q,p) where $s \in S_2$, if (s,p) is perceived winning by P2 (*i.e.*, $(s,p) \in \mathsf{SWin}_2(G(L_2),F)$), then P2 will select a subjectively rationalizable action $a \in \pi_2^2(s,p)$. If (s,p) is not in $\mathsf{SWin}_2(G(L_2),F)$, then any action from P2 is subjective rationzalizable. At a state (s,q,p) where $s \in S_1$, if $(s,p) \in \mathsf{SWin}_1(G(L_2),F)$ but $(s,q) \notin \mathsf{SWin}_1(G(L),F)$, then P1 will select a subjectively rationalizable action $a \in \pi_1^2(s,p)$ so as not to contradict P2's perception. If $(s,p) \notin \mathsf{SWin}_1(G(L_2),F)$ and $(s,q) \notin \mathsf{SWin}_1(G(L),F)$, then any action of P1 is deemed subjectively rationalizable by P2. The solution of reachability game $\widetilde{H}$, is a policy $\pi_1^* : S \times Q \times Q \to A_1$ that ensures starting from a state where π_1^* is defined, *no matter which action P2* selects in $\tilde{H}$, P1 can ensure to reach a state (s,q,p) with $(s,q) \in \mathsf{SWin}_1(G(L),F)$ by following π_1^*, in finitely many steps. By construction, P2 will not know that a misperception exists as P1 takes only subjective rationalizable actions, until P1 reaches $\mathsf{SWin}_1(G(L),F)$. After reaching the set, P1 can follow the true winning strategy defined for $\mathsf{SWin}_1(G(L),F)$. □

3.2.2 Stealthy Deceptive Almost-Sure Winning Strategy

P1's stealthy deceptive sure winning strategy is robust against any deterministic subjectively rationalizable strategy of P2. When synthesizing stealthy deceptive almost-sure winning strategy, we assume that the players use randomized strategy. In this case, we are interested to know whether the use of randomized strategy is more advantageous than using a deterministic strategy. We start by making a reasonable assumption on P2's strategy.

Assumption 2. For a P2 state $v \in V$ in the $\mathcal{H}$, any subjectively rationalizable action at (s,p) in P2's perceptual game $G(L_2)$ is selected by P2 with a non-zero probability.

Because of Assumption 2, P2 can be treated as a random player who chooses an subjectively rationalizable action during every turn. This reduces the hypergame $\mathcal{H}$ to a MDP defined below.

Definition 13. Given the hypergame on a graph $\mathcal{H}$ and the randomized subjectively rationalizable strategies π_1^2, π_2^2 of P1 and P2 in P2's perceptual game $G(L_2)$, the MDP is a tuple,

$$\widetilde{\mathcal{H}} = (V_1, Act_1, \widetilde{\Delta}, v_0, \mathcal{F}),$$

where

- $V_1 = S_1 \times Q \times Q$ is the subset of hypergame states at which P1 chooses an action.
- Act_1 is the set of P1's actions.
- $v_0 \in V_1$ is an initial state.
- $\widetilde{\Delta} : V_1 \times Act_1 \to \mathcal{D}(V_1)$ is defined as follows: For any state $v = (s,q,p) \in V_1$ and an action $a \in Act_1$, let $v' = (s',q',p') = \Delta(v,a)$,
  - If $v \in \mathsf{ASWin}_2^2$ then $Pr(v'' \mid v,a) > 0$ for every state $v'' = (s'',q'',p'') \in V_1$ if there exists a P2's subjectively rationalizable action $b \in Act_2$ such that $\pi_2^2((s',p'))(b) > 0$ and $v'' = \Delta(v',b)$.
  - If $v \in \mathsf{ASWin}_1^2$ then $Pr(v'' \mid v,a) > 0$ for every state $v'' = (s'',q'',p'') \in hgameState_1$ if action a is subjectively rationalizable at the state (s,p) in P2's perceptual game $G(L_2)$, *i.e.*, $\pi_1^2((s,p))(a) > 0$, and there exists a P2's subjectively rationalizable action $b \in Act_2$ such that $\pi_2^2((s',p'))(b) > 0$ and $v'' = \Delta(v',b)$.

The set of final states $\mathcal{F} = \{(s,q,p) \in V \mid (s,q) \in \mathsf{ASWin}_1^1 \text{ and } p \in Q\}$–that is, P1 satisfied her objective in $G(L)$ by visiting any state in $\mathcal{F}$.

Theorem 3-2. *Given a state $v \in V_1$, P1 has a* stealthy deceptive almost-sure winning *strategy at the state v in hypergame $\mathcal{H}$ if and only if she has a* almost-sure winning *strategy at the state v in the game $\widetilde{\mathcal{H}}$.*

The proof is similar to that of Thm. 3-2.

3.3 Decoy Allocation Problem

In this section, we consider a joint deception resource allocation and deceptive strategy synthesis problem for a class of games on graphs with incomplete information. We consider a subclass of games on graphs called reachability games that represent a sequential interaction between two players, namely, a defender (P1) and an attacker (P2). The attacker's objective is to reach a set of target states, while that of the defender is to prevent the attacker from reaching a target state. Employing the solutions of zero-sum reachability games [9], we can identify a set of states from which P1 has no strategy to prevent P2 from visiting a true target. To protect targets when the game starts from a P1's losing position, P1 can allocate deception resources to disinform the attacker and further synthesize a deceptive strategy that exploits the attacker's misinformation to prevent it from reaching the target states. We consider two classes of deception resources that serve the functions of *hiding the real* and *reveal the fiction* [82]. Hiding the real refers to the defender simulating a trap to function like a real state while revealing the fiction corresponds to camouflaging a state to look like a target state for the attacker. Given this setup, we are interested in the following problem: *How to optimally allocate the decoys so that the defender can influence the attacker into taking (or not taking) certain actions that maximize the defender's deceptive winning region?*

3.3.1 Modeling and Problem Formulation

We consider the class of interactions between P1 and P2 characterized by the following information structure.

Assumption 3 (Information Structure)**.** P1 knows the true game, *i.e.*, the locations and types of all decoys. P2 is unaware of the presence of decoys. P1 knows about P2's unawareness.

In a game with incomplete information satisfying Assumption 3, the players perceive their interaction differently. P1 has complete information about the location and type of the decoys and, therefore, knows the true game.

Definition 14 (True Game)**.** Given a base game $G = \langle S, A, T, s_0, F \rangle$, let X and Y be two subsets of $S \setminus F$ such that $X \cap Y = \emptyset$. The deterministic two-player turn-based game representing the *true* interaction between P1 and P2 when the states in X are allocated as traps and those in Y are allocated as fake targets is the tuple,

$$G^1_{X,Y} = \langle S, A, T_{X,Y}, s_0, F \rangle,$$

where

- S, A, s_0 and F are defined as in Def. 2;
- $T_{X,Y}$ is a deterministic transition function. Given any state $s \in S$ and any action $a \in A$,

$$T_{X,Y}(s,a) = \begin{cases} T(s,a) & \text{if } \ s \notin X \cup Y \\ s & \text{otherwise} \end{cases}$$

Note that the states in G which are allocated as decoys are 'sink' states in $G^1_{X,Y}$. Hereafter, we reserve the symbols X, Y to represent traps and fake targets.

On the other hand, P2 is unaware of the presence of decoys. Therefore, in its subjective view of the game, P2 does not mark the states in $X \cup Y$ as sink states; instead, it considers the states in Y to be goal states.

Definition 15 (P2's Perceptual Game)**.** Given a base game $G = \langle S, A, T, s_0, F \rangle$, a set X of traps and a set Y of fake targets, P2's perceptual game is the tuple

$$G^2_{X,Y} = \langle S, A, T, s_0, F \cup Y \rangle,$$

where

- S, A, T, s_0 have the same meanings as Def. 2;
- $F \cup Y$ is a set of goal states as perceived by P2.

Remark 1. When P1 places no fake targets, *i.e.*, $Y = \emptyset$, we have $G^2_{X,Y} = G$.

Given the information structure in Assumption 3, we consider the following problem:

Problem 2. Let $G = \langle S, A, T, s_0, F \rangle$ be a reachability game. Determine the subsets $X, Y \subseteq S \setminus F$ of traps and fake targets that maximize the number of states from which P1 has (i) a sure winning strategy, (ii) an almost-sure winning strategy, to prevent P2 from reaching F, taking into account P2's incomplete information and subject to the constraints that $|X| \leq M$, $|Y| \leq N$ and $X \cap Y = \emptyset$.

We introduce a running example to illustrate the key insights derived in this paper.

Example 1 (Running Example)**.** Consider the game depicted in Fig. (3-1). The game consists of 12 states, where circular states represent P1 states and square states represent P2 states. The final states s_0 and s_1 are indicated by a double boundary. In this game, P2 aims to reach either state s_0 or s_1.

To determine the winning regions of the players, Alg. 2-1 is applied to G with the set of final states $F = \{s_0, s_1\}$. The colors assigned to the states in the figure correspond to the result of this algorithm. Blue-colored states represent the sure/almost-sure winning region of P1, while red-colored states represent the corresponding region of P2. Additionally, the rank of each state is indicated by its column placement. A state belonging to a column with rank $k = n$ possesses a rank equal to n. For instance, the states s_5, s_6 have a rank of 2, while the state s_9 has a rank of 5. The states s_{10}, s_{11} that constitute P1's sure winning region have rank $+\infty$.

3.3.2 P2's Subjectively Rationalizable Strategy

In this subsection, we discuss the effect of decoys on P2's winning region and the subjectively rationalizable strategies in its perceptual game.

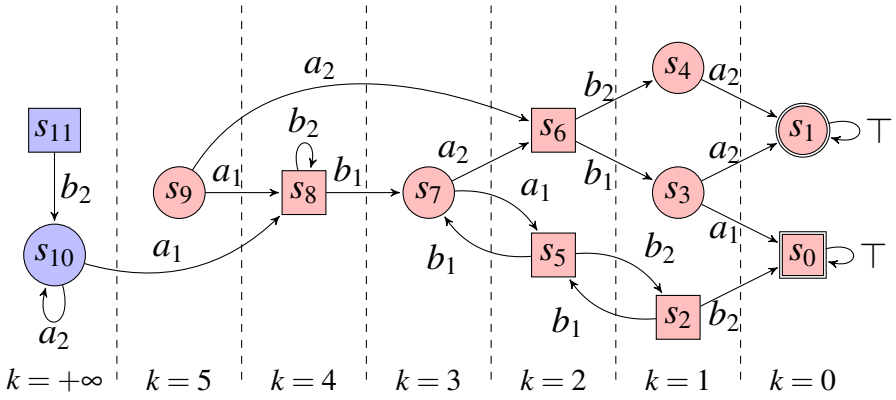


Figure 3-1. Base game considered in the running example.

As discussed in Rmk. 1, traps do not impact P2's perception. Consequently, traps do not influence the winning regions of the players in P2's perceptual game, nor do they affect P2's subjectively rationalizable strategy. Therefore, in this subsection, we focus on the case when Y is a non-empty subset of $S \setminus F$ and P2's objective in $G^2_{X,Y}$ is to reach $F \cup Y$.

We first introduce a lemma that captures the effect of making a subset of states in $\mathsf{SWin}_2(G,F) \setminus F$ to be P2's goal states. The lemma will aid us in proving Proposition 4, which summarizes the effect of decoys on the size of winning regions of P1 and P2 as perceived by P2. The lemma is general and holds for any reachability game.

Lemma 3-2. *Let $G = \langle S, A, T, s_0, F \rangle$ be a game as per Def. 2. Given any $Y \subseteq \mathsf{SWin}_2(G,F) \setminus F$, let $G^2_{\emptyset,Y} = \langle S, A, T_{\emptyset,Y}, s_0, F \cup Y \rangle$ be a game in which a subset of P2's winning region is marked as final states in addition to F. Then, the rank of any state $s \in \mathsf{SWin}_2(G,F)$ in $G^2_{\emptyset,Y}$ is less than or equal to its rank in G.*

Proof. Recall that the rank $\mathsf{rank}_G(s)$ of a state $s \in \mathsf{SWin}_2(G,F)$ in game G is the smallest number of steps in which P2 can ensure a visit to F, regardless of the deterministic strategy followed by P1. By definition, in game G, every path in $\mathsf{Outcomes}_G(s, \pi_1, \pi_2)$ from any state $s \in \mathsf{SWin}_2(G,F)$ is ensured to visit F for any valid P1 strategy π_1 and any sure winning strategy π_2 of P2. Since, in game $G^2_{\emptyset,Y}$, the presence of fake targets does not affect the transitions from any state except those in Y and all states in $F \cup Y$ are sink states, two possibilities arise for any path $\rho \in \mathsf{Outcomes}_{G^2_{\emptyset,Y}}(s, \pi_1, \pi_2)$: either ρ visits Y before visiting F, or ρ visits F without visiting Y.

In both cases, the number of steps required to visit $F \cup Y$ is at most $\mathsf{rank}_G(s)$. The rank of s in $G^2_{\emptyset,Y}$ is strictly smaller than $\mathsf{rank}_G(s)$ when P2 has a sure winning strategy from s to visit Y. □

Since the presence of traps does not affect P2's perception, it does not affect the ranks of the states. Hence, Lma. 3-2 extends naturally to games containing both types of decoys.

Corollary 1. *For any state $s \in \mathsf{SWin}_2(G,F)$, its rank in $G^2_{X,Y}$ is less than or equal to its rank in G.*

We now introduce a proposition to summarize the effect of decoys on the size of winning regions of P1 and P2 as perceived by P2.

Proposition 4. *The following statements about $G^2_{X,Y}$ are true.*

(a) If $Y \subseteq \mathsf{SWin}_1(G,F)$, then $\mathsf{SWin}_1(G^2_{X,Y}, F \cup Y) \subseteq \mathsf{SWin}_1(G,F)$ and $\mathsf{SWin}_2(G^2_{X,Y}, F \cup Y) \supseteq \mathsf{SWin}_2(G,F)$.

(b) If $Y \subseteq \mathsf{SWin}_2(G,F) \setminus F$, then $\mathsf{SWin}_1(G^2_{X,Y}, F \cup Y) = \mathsf{SWin}_1(G,F)$ and $\mathsf{SWin}_2(G^2_{X,Y}, F \cup Y) = \mathsf{SWin}_2(G,F)$.

Proof. **(a).** Consider the statement $\mathsf{SWin}_2(G^2_{X,Y}, F \cup Y) \supseteq \mathsf{SWin}_2(G,F)$. Let $s \in \mathsf{SWin}_2(G,F)$. By Corollary 1, the rank of s in $G^2_{X,Y}$ is smaller than its rank in G. Since any state with a finite rank in $G^2_{X,Y}$ is a winning state for P2, $s \in \mathsf{SWin}_2(G^2_{X,Y}, F \cup Y)$. The statement $\mathsf{SWin}_1(G^2_{X,Y}, F \cup Y) \subseteq \mathsf{SWin}_1(G,F)$ follows from $\mathsf{SWin}_2(G^2_{X,Y}, F \cup Y) \supseteq \mathsf{SWin}_2(G,F)$ using Proposition 1.

(b) Consider the statement $\mathsf{SWin}_2(G^2_{X,Y}, F \cup Y) = \mathsf{SWin}_2(G,F)$.

(⊇). Given any state $s \in \mathsf{SWin}_2(G,F)$, by Corollary 1, its rank in $G^2_{X,Y}$ is finite. Thus, we have $\mathsf{SWin}_2(G^2_{X,Y}, F \cup Y) \supseteq \mathsf{SWin}_2(G,F)$.

(⊆). By way of contradiction, suppose there exists a state $s \in \mathsf{SWin}_2(G^2_{X,Y}, F \cup Y)$ such that $s \notin \mathsf{SWin}_2(G,F)$. This means that P2 has a greedy deterministic strategy, say π_2, to enforce a visit to $F \cup Y$ in $G^2_{X,Y}$. But following π_2 in G does not induce a visit to F. Now, if π_2 induces a visit to F in $G^2_{X,Y}$, then it must also be a sure winning strategy for P2 in G, as the presence of fake targets only affects the outgoing transitions from Y. Therefore, it must be the case that the following π_2

induces a visit to Y in $G^2_{X,Y}$. Since $Y \subseteq \mathsf{SWin}_2(G,F)$, in game G, P2 has a greedy deterministic strategy to enforce a visit to F from any state in Y. Thus, by following π_2 until visiting Y and then following any greedy sure winning strategy in game G to visit F from Y, P2 can enforce a visit to F from state s—a contradiction.

The statement $\mathsf{SWin}_1(G^2_{X,Y}, F \cup Y) = \mathsf{SWin}_1(G,F)$ follows from $\mathsf{SWin}_2(G^2_{X,Y}, F \cup Y) = \mathsf{SWin}_2(G,F)$ using Proposition 1. □

Intuitively, Proposition 4(a) states that when fake targets are placed within P1's sure winning region in G, P2 misperceives some states that are truly winning for P1 to be winning for itself. This is because P2 misperceives fake targets Y as goal states.

Proposition 4(b) is particularly noteworthy, as it reveals that placing fake targets within P2's sure/almost-sure winning region in game G has no impact on the sure/almost-sure winning regions of the players in P2's perceptual game. This observation is intuitively supported by Corollary 1, which states that the rank of a state in $\mathsf{SWin}_2(G,F) \setminus F$ cannot increase when a subset of states from this set are assigned as fake targets. Additionally, there cannot exist a state outside $\mathsf{SWin}_2(G,F)$ from which P2 can enforce a visit to $F \cup Y$ in $G^2_{X,Y}$. Because, if such a state existed, then it should have been included in $\mathsf{SWin}_2(G,F)$ since from all states in Y in game G, P2 has a strategy to enforce a visit to F.

However, the inclusion of fake targets in $\mathsf{SWin}_2(G,F)$ results in modifying the set of strategies that are subjectively rationalizable for P2 when players use greedy sure winning strategies. This is due to the alteration of state ranks, which influence the set of subjectively rationalizable actions under the sure winning condition available at each state. The following example illustrates this phenomenon.

Example 2. Fig. (3-2) shows the perceptual games of P1 and P2 when a fake target is placed at the state s_7. Fig. (3-2)(a) shows P1's perceptual game, in which s_7 is marked as a sink state (see honeypot symbol). The sure winning region of P1 in this game contains the states $\{s_7, s_8, s_9, s_{10}, s_{11}\}$ (shown in blue), and that of P2 contains $\{s_0, s_1, s_2, s_3, s_4, s_5, s_6\}$ (shown in red). Fig. (3-2)(b) shows P2's perceptual game, where P2 misperceives s_7 as a target. Consequently, the

sure winning region of P1 is $\{s_{10}, s_{11}\}$ (shown in blue) and that of P2 contains the states $\{s_0, \ldots, s_9\}$ (shown in red).

Observe how the fake target s_7 affects the ranks of the states $s_0, \ldots, s_9$. When players use greedy sure winning strategies, s_7's rank changes from 3 in the base game (see Fig. (3-1)) to 0 in P2's perceptual game. Similarly, the states s_5 and s_8, from which P2 has a strategy to visit s_7 in one step, attain rank 1 in P2's perceptual game.

The changes to the ranks of the states affect P2's subjectively rationalizable strategy in its perceptual game. For instance, consider the action b_2 at state s_5. In the base game, b_2 is subjectively rationalizable for P2 because it is rank-reducing. However, in P2's perceptual game, the action b_2 is not rank-reducing. Therefore, it is not subjectively rationalizable. In fact, the action b_1, which was not rationalizable in the base game, becomes subjectively rationalizable for P2 in its perceptual game.

3.3.3 Stealthy Deceptive Sure Winning Strategy

In this subsection, we introduce a new *hypergame on graph* model to synthesize a stealthy deceptive sure winning strategy for P1. Two key observations facilitate our definition of the hypergame on graph:

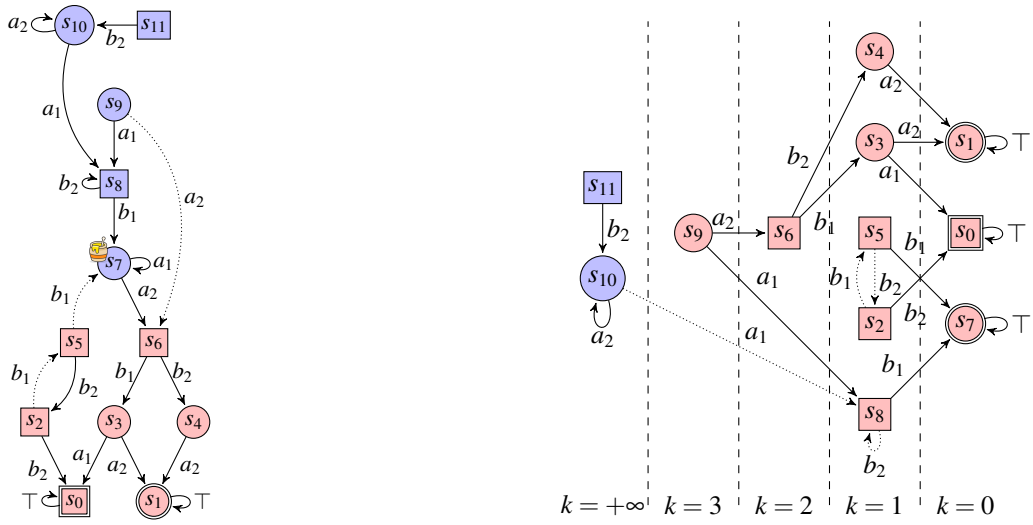
(i) When the game starts at a P1's sure/almost-sure winning state in $G^1_{X,Y}$, P1 can prevent the game from reaching F without the use of decoys.

(ii) When the game starts from a P2's sure/almost-sure winning state in $G^1_{X,Y}$, the only way for P1 to prevent the game from visiting F is by forcing a visit to a decoy state.

As a result, P1's safety objective to prevent a visit to F reduces to a reachability objective to visit $X \cup Y$.

Lemma 3-3. *Any P1 strategy π_1 at a state $s \in \mathsf{SWin}_2(G, F)$ that prevents a visit to F in the true game $G^1_{X,Y}$ must ensure a visit to a state in $X \cup Y$.*

Proof. We will focus on the case where players utilize randomized strategies, given that deterministic strategies are a special case of randomized strategies.



(a) P1's perceptual game.

(b) P2's perceptual game.

Figure 3-2. Perceptual games when the state s_7 is a fake target. In both sub-figures, the blue-colored states are winning for P1, and the red-colored states are winning for P2. Dotted transitions depict actions that are not subjectively rationalizable for P2 when players use greedy deterministic strategies.

By way of contradiction, suppose that there exists a strategy π_1 for P1 to prevent the game $G^1_{X,Y}$ from reaching F starting from state s while ensuring that no state in $X \cup Y$ is visited. In other words, the game remains indefinitely within the set $\mathsf{SWin}_2(G,F) \setminus (F \cup X \cup Y)$. However, by definition, for every state $s \in \mathsf{SWin}_2(G,F)$, P2 possesses a strategy π_2 that guarantees a visit to F from s in the original game G, regardless of P1's strategy. Therefore, if P1 follows π_1 and P2 follows π_2 in the game $G^1_{X,Y}$, the resulting path must indefinitely remain within the set $\mathsf{SWin}_2(G,F) \setminus (F \cup X \cup Y)$ while also visiting F—a contradiction. Consequently, the only way for P1 to prevent the game from reaching F is by visiting the set $X \cup Y$, which contains sink states. □

Following Observation (i) and Lma. 3-3, we define our hypergame on graph model as a reachability game, in which the players only follow strategies that are subjectively rationalizable for P2 and P1's objective is to reach a decoy state. When players use greedy sure winning

strategies, the set of subjectively rationalizable actions at a P2 state in $\mathsf{SWin}_2(G,F)\setminus F$ is given by

$$\mathsf{SRAct}(s) = \{a \in A_2 \mid s' = T(s,a) \wedge \mathsf{rank}_{G^2_{X,Y}}(s') < \mathsf{rank}_{G^2_{X,Y}}(s)\}. \tag{3-3}$$

Definition 16 (Hypergame on Graph)**.** Given the game G, the sets of decoys $X,Y \subseteq \mathsf{SWin}_2(G,F)$, and a function SRAct that maps every state in G to a set of subjectively rationalizable actions for P2, the hypergame on graph representing the L1-hypergame H_1 is the tuple,

$$\widehat{H}_1(X,Y) = \langle \mathsf{SWin}_2(G,F), A, \widehat{T}_{X,Y}, X \cup Y \rangle,$$

where

- $\mathsf{SWin}_2(G,F)$ is set of states.
- $\widehat{T}_{X,Y} : S \times A \to S$ is a *deterministic* transition function such that, for any state $s \in \mathsf{SWin}_2(G,F)$, $\widehat{T}_{X,Y}(s,a) = T(s,a)$ if and only if $a \in \mathsf{SRAct}(s)$. Otherwise, $\widehat{T}_X(s,a)$ is undefined.
- $X \cup Y \subseteq \mathsf{SWin}_2(G,F)\setminus F$ is the set of states representing P1's reachability objective.

It is noted that the set $X \cup Y$ in $\widehat{H}_1(X,Y)$ defines P1's reachability objective, not P2's objective.

Theorem 3-3. *Every sure winning strategy of P1 in $\widehat{H}(X,Y)$ is a stealthy deceptive sure winning strategy for P1 in the L2-hypergame, $H_2(X,Y)$.*

Proof. Every action available to P1 and P2 in $\widehat{H}(X,Y)$ is greedy and subjectively rationalizable for P2 by construction. Therefore, every sure winning strategy of P1 in $\widehat{H}(X,Y)$ is greedy and subjectively rationalizable for P2. By Lma. 3-3, the strategy is stealthy deceptive sure winning for P1 in $H_2(X,Y)$. □

Example 3. In Fig. (3-3), we present the hypergame on a graph that captures the interaction between P1 and P2 as described in Example (1). The hypergame includes states $s_0 \ldots s_9$,

representing P2's sure winning region in the base game. Dotted transitions represent P2's actions that are not subjectively rationalizable for P2 in its perceptual game and are excluded from the hypergame on graph. The result of applying Alg. 2-1 to the hypergame on graph is shown by coloring the states of the hypergame on graph. Cyan-colored states indicate that P1 has a sure winning strategy to reach s_7 from those states, representing P1's stealthy deceptive sure winning region. Red-colored states indicate P2's sure winning region, from which P1 has no deceptive strategy to prevent a visit to s_0 or s_1.

For example, P1's sure winning strategy at s_9 is to select action a_1, which leads to the P2 state s_8. From there, the only subjectively rationalizable action for P2 is b_1, which leads the game to visit the fake target. It is important to note that action a_2 at state s_9 is stealthy since it is subjectively rationalizable for P2 but not deceptive sure winning for P1, as it would lead to state s_6 from which P1 does not possess a strategy to prevent the game from reaching either s_0 or s_1.

Now, consider states s_2 and s_5. State s_5 is a stealthy deceptively sure winning state for P1 because the only greedy strategy available to P2 at s_5 selects action b_1, which leads to the fake target s_7. Note that a strategy that selects b_2 at s_5 is not greedy because s_5 and s_2 have ranks equal to 1. Similarly, state s_2 is not stealthy deceptively sure winning state for P1 because the only greedy strategy at s_2 is to select action b_2 that leads to a true final state s_0, which P1 aims to prevent.

3.3.4 Stealthy Deceptive Almost-Sure Winning Strategy

In this section, we examine Problem 2 under the almost-sure winning criterion when players employ randomized strategies. Unlike the result from Sec. 3.3.5, we find that there is no clear advantage of either fakes or traps over the other. This difference stems from the fact that, when using randomized strategies, the players are not required to use rank-reducing strategies. The set of actions subjectively rationalizable for P2 in this case is given by

$$\widehat{\mathsf{SRAct}}(s) = \{a \in A_2 \mid T(s,a) \in \mathsf{SWin}_2(G,F)\}, \tag{3-4}$$

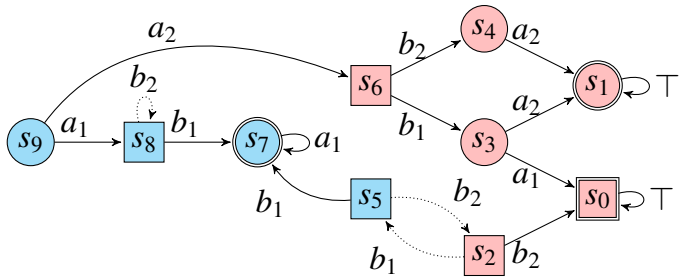


Figure 3-3. Hypergame on graph constructed based on P1 and P2's perceptual games shown in Fig. (3-2). Dotted lines depict P2's subjectively rationalizable actions. The cyan-colored states are stealthy deceptive sure winning states for P1, whereas the red-colored states are sure winning for P2.

for any state $s \in S_2 \cap \mathsf{SWin}_2(G,F) \setminus F$. By definition, all available actions are subjectively rationalizable for P2 at every other state.

Intuitively, starting from a P2's almost-sure winning state in $G^2_{X,Y}$, every P2 action that ensures that the game remains within the same region is subjectively rationalizable for P2. This is because (a) from every state in this region, P2 can enforce a visit to $F \cup Y$ with a positive probability, and (b) P1 has no strategy to exit this region. Therefore, a randomized strategy that selects every subjectively rationalizable action at a state with a positive probability is guaranteed to enforce a visit to $F \cup Y$ with probability one [68]. Such a randomized is an almost-sure winning strategy for P2 in game G [78, Chapter 10].

Lemma 3-4. *Let $s \in \mathsf{SWin}_2(G,F) \setminus (F \cup Y)$ be a state in P2's perceptual game $G^2_{X,Y}$ with the decoys $X, Y \subseteq \mathsf{SWin}_2(G,F) \setminus F$. Then, the set of subjectively rationalizable actions at s in game G is equal to that in game $G^2_{X,Y}$.*

Proof. The lemma follows from two observations. First, by Proposition 4, since $Y \subseteq \mathsf{SWin}_2(G,F)$, we have $\mathsf{SWin}_2(G,F) = \mathsf{SWin}_2(G^2_{X,Y}, F \cup Y)$. That is, P2's winning region in G and $G^2_{X,Y}$ are equal. Second, by Def. 15, the transitions from any state $s \notin Y$ are identical in the two games, G and $G^2_{X,Y}$. The statement follows by the definition of $\widehat{\mathsf{SRAct}}$. □

Based on Lma. 3-4 and the knowledge that P2 follows a randomized almost-sure winning strategy in $G^2_{X,Y}$, P1 can construct a MDP to represent the L1-hypergame H_1 by marginalizing the

true game $G^1_{X,Y}$ with P2's randomized almost-sure winning strategy. Since P1 does not know P2's choice of its strategy, P1 would assume, in the worst case, that P2's randomized strategy may choose any subjectively rationalizable action at a given state with positive probability. This results in the following hypergame MDP (adapted from [68]).

Definition 17 (Hypergame MDP)**.** Given the true game $G^1_{X,Y}$ and the function $\widehat{\mathsf{SRAct}}$ that maps every state $s \in \mathsf{SWin}_2(G,F)$ to the set of subjectively rationalizable actions for P2 at s, the hypergame MDP that represents L1-hypergame $H_1(X,Y)$ is the following tuple,

$$\widetilde{H}_1(X,Y) = \langle \widetilde{S}, A, \widetilde{T}_{X,Y}, X \cup Y \rangle,$$

where

- $\widetilde{S} = \mathsf{SWin}_2(G,F)$ is P2's sure winning region in G. At P1 states in $\widetilde{S}_1 = \mathsf{SWin}_2(G,F) \cap S_1$, P1 chooses the next action strategically. Whereas, the states in $\widetilde{S}_2 = \mathsf{SWin}_2(G,F) \cap S_2$ are *nature* states. At a nature state, the next state is chosen at random according to a predefined probability distribution.

- $\widehat{T}_{X,Y} : \widetilde{S} \times A \to \mathcal{D}(\widetilde{S})$ is a transition function defined as follows: any state $s \in X \cup Y$ is a sink state. At a state $s \in \widetilde{S}_1$, we have $\widehat{T}_{X,Y}(s,a,s') = 1$ if and only if $s' = T(s,a)$. At a state $s \in \widetilde{S}_2$, we have $\widehat{T}_{X,Y}(s,a,s') > 0$ if and only if $a \in \widehat{\mathsf{SRAct}}(s)$ and $s' = T(s,a)$. Otherwise, $\widehat{T}_{X,Y}(s,a,s') = 0$.

- $X \cup Y$ is the set of states representing P1's reachability objective.

It follows by construction that an almost-sure winning strategy of P1 in the hypergame MDP to visit $X \cup Y$ is a stealthy deceptive almost-sure winning strategy.

Theorem 3-4. *P1 can guarantee a visit to $X \cup Y$ from a state $s \in \mathsf{SWin}_2(G,F)$ in the true game $G^1_{X,Y}$ if and only if P1 has an almost-sure winning strategy to visit $X \cup Y$ from the state s in $\widetilde{H}_1(X,Y)$.*

With this, we can prove the key result of this section: *When players use randomized strategies and the games are analyzed under almost-sure winning condition, fake targets are equally valuable as traps*.

Theorem 3-5. *For any* $Z \subseteq \mathsf{SWin}_2(G,F) \setminus F$*, we have* $\mathsf{DASWin}_1(Z,\emptyset) = \mathsf{DASWin}_1(\emptyset,Z)$.

Proof. By Lma. 3-4, the hypergame MDPs $H_1(Z,\emptyset)$ and $H_1(\emptyset,Z)$ are identical. Therefore, P1's almost-sure winning regions in the two hypergames are equal. □

Since the fake targets and traps are equally valuable, Alg. 3-1 can be used to place the decoys in this setting by replacing $\mathsf{DSWin}_1(X,Y)$ with $\mathsf{DASWin}_1(X,Y)$ on line 6 and 12 of Alg. 3-1. However, in this case, the complexity of the algorithm is $\mathcal{O}((V+E)^2(M+N)^2)$ since the algorithm for computing the almost-sure winning region in the hypergame MDP has a time complexity of $\mathcal{O}((V+E)^2)$ [78].

We conclude this section by establishing that P1 may benefit more from deception when playing against P2 using an almost-sure winning strategy than when playing against P2 using a sure-winning strategy in P2's perceptual game.

Theorem 3-6. *For any* $X,Y \subseteq \mathsf{SWin}_2(G,F) \setminus F$*, we have* $\mathsf{DASWin}_1(X,Y) \subseteq \mathsf{DSWin}_1(X,Y)$.

Proof. We will establish that, for any state $s \in \mathsf{DASWin}_1(X,Y)$, it also belongs to $\mathsf{DSWin}_1(X,Y)$. To achieve this, we construct a stealthy deceptive sure-winning strategy π_1^d for P1, given any stealthy deceptive almost-sure winning strategy π_1^r.

Let π_1^d be a deterministic strategy such that $\pi_1^d(s) = a$, for some $a \in \mathsf{Supp}(\pi_1^r(s))$.

We will show that π_1^d is a stealthy deceptive sure winning strategy for P1. Recall that every stealthy deceptive sure winning strategy is a greedy, deterministic strategy subjectively rationalizable for P2 that ensures a visit to $X \cup Y$ in finitely many steps, regardless of the greedy, deterministic strategy followed by P2.

(π_1^d is subjectively rationalizable for P2). π_1^d is subjectively rationalizable for P2 whenever $\pi_1^d(s) \in \mathsf{SRAct}(s)$. This is indeed the case because the following three conditions hold

for all P1 state $s \in \mathsf{DASWin}_1(X,Y)$ by definition: (i) $\pi_1^d(s) \in \mathsf{Supp}(\pi_1^r(s))$, (ii) $\mathsf{Supp}(\pi_1^r(s)) \subseteq \widehat{\mathsf{SRAct}(s)}$, and (iii) $\widehat{\mathsf{SRAct}(s)} = \mathsf{SRAct}(s)$.

(π_1^d is greedy). The strategy π_1^d is greedy because every action enabled at a P1 state $s \in \mathsf{DASWin}_1(X,Y)$ is rank-reducing. This is because every state $s \in \mathsf{DASWin}_1(X,Y)$ is also a member of $\mathsf{SWin}_2(G,F)$ and Alg. 2-1 includes a P1 state s in $\mathsf{SWin}_2(G,F)$, if and only if all actions from s are rank-reducing.

(π_1^d induces a visit to $X \cup Y$). We establish that, given any greedy, deterministic P2 strategy π_2^d, every path $\rho \in \mathsf{Outcomes}_{\widehat{H_1(X,Y)}}(s, \pi_1^d, \pi_2^d)$ visits $X \cup Y$ within a finite number of steps. First, we note that $\mathsf{Outcomes}_{\widehat{H}_1(X,Y)}(s, \pi_1^d, \pi_2^d) \subseteq \mathsf{Outcomes}_{\widehat{H}_1(X,Y)}(s, \pi_1^r, \pi_2^r)$ holds for any randomized strategy π_2^r of P2. This is true because of two facts: (i) $\pi_1^d(s) \in \mathsf{Supp}(\pi_1^r(s))$, by definition, and (ii) $\pi_2^d(s) \in \mathsf{Supp}(\pi_2^r(s))$, which is true because $\widehat{\mathsf{SRAct}(s)} \subseteq \mathsf{SRAct}(s)$ holds for all P2 states. Second, we note that, since π_1^r is a stealthy deceptive almost-sure winning strategy, every path in $\mathsf{Outcomes}_{\widehat{H}_1(X,Y)}(s, \pi_1^r, \pi_2^r)$ eventually visits $X \cup Y$. Clearly, it cannot visit F because all states in F are sink states. Therefore, no path in $\mathsf{Outcomes}_{\widehat{H}_1(X,Y)}(s, \pi_1^d, \pi_2^d)$ visits F. Since both the strategies π_1^d and π_2^d are greedy, it follows by Lma. 3-3 that ρ must visit $X \cup Y$ within finitely many steps. □

Fig. (3-4) illustrates a toy example where the subset relation is strict, *i.e.*, $\mathsf{DASWin}_1(X,Y) \subsetneq \mathsf{DSWin}_1(X,Y)$. In this example, $F = \{s_0\}$ is a singleton final state that P2 aims to reach, $X = \{s_1\}$ is the set of traps, and $Y = \{s_2\}$ is the set of fake targets. This results in $\mathsf{DSWin}_1(\{s_1\},\{s_2\}) = \{s_1, s_2, s_4\}$ and $\mathsf{DASWin}_1(\{s_1\},\{s_2\}) = \{s_1, s_2\}$. Notice that s_4 is stealthy deceptively sure winning for P1, but not stealthy deceptively almost-sure winning. This is because, when players use greedy deterministic strategies, b is the only action at s_4 which is subjectively rationalizable for P2. Since $T(s_4, b) = s_2$ and s_2 is a fake target, the game is guaranteed to visit $X \cup Y$. However, when players used randomized strategies, both the actions b and c are subjectively rationalizable for P2 at s_4. Thus, the game may reach s_5 with a positive probability, from where P1 has no strategy to prevent the game from reaching F.

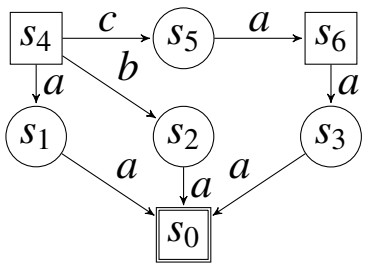


Figure 3-4. A scenario where $\mathsf{DASWin}_1(X,Y) \subsetneq \mathsf{DASWin}_1(X,Y)$.

3.3.5 Compositional Synthesis for Decoy Placement

Given a placement of traps and fake targets, Thm. 3-3 provides a way to compute P1's deceptive sure winning region given a fixed decoy allocation X,Y. Next, we formulate a combinatorial optimization problem in which P1 aims to maximize the size of its stealthy deceptive sure winning region by allocating traps and fake targets.

$$\begin{aligned} X^*,Y^* = &\underset{X,Y\subseteq \mathsf{SWin}_2(G,F)\setminus F}{\operatorname{argmax}} |\mathsf{DSWin}_1(X,Y)| \\ &\text{subject to: } |X|\leq M, |Y|\leq N, X\cap Y=\emptyset. \end{aligned} \tag{3-5}$$

In Eq. (3-5), every distinct choice of X,Y defines a hypergame, $\widehat{H}_1(X,Y)$, which must be solved to determine the size of $\mathsf{DSWin}_1(X,Y)$. A naïve approach to solving Eq. (3-5) is to compute $\mathsf{DSWin}_1(X,Y)$ for each valid placement of X,Y and then select a set $X\cup Y$ for which $|\mathsf{DSWin}_1(X,Y)|$ is the largest. However, this approach is not scalable because number of hypergames to solve is $\binom{|\mathsf{SWin}_2(G,F)\setminus F|}{M+N}\binom{M+N}{M}$, which grows rapidly with the size of game and number of decoys to place. To address this issue, we introduce a compositional approach to decoy placement in which we show that, when certain conditions hold, the decoy allocation problem can be formulated as a constrained supermodular maximization problem, for which a $(1-\frac{1}{e})$-approximation can be computed in polynomial time using a greedy algorithm [83].

The key insight behind our algorithm is that *fake targets could be more advantageous than traps*. This enables us to decouple the placement of traps and fake targets.

Theorem 3-7. *For any subset* $Z\subseteq \mathsf{SWin}_2(G,F)\setminus F$, *we have* $\mathsf{DSWin}_1(Z,\emptyset)\subseteq \mathsf{DSWin}_1(\emptyset,Z)$.

Proof. Recall that the stealthy deceptive sure winning region in the true game is determined by computing P1's sure winning region to reach the decoys in the hypergame. Therefore, the winning regions $\mathsf{DSWin}_1(Z,\emptyset)$ and $\mathsf{DSWin}_1(\emptyset,Z)$ have an attractor structure. Given any $X,Y \subseteq \mathsf{SWin}_2(G,F)\setminus F$, let $\mathsf{DSWin}_1^i(X,Y)$ denote the i-th level of attractor of the sure winning region $\mathsf{DSWin}_1(X,Y)$ in hypergame $\widehat{H}_1(X,Y)$.

We will prove by induction that, for any $n \geq 0$,

$$\mathsf{DSWin}_1^n(Z,\emptyset) \subseteq \mathsf{DSWin}_1^n(\emptyset,Z). \tag{3-6}$$

(Base Case). The statement is true for $n = 0$ because $\mathsf{DSWin}_1^0(Z,\emptyset) = \mathsf{DSWin}_1^0(\emptyset,Z) = Z$.

(Induction Step). Let $k \geq 0$ be an integer. Suppose that Eq. (3-6) holds for $n = k$. To show that every state $s \in \mathsf{DSWin}_1^{k+1}(Z,\emptyset)$ is an element of $\mathsf{DSWin}_1^{k+1}(\emptyset,Z)$, we consider two cases.

First, when s is a P1 state, P1 has an action in game $G^2_{Z,\emptyset}$ at state s to visit $\mathsf{DSWin}_1^k(Z,\emptyset)$ in one step. Since all P1 actions at a state in $\mathsf{SWin}_2(G,F)$ are subjectively rationalizable for P2, due to the induction hypothesis, using the same action at s would lead the game $G^2_{\emptyset,Z}$ to visit $\mathsf{DSWin}_1^k(\emptyset,Z)$ in one step. Hence, every P1 state in $\mathsf{DSWin}_1^{k+1}(Z,\emptyset)$ is an element in $\mathsf{DSWin}_1^{k+1}(\emptyset,Z)$.

Next, consider the case when s is a P2 state. Since $s \in \mathsf{DSWin}_1^{k+1}(Z,\emptyset)$, in game $G^2_{Z,\emptyset}$, P1 can ensure the game to visit Z in at most $(k+1)$-steps. Now, consider the state s in game $G^2_{\emptyset,Z}$. Since $G^2_{Z,\emptyset} = G$, the rank of s in G (and thus $G^2_{Z,\emptyset}$) must be smaller than or equal to $k+1$ in game $G^2_{\emptyset,Z}$ due to Corollary 1. That is, $s \in \mathsf{DSWin}_1^{k+1}(\emptyset,Z)$. □

Thm. 3-7 shows that any greedy algorithm to place traps and fake targets to solve Problem 2 must place fake targets before placing the traps.

In our previous work [4], we have studied Problem 2 when only traps are placed, *i.e.*, $Y = \emptyset$. Hence, we first investigate how to place the fake targets to maximize the deceptive sure-winning region for P1, given only fake targets. Then, we propose an algorithm to solve Problem 2 under sure winning condition by sequentially placing the fake targets and traps.

The concept of compositionality is important in developing a greedy algorithm for Problem 2. It enables us to incrementally place fake targets one by one, thereby constructing $\mathsf{DSWin}_1(\emptyset, Y)$ in an incremental manner. The following proposition states that $\mathsf{DSWin}_1(\emptyset, Y)$ supports compositionality.

Proposition 5. *Consider three placements of fake targets given by $Y_1 = \{s_1\}$, $Y_2 = \{s_2\}$, and $Y = Y_1 \cup Y_2$. Let $\mathsf{DSWin}_1(\emptyset, Y_1)$ and $\mathsf{DSWin}_1(\emptyset, Y_2)$ be P1's deceptive sure-winning regions in the hypergames $\widehat{H}_1(\emptyset, Y_1)$ and $\widehat{H}_1(\emptyset, Y_2)$, respectively. Then, P1's deceptive sure-winning region $\mathsf{DSWin}_1(\emptyset, Y)$ in the hypergame $\widehat{H}_1(\emptyset, Y)$ is equal to the sure-winning region for P1 in the following game:*

$$\widehat{H}_1(\emptyset, Y) = \langle \mathsf{SWin}_2(G^2_{\emptyset,Y}, F), A, \widehat{T}_{\emptyset,Y}, \\ \mathsf{DSWin}_1(\emptyset, Y_1) \cup \mathsf{DSWin}_1(\emptyset, Y_2) \rangle,$$

where P1's goal is to reach the target set $\mathsf{DSWin}_1(\emptyset, Y_1) \cup \mathsf{DSWin}_1(\emptyset, Y_2)$ and P2's goal is to prevent P1 from reaching the target set.

Proof. It is observed that the underlying graphs of the three deceptive reachability games, namely $\widehat{H}_1(\emptyset, Y_1)$, $\widehat{H}_1(\emptyset, Y_2)$, and $\widehat{H}_1(\emptyset, Y)$, are identical. They only differ in terms of the reachability objectives of P1. Applying Proposition 3, we have

$$\mathsf{DSWin}_1(\emptyset, Y) = \mathsf{DSWin}_1(\emptyset, \mathsf{DSWin}_1(\emptyset, Y_1) \cup \mathsf{DSWin}_1(\emptyset, Y_2)),$$

which concludes the proof. □

Corollary 2. *Given a set of fake targets, $Y \subseteq \mathsf{SWin}_2(G, F) \setminus F$ and a state $s \in \mathsf{SWin}_2(G, F) \setminus F$, we have*

$$\mathsf{DSWin}_1(\emptyset, Y) \cup \mathsf{DSWin}_1(\emptyset, \{s\}) \subseteq \mathsf{DSWin}_1(\emptyset, Y \cup \{s\})$$

Proof. Follows immediately by Proposition 3 and the property of the sure-winning region that the goal states of a reachability objective are a subset of the sure-winning region. □

Thus, if we consider the size of $\mathsf{DSWin}_1(\emptyset, Y)$ to be a measure of the effectiveness of allocating the states in $\mathsf{SWin}_2(G,F)$ as fake targets, then Corollary 2 states that the effectiveness of adding a new state to a set of decoys is greater than or equal to the sum of their individual effectiveness. In other words, DSWin_1 operator is *superadditive* [84, 85].

Let $\uplus$ represent the operation of composing two deceptive sure winning regions of P1. That is, given any subset $Y \subseteq \mathsf{SWin}_2(G,F) \setminus F$ and a state $s \in \mathsf{SWin}_2(G,F) \setminus F$,

$$\mathsf{DSWin}_1(\emptyset, Y \cup \{s\}) = \mathsf{DSWin}_1(\emptyset, Y) \uplus \mathsf{DSWin}_1(\emptyset, \{s\}).$$

With this notation, the problem of optimally placing the fake targets becomes equivalent to identifying a set $Y^* \subseteq \mathsf{SWin}_2(G,F) \setminus F$ such that,

$$Y^* = \underset{Y \subseteq \mathsf{SWin}_2(G,F) \setminus F}{\operatorname{argmax}} \left| \biguplus_{s \in Y} \mathsf{DSWin}_1(\emptyset, \{s\}) \right| \quad \text{subject to: } |Y| \leq N. \tag{3-7}$$

Let $g(Y) = \left| \biguplus_{s \in Y} \mathsf{DSWin}_1(\emptyset, \{s\}) \right|$ be a function that counts the number of P1's deceptive sure winning states when the set $Y \subseteq \mathsf{SWin}_2(G,F) \setminus F$ is allocated as fake targets.

Theorem 3-8. *The following statements are true.*

(a) g is a monotone, non-decreasing, and superadditive function.

(b) g is submodular if, for all $Y \subseteq S \setminus F$ and any $s \in S \setminus F$, we have
$\mathsf{DSWin}_1(\emptyset, Y) \cup \mathsf{DSWin}_1(\emptyset, \{s\}) = \mathsf{DSWin}_1(\emptyset, Y \cup \{s\})$.

(c) g is supermodular if, for all $Y \subseteq S \setminus F$ and any $s_1, s_2 \in S \setminus F$ and $s_1 \neq s_2$, we have
$\mathsf{DSWin}_1(X, Y \cup \{s_1\}) \cap \mathsf{DSWin}_1(X, Y \cup \{s_2\}) = \mathsf{DSWin}_1(X, Y)$

Proof. (**a**). Since for any set $Y \subseteq \mathsf{SWin}_2(G,F) \setminus F$ and any state $s \in \mathsf{SWin}_2(G,F) \setminus (F \cup Y)$, we have $\mathsf{DSWin}_1(\emptyset, Y) \cup \mathsf{DSWin}_1(\emptyset, \{s\}) \subseteq \mathsf{DSWin}_1(\emptyset, Y \cup \{s\})$, DSWin_1 is a non-decreasing, monotone function. Consequently, g is also a non-decreasing monotone. The function g is

superadditive because, by Corollary 2, $\mathsf{DSWin}_1(\emptyset, Y) \cup \mathsf{DSWin}_1(\emptyset, \{s\}) \subseteq \mathsf{DSWin}_1(\emptyset, Y \cup \{s\})$. Therefore, $g(Y) + g(\{s\}) \leq g(Y \cup \{s\})$.

(**b**). When $\mathsf{DSWin}_1(\emptyset, Y \cup \{s\}) = \mathsf{DSWin}_1(\emptyset, Y) \cup \mathsf{DSWin}_1(\emptyset, \{s\})$, we have $g(Y) = \left| \biguplus_{s \in D} \mathsf{DSWin}_{\{s\}} \right| = \left| \bigcup_{s \in D} \mathsf{DSWin}_{\{s\}} \right|$, which is submodular [86].

(**c**). The function g is supermodular if and only if

$$g(Y \cup \{s_1\}) + g(Y \cup \{s_2\}) - g(Y) \leq g(Y \cup \{s_1, s_2\}).$$

Given that $\mathsf{DSWin}_1(\emptyset, Y \cup \{s_1\}) \cap \mathsf{DSWin}_1(\emptyset, Y \cup \{s_2\}) = \mathsf{DSWin}_1(\emptyset, Y)$ holds for any holds for any $Y \subseteq \mathsf{SWin}_2(G, F)$ and any $s_1, s_2 \in \mathsf{SWin}_2(G, F)$, the LHS counts every state in $\mathsf{DSWin}_1(\emptyset, Y \cup \{s_1\}) \cup \mathsf{DSWin}_1(\emptyset, Y \cup \{s_2\})$ exactly once. On the other hand, RHS counts the number of states in $\mathsf{DSWin}_1(\emptyset, Y \cup \{s_1, s_2\})$. By Proposition 5, we know that RHS may contain states that are neither in $\mathsf{DSWin}_1(\emptyset, Y \cup \{s_1\})$ nor $\mathsf{DSWin}_1(\emptyset, Y \cup \{s_2\})$. □

Given the properties of $g(Y)$, we now consider the incremental placement of traps. The following proposition, which follows from Proposition 3, provides insight into the construction of the stealthy deceptive sure winning region when traps are placed given a fixed placement of fake targets.

Proposition 6. *Let* $\mathsf{DSWin}_1(\{s_1\}, Y)$ *and* $\mathsf{DSWin}_1(\{s_2\}, Y)$ *be P1's deceptive sure-winning regions in the hypergames* $\widehat{H}_1(\{s_1\}, Y)$ *and* $\widehat{H}_1(\{s_2\}, Y)$*, respectively. Then, P1's deceptive sure-winning region* $\mathsf{DSWin}_1(\{s_1, s_2\}, Y)$ *in the reachability game* $\widehat{H}_1(\{s_1, s_2\}, Y)$ *is equal to the sure-winning region for P1 in the following game:*

$$\widehat{H}_1(\{s_1, s_2\}, Y) = \langle \mathsf{SWin}_2(G^2_{X,Y}, F), A, \widehat{T}, \\ \mathsf{DSWin}_1(\{s_1\}, Y) \cup \mathsf{DSWin}_1(\{s_2\}, Y) \rangle,$$

where P1's goal is to reach the target set $\mathsf{DSWin}_1(\{s_1\}, Y) \cup \mathsf{DSWin}_1(\{s_2\}, Y)$ *and P2's goal is to prevent P1 from reaching the target set.*

Now, recall the following theorem regarding the exclusive placement of traps is known from [4].

Theorem 3-9. *For any $X \subseteq \mathsf{SWin}_2(G,F)$, let $f(X) \mapsto \mathbb{N}$ be a function that counts the size of $\mathsf{DSWin}_1(X,\emptyset)$. The following statements are true.*

(a) f is a monotone, non-decreasing, and superadditive function.

(b) f is submodular if, for all $X \subseteq S \setminus F$ and any $s \in S \setminus F$, we have
$\mathsf{DSWin}_1(X,\emptyset) \cup \mathsf{DSWin}_1(\{s\},\emptyset) = \mathsf{DSWin}_1(X \cup \{s\},\emptyset)$.

(c) f is supermodular if, for all $X \subseteq S \setminus F$ and any $s_1, s_2 \in S \setminus F$ and $s_1 \neq s_2$, we have
$\mathsf{DSWin}_1(X \cup \{s_1\},\emptyset) \cap \mathsf{DSWin}_1(X \cup \{s_2\},\emptyset) = \mathsf{DSWin}_1(X,\emptyset)$

Given Theorems 3-7, 3-8 and 3-9, the optimal placement of decoys reduces to that of sequentially solving two superadditive function maximization problems, first maximize $g(Y)$ and then maximize $f(Y)$. However, to the best of our knowledge, there are no approximation algorithms available for maximizing superadditive functions that are applicable to our setting. Therefore, we present Alg. 3-1 that returns an $(1-1/e)$-approximate solution to Problem 2 when either condition (b) or (c) in Theorems 3-8 and 3-9 are satisfied. This greedy algorithm is based on the GreedyMax algorithm for maximizing monotone submodular-supermodular functions in [83] and extends the algorithm discussed in [4, Algorithm 1].

Alg. 3-1 starts with empty sets of states X and Y. It first constructs the set Y by adding a new fake target in each iteration. In every step, a new fake target s is selected from the set of potential decoys D such that its inclusion, along with the previously chosen fake targets, maximizes the coverage of P1's deceptive sure-winning region over the states in $\mathsf{SWin}_2(G,F)$. The process continues until either a total of N fake targets have been selected, or the set of potential decoys is empty. Subsequently, the algorithm proceeds to construct X using a similar procedure, where the set of fake targets Y remains fixed, and a new trap is added to X in each iteration.

Complexity. Let V,E denote the number of states and transitions in the underlying graph of the hypergame $H_1(X,Y)$. Then, the time complexity of Alg. 3-1 is $\mathcal{O}((V+E)\cdot(M+N)^2)$. This is

Algorithm 3-1 Greedy algorithm for decoy placement.

Inputs: $\langle S, A, T, F \rangle$: Base game, M: Number of traps to placed, N: Number of fake targets to be placed.

Outputs: X, Y: Greedy placement of traps and fake targets.

$X \leftarrow \emptyset, Y \leftarrow \emptyset$
while $N - |Y| > 0$ **do**
  $D \leftarrow \{s \in \mathsf{SWin}_2(G,F) \mid s \notin (F \cup Y)\}$
  if D is empty **then**
    Exit While
  end if
  $d \leftarrow \arg\max_s |\mathsf{DSWin}_1(\emptyset, Y \cup \{s\})|$
  $Y \leftarrow Y \cup \{d\}$
end while
while $M - |X| > 0$ **do**
  $D \leftarrow \{s \in \mathsf{SWin}_2(G,F) \mid s \notin (F \cup X \cup Y)\}$
  if D is empty **then**
    Exit While
  end if
  $d \leftarrow \arg\max_s |\mathsf{DSWin}_1(X \cup \{s\}, Y)|$
  $X \leftarrow X \cup \{d\}$
end while
return X, Y

because the DSWin_1 computation, which uses Alg. 2-1, has a complexity of $\mathcal{O}(V+E)$ [21], and Alg. 3-1 must solve $|\mathsf{SWin}_2(G,F)| - |F| - j$ hypergames to determine the j-th decoy.

3.3.6 Experimental Evaluation

We use two experiments to illustrate the key results from our paper. The first experiment employs a gridworld example to demonstrate the proposed Alg. 3-1 and the effectiveness of the decoy placement. The second experiment highlights several key properties of the decoy placement determined by Alg. 3-1.

3.3.6.1 Tom and Jerry Gridworld

In this experiment, we consider a gridworld example featuring the characters Tom and Jerry as shown in Fig. (3-5). The 7×7 gridworld has 2 cheese blocks. Tom is equipped with M mouse traps and N fake cheese blocks to protect the real cheese from Jerry. Jerry's objective is to steal the cheese without getting caught by Tom (Tom captures Jerry when they are simultaneously in the same cell). On the other hand, Tom's objective is to place the decoys to safeguard the real

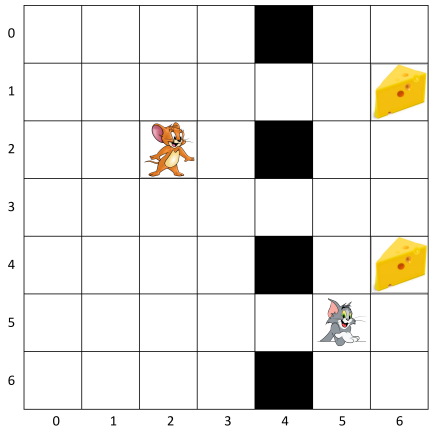


Figure 3-5. Gridworld example with Tom and Jerry with 2 cheese blocks.

cheese strategically. To achieve this, Tom intends to behave in a way that would either lead to Tom capturing Jerry or induce Jerry to visit a decoy. Jerry is assumed to be unaware of the presence of decoys. Both Tom and Jerry can occupy any cell in the gridworld that does not contain an obstacle (black cells). To avoid trivial cases, we assume that the game does not start with Jerry in a cell containing real cheese or a decoy.

A state in the base game between Tom and Jerry is represented as `(tom.row, tom.col, jerry.row, jerry.col, turn)` that captures the positions (a position is expressed in the row-column format) of Tom and Jerry and the player who selects the next action at that state. At any state, the player whose turn it is to play chooses an action from the set $\{N,E,S,W\}$ and moves to the cell in the intended direction. If the result of the action leads the player to a cell outside the bounds of gridworld or an obstacle, the player returns to the same cell where it started from.

We observe the effect of decoys on Tom's stealthy deceptive sure winning region in the gridworld configuration shown in Fig. (3-5) with two blocks of real cheese placed at cells $(1,6)$ and $(4,6)$. We consider three scenarios: (A) where $M = 2$ and $N = 0$, (B) where $M = 1$ and $N = 1$, and (C) where $M = 0$ and $N = 2$. This results in the base game's underlying graph having 4050 states and 16200 transitions. We use Alg. 3-1 for each scenario to determine the decoy placement under the sure winning criteria. The algorithm solves a total of 85 hypergames during the two iterations of the `While` loop (specifically, on lines `6` and `13`). The first iteration explores

43 candidate cells without obstacles or real cheese to determine the placement of the first decoy, while the second iteration explores 42. The algorithms are implemented in Python 3.10[1], and executed on a Windows 10 machine with a core i7 CPU running at 3.30GHz and equipped with 32GB of memory.

To measure and compare the effectiveness of a given placement of traps and fake targets during the iterations of Alg. 3-1, we introduce a real-valued metric called *value of deception*. Intuitively, the value of deception measures the proportion of P2's winning states in the base game G that become winning for P1 in the hypergame $\widehat{H}(X,Y)$ or $\widetilde{H}(X,Y)$. Under the stealthy deceptive sure winning condition, when $\mathsf{SWin}_2(G,F) \neq F$, the value of deception is defined as follows:

$$\mathsf{VoD}(X,Y) = \frac{|\mathsf{DSWin}_1(X,Y)|}{|\mathsf{SWin}_2(G,F)| - |F|}$$

If $\mathsf{SWin}_2(G,F) = F$, *i.e.*, when no states apart from the final states are winning for P2 in G, we set $\mathsf{VoD}(X,Y) = 0$. The value of deception is defined analogously when the interaction is analyzed under an almost-sure winning criterion.

We analyze the key insights obtained by solving 85 hypergames and examining the resulting value of deception. Fig. (3-7) depicts a heatmap, where the value displayed in each cell denotes the value of deception achieved by allocating the next decoy in that cell. The value in each cell is computed based on the map Z constructed during each of the two iterations of Alg. 3-1. The figure includes two heatmaps each for the three scenarios (A), (B), and (C). Specifically, Figures 3-7a and 3-7b depict the heatmaps corresponding to the first and second iteration of the algorithm for scenario (A). Similarly, Figures 3-7c and 3-7d show the two heatmaps for scenario (B), and Figures 3-7e and 3-7f for scenario (C).

In Fig. (3-7a), the cell values indicate the value of deception achieved by placing the first trap at each respective cell. For instance, the value 0.28 in cell $(1,5)$ indicates the value of deception obtained by placing the first trap at that location. The first trap is positioned at $(1,5)$ as it is the highest value. In Fig. (3-7b), the cell values indicate the combined value of deception

1 The source code is available at https://github.com/abhibp1993/decoy-allocation-problem.

achieved by placing the second trap at a given cell in addition to the trap selected in the first iteration. For instance, the value 0.5 in cell $(5,5)$ represents the value of deception obtained by placing two traps: the first trap at location $(1,5)$ (as determined in the first iteration) and the second trap at $(5,5)$. The second trap is placed there since the maximum deception value is observed at $(5,5)$. The heatmaps in Figures 3-7c-3-7f are understood in a similar manner.

We now discuss key observations and insights from Fig. (3-7). First, observe that when only traps are placed (Figures 3-7a, 3-7b, and 3-7d), the value of deception increases as we move closer to the real cheese. This is because traps cut Jerry's winning paths to real cheese. For instance, in Fig. (3-7a), suppose that Jerry starts from a cell in row 1 and Tom starts from a cell $(4,1)$. Then, Jerry has a sure winning strategy to steal the cheese at $(1,6)$. Now, consider two placements of the first trap: $(1,1)$ and $(1,5)$. The trap at $(1,1)$ will be effective only if Jerry starts at $(1,0)$ since if Jerry begins from a cell to the right of $(1,1)$, she is guaranteed to visit $(1,6)$ without being trapped or caught. On the other hand, the trap at $(1,5)$ will be effective whenever Jerry starts between $(1,0)$ and $(1,4)$ because every path induced by any of her sure winning strategies to visit $(1,6)$ from these initial positions passes through $(1,5)$. Hence, placing a trap at $(1,5)$ yields a higher value of deception than placing it at $(1,1)$.

In contrast, fake cheese attracts Jerry by providing an alternative to visiting the real cheese. Therefore, when placing the fake cheese, the value of deception increases as we move closer to the fake cheese. For instance, in Fig. (3-7c), we notice that the values in cells $(2,3)$ and $(3,3)$ are higher than their neighboring cells. This is because when fake cheese is present at either of these cells, Jerry believes there are three cheese blocks in the game instead of two. Consequently, when Tom starts at $(5,1)$ and Jerry starts at any cell with row coordinates of $0,1,2$ and column coordinates of $0,1,2$, Jerry's subjectively rationalizable sure winning strategy would lead him to visit either the fake cheese at $(2,3)$ or $(3,3)$ instead of the real cheese at $(1,6)$ or $(4,6)$. Since highest value of deception is observed at cell $(3,3)$, Tom places the first fake cheese at that cell.

The results also confirm our conclusion that *fake targets have a higher value than traps when the game is analyzed under sure winning condition*. To see this, compare the value of

deception for any cell in Fig. (3-7d) and Fig. (3-7f), and Fig. (3-7d) and Fig. (3-7f). We observe that the value in the second heatmap (where a fake cheese is placed in the cell) is greater than or equal to that in the first heatmap (where a trap is placed in the cell).

3.3.6.2 Decoy Placement over Randomly Generated Game on Graphs

In this second experiment, we compare the effectiveness of placing traps versus fake targets under stealthy, deceptive sure and almost-sure winning conditions. We employ randomly generated graphs to explore interesting case studies. Each game consists of 150 states, of which 75 are P1 states, and the remaining are P2 states. At every state in each game, we randomly select an integer between 1 and 5 to determine the number of actions enabled at that state. Subsequently, the next state on performing each enabled action at a given state is determined at random.

With these exploratory experiments, we focus our analysis on four games on graphs as these present interesting results. For each of the four games, we use Alg. 3-1 to determine decoy placement and compute the corresponding value of deception under four conditions: (i) placing 5 traps under stealthy deceptive sure winning condition, (ii) placing 5 fake targets under stealthy deceptive sure winning condition, (iii) placing 5 traps under stealthy deceptive almost-sure winning condition, and (iv) placing 5 fake targets under stealthy deceptive almost-sure winning condition. Fig. (3-6) depicts the variation in the value of deception for cases (i)-(iv) as we progressively introduce the traps or fake targets in four selected games.

Figures 3-6a and 3-6b present instances that align with our theoretical findings. Since the dashed blue line remains at par or below the solid blue line, we observe that the value of deception obtained by placing fake targets is greater than or equal to that obtained by placing traps, both under stealthy deceptive sure winning condition. This confirms the findings in Thm. 3-7. Furthermore, the overlapping of the red dotted line and green lines indicates that placing traps and fake targets under stealthy deceptive almost-sure winning condition yield the same value of deception, which is aligned with the findings of Thm. 3-5. Lastly, the outcomes also align with the implications outlined in Thm. 3-6, as both the red-dotted and green lines consistently remain positioned below the blue lines. Consequently, the value of deception achieved under the sure

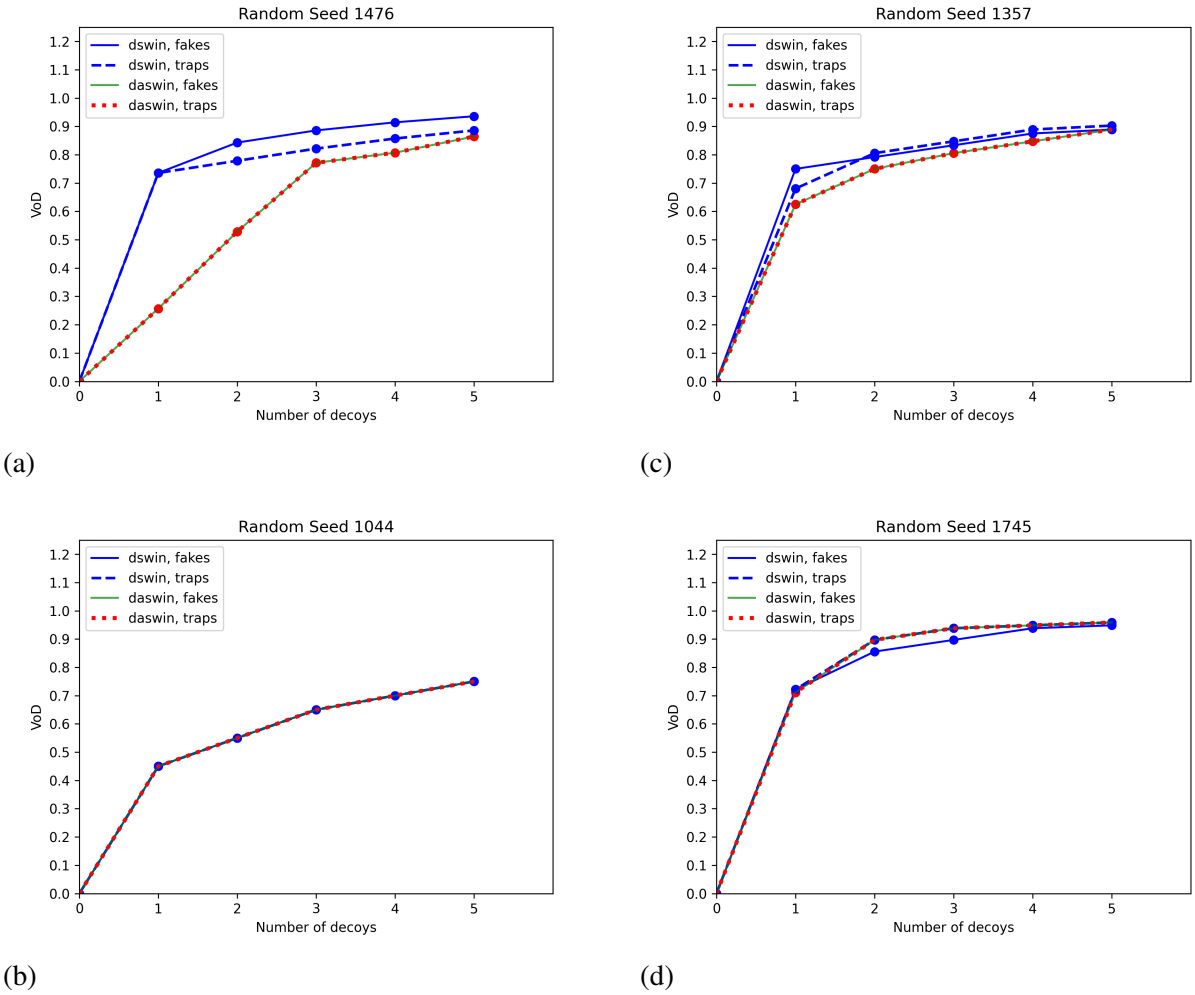


Figure 3-6. The value of deception obtained by placing traps and fake targets under stealthy deceptive sure and almost-sure winning conditions in four selected games.

winning condition is consistently greater than or equal to that attained under the almost-sure winning condition. Fig. (3-6b) presents a special case wherein the intrinsic topology of the game graph leads to a convergence of deception values across all four cases (i)-(iv).

Figures 3-6c and 3-6d present instances where the results appear to diverge from our theoretical predictions. In Fig. (3-6c), we encounter a situation where the value of deception achieved under the sure winning condition by strategically placing traps is greater than the value obtained by placing fake targets. This outcome seemingly contradicts the assertions made in Thm. 3-7. In Fig. (3-6d), we encounter another scenario where the value of deception obtained by

deploying either traps or fake targets under the almost-sure winning condition exceeds the value attained by placing fake targets under the sure winning condition, thereby deviating from the anticipated results stipulated in Thm. 3-6. However, these disparities can be attributed to the greedy approach employed by Alg. 3-1. For instance, in Fig. (3-6c), Alg. 3-1 determined the states `s22`, `s80` as the first two fake targets and `s101`, `s74` as the first two traps. To understand these choices, let us examine the values of deception for the following placements:

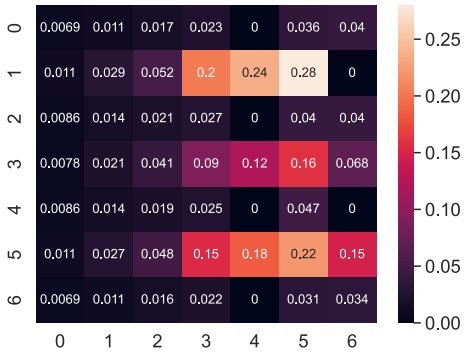
$$\mathsf{VoD}(\emptyset, \{\texttt{s22}\}) = 0.7500, \qquad \mathsf{VoD}(\emptyset, \{\texttt{s101}\}) = 0.6805$$

$$\mathsf{VoD}(\{\texttt{s22}\}, \emptyset) = 0.4166, \qquad \mathsf{VoD}(\{\texttt{s101}\}, \emptyset) = 0.6805$$

$$\mathsf{VoD}(\emptyset, \{\texttt{s101}, \texttt{s74}\}) = 0.8055, \qquad \mathsf{VoD}(\emptyset, \{\texttt{s22}, \texttt{s80}\}) = 0.7916$$

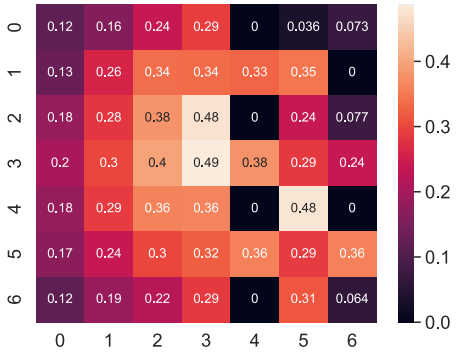
We observe that the value of deception attained by placing fake targets at `s101`, `s74` is higher than that obtained by placing them at `s22`, `s80`. Thus, we would expect the algorithm to select the latter states to be the fake targets. However, the Alg. 3-1 follows a greedy approach. Since the value of deception when the first fake target is placed at `s22` is greater than when it is placed at all other states, including `s101`, `s22` is selected as the first fake target. Given the first fake target, the choice of the second fake target that yields that maximum value of deception is `s80`. In other words, the deviation from theoretical expectations is due to the sub-optimal placement suggested by the greedy algorithm.

We conclude by noting that the value of deception increases monotonically until the value of 1.0 is attained. In any game, the value of 1.0 is guaranteed to be achieved if there is no bound on the number of decoys. In the worst case (for example, consider star topology), a decoy must be placed at every state for the value of deception to be one.

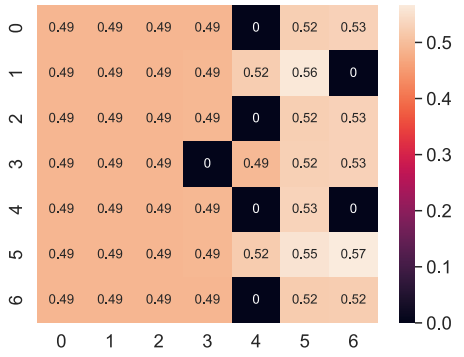


(a) Scenario (A): Value of deception when the first trap is placed within the given cell.

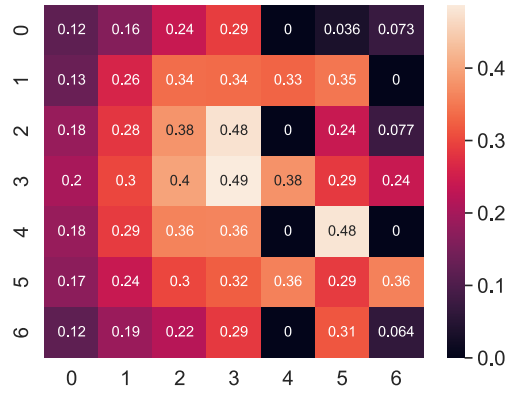
(b) Scenario (A): Value of deception when first trap is placed at $(1,5)$ and second trap is placed within the given cell.



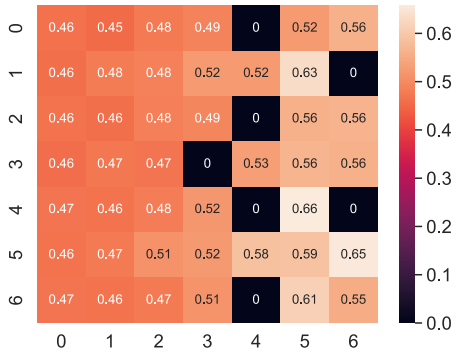
(c) Scenario (B): Value of deception when first fake cheese is placed within the given cell.



(d) Scenario (B): Value of deception when first fake cheese is placed at $(4,5)$ and a trap is placed within the given cell.



(e) Scenario (C): Value of deception when first fake cheese is placed within the given cell.



(f) Scenario (C): Value of deception when first fake cheese is placed at $(4,5)$ and the second fake cheese is placed within the given cell.

Figure 3-7. The values of deception compared by Alg. 3-1 in each of the two iterations to determine the two decoys for scenarios (A)-(C).

CHAPTER 4
SYNTHESIS WITH MISPERCEPTION OF ACTION CAPABILITIES

This chapter investigates the synthesis of deceptive winning strategies for the sub-class of games with incomplete information where P2 misperceives P1's action capabilities. These scenarios often arise in various domains, such as football, where the opposing team may be uncertain about a player's newly acquired skills before a match, or in economic situations where a firm may be unaware of another firm developing a similar product.

In a game, when a player realizes deceptive tactics are in play, their subsequent behavior can be uncertain [87]. There are two potential outcomes in this scenario. The player may opt to withdraw from the game; for instance, when an attacker learns that the defender has hidden action capabilities, it may choose to discontinue the attack. Alternately, the player may choose or may be forced to continue their engagement by adapting their knowledge and, consequently, their strategy; for instance, in football, the game must continue even after the new capabilities of the opponent team are revealed. This chapter focuses primarily on investigating the behavior of players in the latter case.

4.1 Effect of Action Misperception

Consider a reachability game between P1 and P2 characterized by a deterministic two-player turn-based zero-sum game, $G = \langle S, Act, T, s_0, F \rangle$, as defined in Def. 2. In this game, P1's objective is to visit a final state in F. P2's objective is to prevent the game from reaching a final state.

In this chapter, we study the game in which P2 does not know the complete action set of P1 at the beginning. Hence, the information structure of the game is captured by the following assumption.

Assumption 4 (Information Structure)**.** P1 knows its complete action set Act_1. P2 misperceives P1's action set to be a subset $X \subsetneq Act_1$. The components S and F of the game arena G are common knowledge for both the players.

Perceptual games. As a result of Assumption 4, the interaction between P1 and P2 is a game with incomplete information about action capabilities. Hence, P1 and P2 play different games in their minds to synthesize their respective winning strategies. P1's perceptual game is identical to the true game; $\langle S, Act_1 \cup Act_2, T, s_0, F \rangle$. Whereas, P2's perceptual game is a game under misperception; $\langle S, X \cup Act_2, T, s_0, F \rangle$. Let us formalize the new notation used to distinguish between the perceptual games of P1 and P2.

Notation 2. Given a subset of P1's action set, $X \subseteq Act_1$, let $G(X) = \langle S, X \cup Act_2, T, s_0, F \rangle$ denote the deterministic two-player turn-based game on a graph in which P1's action set X.

Therefore, P1's perceptual game is $G(Act_1)$ and P2's perceptual game is $G(X)$. Assuming P1 and P2 to be rational players, they would use the solution approach reviewed in Sec. 2 to compute their winning strategies in their respective perceptual games. However, P1 is likely to compute a conservative strategy because P1 over-estimates the information available to P2. Naturally, we want to know *whether P1 can improve its strategy if P1 is made aware of P2's current misperception X?*

Before we answer the above question, recall that we allow P2's misperception to evolve during the game. For instance, what would happen when P2 observes P1 playing an action $a \in Act_1$, which P2 did not believe to be in P1's action set? We might argue that P2 will at least add a new action a_1 to its perceived action set, X, of P1. Thus, the new perception would be $X \cup \{a_1\}$. Also, P2 might be capable of complex inference. That is, on observing that P1 can perform an action a_1, P2 might infer that P1 must be capable of actions a_2 and a_3, thus, updating its perception set to $X \cup \{a_1, a_2, a_3\}$. To capture such inference capabilities, we introduce a generic perception update function for P2.

Definition 18 (Inference Mechanism)**.** A deterministic inference mechanism is a function $\kappa : 2^{Act_1} \times Act_1 \to 2^{Act_1}$ that maps a subset of actions $X \subseteq Act_1$ and an action $a \in Act_1$ to an updated subset of actions $Y = \kappa(X, a)$ such that $a \in Y$.

If P2's misperception evolves during the game, then P1 must strategize when to reveal an

action that is not currently known to P2. By doing so, P1 may partially control the evolution of P2's perception to its advantage. Such a strategy, where P1 intentionally controls P2's misperception, is a *deceptive strategy*, by definition. We formalize our problem statement.

Problem 3. Consider a reachability game G in which Assumption 4 holds. If P1 is informed of the initial misperception of P2, X_0, and its inference mechanism η, then synthesize a deceptive strategy using which P1 can satisfy its reachability objective under sure and almost-sure winning conditions.

In particular, we want to investigate whether the use of deception is advantageous for P1 or not. We say P1 gets advantage with deception if at least one game state that is not sure/almost-sure winning for P1 in the game without deception becomes winning for P1 with the use of deception.

4.2 Dynamic Hypergame on Graph

When two players play different games in their minds, their interaction can be modeled as a hypergame [27]. While P1 and P2 play different games in their minds as per Problem 3, their interaction is distinguished by the ability of P2 to update its game as P2 learns about P1 actions that were previously unknown to him. The hypergame model described in Sec. 2.3 is insufficient to model this situation. Hence, we propose a new model called dynamic hypergame that makes the evolution of P2's game explicit.

The first-level dynamic hypergame is the tuple of the perceptual games being played by the players,

$$H^1(X) = \langle G(Act_1), G(X) \rangle,$$

where, given the current perception of P2, $X \subseteq Act_1$, P1 and P2 respectively solve the games $G(Act_1)$ and $G(X)$ to compute their winning strategies. Notice the dependence of the hypergame $H^1(X)$ on P2's perception X captures the fact that $H^1(X)$ is indeed a dynamic hypergame.

Given that P1 is aware of the P2's perception, the interaction is modeled as a second-level hypergame. Specifically, we assume P1 knows X. Therefore, the second-level hypergame is

$H^2 = \langle H^1(X), G(X)\rangle$. Similar to Ch. 3, we introduce a graphical model called the *hypergame on a graph* to represent the dynamic hypergame $H^2(X)$.

Definition 19 (Dynamic Hypergame Transition System)**.** Let $\Gamma = \wp(Act_1)$ be the powerset of P1's action set. The dynamic hypergame on graph representing the second-level dynamic hypergame $H^2(X)$ is the tuple,

$$\mathcal{H} = \langle V, Act, \Delta, v_0, \mathcal{F}\rangle,$$

where

- $V = S \times \Gamma$ is the set of hypergame states,
- $Act = Act_1 \cup Act_2$ is the set of actions of P1 and P2,
- $\Delta : V \times Act \to V$ is the transition function such that $(s', X') = \Delta((s, X), a)$ if and only if $s' = T(s, a)$ and $X' = \kappa(X, a)$,
- $v_0 \in V$ is an initial state,
- $\mathcal{F} = F \times \Gamma$ is the set of final states.

Intuitively, the hypergame on a graph can be viewed as unrolling the game with different information states of P2.

Example 4 (Running Example)**.** Consider the game graph as shown in Fig. (4-1). The circle states $\{s_1, s_3\}$ are P1 states and the square states $\{s_0, s_2\}$ are P2 states. The objective of P1 is to reach to the final states set $F = \{s_0\}$ from the initial state s_2. P1's action set is $Act_1 = \{a_1, a_2\}$ and P2's action set is $Act_2 = \{b_1, b_2\}$.

The sure (or almost-sure) winning region of P1 in the game is $\mathsf{SWin}_1 = \{s_0, s_1\}$, shown in Fig. (4-1) as blue states. This is intuitively understood as follows. P1 can win from state s_1 by choosing the action a_1. However, the states $\mathsf{SWin}_2 = \{s_2, s_3\}$, shown in Fig. (4-1) as red states, are losing for P1 because P2 has a strategy to indefinitely restrict the game within SWin_2 by always selecting action b_2 at state s_2.

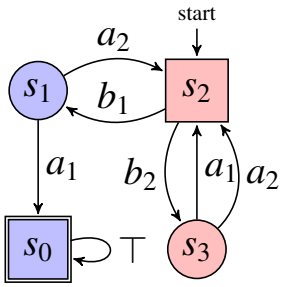


Figure 4-1. An example game on graph. The state space is divided into two parts: blue states $\mathsf{SWin}_1 = \{s_0, s_1\}$ are sure (almost-sure) winning for P1, and red states $\mathsf{SWin}_2 = \{s_2, s_3\}$ are sure (almost-sure) winning for P2.

Suppose that the action a_1 of P1 is initially not known to P2. Thus, at the beginning of the interaction, P2's perception of P1's action set is $X_0 = \{a_2\}$ and its perceptual game is the game $G(X_0)$ as shown in Fig. (4-2). Notice that Fig. (4-2) does not include edges corresponding to action a_1. On the other hand, P1's perceptual game is same as the *true* game $G(Act_1)$ shown in Fig. (4-1). Given that the final states set $\{s_0\}$ is not reachable in $G(X_0)$, P2 misperceives both of its actions, b_1 and b_2, to be safe to play at state s_2. However, in reality, only the action b_2 is safe in the *true* game, G_1.

Moreover, when P1 is aware of P2's misperception X_0, P1 may compute a deceptive strategy which would not use a_1 unless the game state is s_1. Because, if P1 uses a_1 at s_3 then P2 will update its perception to $X_1 = Act_1$ and conclude that action b_1 is unsafe to play at state s_2. In this case, P1 will not be able to win the game starting at s_2 or s_3.

The hypergame corresponding to above interaction is shown in Fig. (4-3). The figure only shows the reachable states. Every state in the hypergame is represented as a tuple of a game state and the current perception of P2 at that state. Given $X_0 = \{a_2\}$, two perceptual games of P2: $G(\{a_2\})$ and $G(\{a_1, a_2\})$, are possible. Any hypergame-play that visits the final state (s_0, X_1) is winning for P1. Therefore, the hypergame-plays $\tau_1 = (s_2, X_0)b_1(s_1, X_0)a_1(s_0, X_1)$ and $\tau_2 = (s_2, X_0)b_2(s_3, X_0)a_1(s_2, X_1)b_1(s_1, X_1)a_1(s_0, X_1)$ are the examples of winning plays for P1. Interestingly, in the next section, we will show that the play τ_2 may never occur if both players act rationally. However, it is possible for the play τ_1 to be observed.

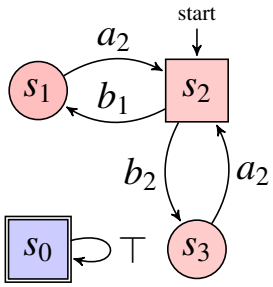


Figure 4-2. Perceptual game of P2 when P2 misperceives P1's action set to be $X_0 = \{a_2\}$. The state space is divided into two parts: the blue state $\{s_0\}$ is perceived by P2 as the only winning state of P1, and the red states $\{s_1, s_2, s_3\}$ are perceived by him to be winning for himself. Due to misperception, this partition is different from the partition in Fig. (4-1).

4.2.1 P2's Subjectively Rationalizable Strategy

To design an algorithm to synthesize a deceptive strategy in the hypergame $\mathcal{H}$, we must reason about P2's perception and its SR strategy. Because P2 plays a safety game, its strategy in a game on graph is a permissive strategy. Recall that an action is permissive for a player at a given state if the player can stay within the winning region by performing that action [88]. However, in a game with incomplete information, whether a state is perceived to be winning or not depends on the player's perception. The following definition characterizes the actions that P2 considers to be rational given its perceptual game. As the perceptual game of P2 evolves during the interactions, so does the set of its subjectively rationalizable actions.

Definition 20 (P2's Subjectively Rationalizable Actions)**.** Let $u = (s, X) \in V_2$ and $v = (s', X) \in V_2$ be two hypergame states such that $v = \Delta(u, b)$ for some $b \in Act_2$. Then, the set of P2's subjectively rationalizable actions at u is the set

$$\mathsf{SRAct}_2(u) = \{a \in Act_2 \mid s' \in \mathsf{SWin}_2(X)\}.$$

In words, the set of P2's subjectively rationalizable actions at a given state $u = (s, X)$ is the set of permissive actions for P2 in the perceptual game with action set X.

We make two important observations about P2's subjectively rationalizable actions. First, P2's action has no effect on its perception. Therefore, if P2's perception was X at a state $u \in V_2$

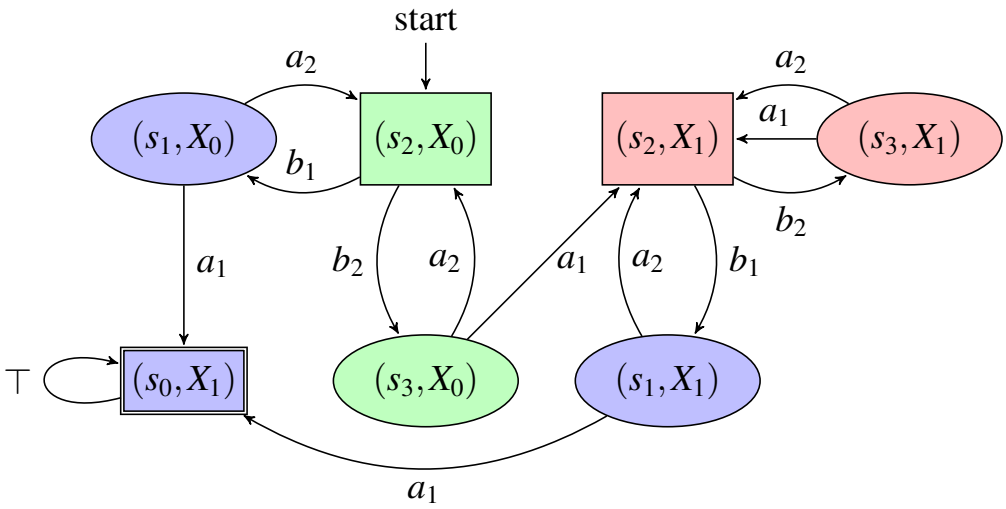


Figure 4-3. The dynamic hypergame on graph. The state space is divided into three parts: blue states $\{(s_0,X_1),(s_1,X_0),(s_1,X_1)\}$ are sure (almost-sure) winning for P1, and red states $\{(s_2,X_1),(s_3,X_1)\}$ are sure (almost-sure) winning for P2 regardless of whether P1 uses deception or not. The green states $\{(s_2,X_0),(s_3,X_0)\}$ are almost-sure winning, but not sure winning, for P1 when P1 uses deception.

then, for any $b \in Act_2$, P2's perception at a state $v = \Delta(u,b)$ is also X. This observation follows from Def. 18.

The second observation states that if an action of P2 is permissive at a some state in which P2 knows the complete action set of P1 then it is subjectively rationalizable under any perception. This is because a sure winning action of P2 remains a sure winning action regardless of P2's misperception.

Proposition 7. *If a P2 action $b \in Act_2$ is subjectively rationalizable at the state (s,Act_1) then it is subjectively rationalizable at any state $(s,X) \in V_2$ for any $X \subseteq Act_1$.*

It is noted that the converse of Proposition 7 may not hold. That is, under misperception, P2 might misperceive its non-permissive action to be permissive. Consequently, if P1 could trick P2 into selecting such a non-permissive action, P1 may force the game from a P1's losing state to a P1's winning state in the true game, $G(Act_1)$. In the next two sections, we investigate when P1 has a strategy to enforce P2 into choosing a non-permissive action under sure and almost-sure winning conditions.

4.2.2 Deceptive Sure Winning Strategy

Given the notion of P2's subjectively rationalizable strategy, we formally define a deceptive sure winning strategy of P1. In contrast to Ch. 3.1, we do not require the deceptive strategy to be stealthy since we want P1 to influence P2's perception.

Definition 21 (Deceptive Sure Winning Strategy)**.** A memoryless, deterministic strategy $\pi_1 : V \to Act_1$ is said to be a *deceptively sure winning* for P1 at a state $v \in V$ if and only if, for any memoryless, deterministic subjectively rationalizable strategy $\mu : V_2 \to Act_2$ of P2 and for any run $\rho \in \mathsf{Outcomes}(v, \pi_1, \mu)$, we have $\mathsf{Occ}(\rho) \cap \mathcal{F} \neq \emptyset$.

In Def. 21, P1 reasons only about all possible subjectively rationalizable strategies of P2, which is in contrast to Def. 3 where P1 reasons about all possible strategies of P2. A hypergame state $v \in V$ from which P1 has a deceptively sure winning strategy is called as a *deceptively sure winning state*. The exhaustive set of deceptively sure winning states is called the *deceptively sure winning region*, denoted by DSWin_1. Note that deceptive sure winning region cannot be defined for P2 because P2 does not know the hypergame, H.

The following theorem proves a negative result that deceptive sure winning strategy provides P1 with no advantage over a non-deceptive sure winning strategy.

Theorem 4-1. *Let* $\mathsf{DSWin}_1 \lfloor_S = \{s \in S \mid v \in \mathsf{DSWin}_1 \text{ and } s = v \lfloor_S\}$ *be the set of projection of the deceptively sure winning states onto the game state space. It holds that* $\mathsf{SWin}_1(Act_1) = \mathsf{DSWin}_1 \lfloor_S$.

To prove Thm. 4-1, we need the following lemma which states that every non-deceptively sure winning state is also deceptively sure winning.

Lemma 4-1. *If a game state $s \in S$ is a non-deceptive sure winning state for P1 then, for any $X \in \Gamma$, the hypergame state $v = (s, X)$ is a deceptively sure winning state for P1.*

Now, we prove Thm. 4-1.

Proof (Thm. 4-1). ($\subseteq$) By Proposition 7, at given any state $v = (s, X) \in V_2$ such that $s \in \mathsf{SWin}_2(Act_1)$, every permissive action of P2 at s is also subjectively rationalizable at v for any $X \subseteq Act$. Therefore, P2's subjectively rationalizable strategy μ at v may select a truly permissive action. By definition, v cannot be sure winning for P1.

($\supseteq$) Follows from Lma. 4-1. □

4.2.3 Deceptive Almost-Sure Winning Strategy

The fundamental reason behind why deception does not yield advantage under sure winning condition is that the players use deterministic strategies. If there exists a truly permissive action at a P2 state, there is a possibility that P2's subjectively rationalizable strategy chooses that action every time that state is visited. In this section, we study P1's deceptive strategy under almost-sure winning condition in which players use randomized strategies. In contrast to sure winning condition, we show that P1 may gain advantage under almost-sure winning condition.

Assumption 5. At a state $v \in V_2$, P2 selects every subjectively rationalizable action $b \in \mathsf{SRAct}_2(v)$ with a positive probability. That is, $\mathsf{Supp}(\mu(v)) = \mathsf{SRAct}_2(v)$.

Now, we formalize the notion of deceptive almost-sure winning strategy.

Definition 22 (Deceptive Almost-Sure Winning Strategy)**.** Given a hypergame state $v \in V$, a memoryless, randomized strategy π is said to be *deceptive almost-sure winning strategy* for P1 if and only if for every memoryless, randomized subjectively rationalizable strategy μ of P2 satisfying Assumption 5, the probability that a run $\rho \in \mathsf{Outcomes}(v, \pi, \mu)$ in the hypergame $\mathcal{H}$ satisfies the condition $\mathsf{Occ}(\rho) \cap \mathcal{F} \neq \emptyset$ is one.

The states at which P1 has a deceptive almost-sure winning strategy are called as deceptive almost-sure winning states. The exhaustive set of all deceptive almost-sure winning states is called deceptive almost-sure winning region and is denoted by DASWin. Note that deceptive almost-sure winning region cannot be defined for P2 because P2 does not know the hypergame, H.

We propose Alg. 4-1 to compute the deceptive almost-sure winning region for P1. The idea behind Alg. 4-1 is to identify and exploit the states $v = (s, X)$ at which P2's subjectively rationalizable actions $\mathsf{SRAct}_2(v)$ includes some of its non-permissive actions in the *true* game, $G(Act_1)$. To this end, we define the following sub-routines:

$$\mathsf{DAPre}_1^1(U) = \{v \in V_1 \mid \exists a \in Act_1 \text{ s.t. } \Delta(v,a) \in U\}, \tag{4-1a}$$

$$\mathsf{DAPre}_1^2(U) = \{v \in V_2 \mid \forall b \in \mathsf{SRAct}_2(v) \text{ s.t. } \Delta(v,b) \in U\}, \tag{4-1b}$$

$$\mathsf{DAPre}_2^1(U) = \{v \in V_1 \mid \forall a \in Act_1 \text{ s.t. } \Delta(v,a) \in U\}, \tag{4-1c}$$

$$\mathsf{DAPre}_2^2(U) = \{v \in V_2 \mid \forall b \in \mathsf{SRAct}_2(v) \text{ s.t. } \Delta(v,b) \in U\}. \tag{4-1d}$$

Proposition 8. *If a game state $s \in S$ is a non-deceptive almost-sure winning state for P1 then, for any $\gamma \in \Gamma$, the hypergame state $v = (s, \gamma)$ is a deceptively almost-sure winning state for P1.*

Alg. 4-1 works as follows. Following Proposition 8, we initialize the algorithm with $Z_0 = \mathsf{ASWin}_1(Act_1) \times \Gamma$ and then iteratively compute the sets C_k and Z_{k+1} for $k = 0, 1, \ldots$ until a fixed-point is reached. In the k-th iteration, the set $C_k \subseteq V \setminus Z_k$ is computed using sub-routine `Safe-2`, which identifies the subset of states in $V \setminus Z_k$ from which P1 has no strategy to exit $V \setminus Z_k$. In other words, C_k is a set of states in which P2 can enforce P1 to stay. The sub-routine `Safe-2` starts with $Y_0 = V \setminus Z_k$ and iteratively computes Y_j for $j = 0, 1, \ldots$ by identifying (i) $W_1 = \mathsf{DAPre}_2^1(Y_j)$: P1 states within Y_j, from which any action $a \in Act_1$ leads to a state in Y_j, and (ii) $W_2 = \mathsf{DAPre}_2^2(Y_j)$: P2 states within Y_j, from which any of its subjectively rationalizable action $a \in \mathsf{SRAct}_2(v)$ leads to a state in Y_j. Next, the set Z_{k+1} is computed using the sub-routine `Safe-1`, which identifies the subset of states in $V \setminus C_k$ from which P1 is ensured to visit Z_k in one-step. The sub-routine `Safe-1` starts with $Y_0 = V \setminus C_k$ and iteratively computes Y_j for $j = 0, 1, \ldots$ by identifying (i) $W_1 = \mathsf{DAPre}_1^1(Y_j)$: P1 states within Y_j from which P1 has an action to enter Y_j in one step, and (ii) $\mathsf{DAPre}_1^2(Y_j)$: P2 states within Y_j from which any subjectively rationalizable action of P2 leads to a state in Y_j. It is observed that as k increases, the set C_k shrinks while the set Z_k expands. Intuitively, this is because the states in C_k may have transitions leading outside C_k,

while remaining within $V \setminus Z_k$. If a state, say $v \in V \setminus Z_k$ that is not in C_k, is included in Z_{k+1}, then all states in C_k that have a transition going to v are excluded from C_{k+1} and have a potential to be included in Z_{k+2}. However, once the fixed-point is reached, say in iteration K, we show that all deceptive almost-sure winning states of P1 are included in Z_K. A deceptive almost-sure winning strategy can then be computed based on the proof of Thm. 4-3.

Example 5 (Example (4) contd.)**.** Consider the hypergame graph as shown in Fig. 4-3. Recall from Example (4) that Almost-Sure Winning (ASW) region is $\mathsf{ASWin}_1(Act_1) = \{s_0, s_1\}$, therefore, we have $Z_0 = \{(s_0, X_2), (s_1, X_2), (s_1, X_1)\}$ (we omit (s_0, X_1) as it is unreachable). The subjectively rationalizable actions for P2 are $\mathsf{SRAct}_2((s_2, X_1)) = \{b_1, b_2\}$ and $\mathsf{SRAct}_2((s_2, X_2)) = \{b_2\}$.

Iteration 1 of `DASW`**.** The first step is to compute C_0, *i.e.*the subset of $V \setminus Z_0$ which P2 perceives to be safe for himself. The `Safe-2` sub-routine takes 3 iterations to reach a fixed-point, at the end of which $C_0 = \{(s_2, X_2), (s_3, X_2)\}$. The next step is to compute Z_1, which the largest subset of $V \setminus C_0$ in which P1 can stay indefinitely. The `Safe-1` sub-routine takes 2 iterations to reach a fixed point. In its first iteration, DAPre_1^1 adds a state (s_3, X_1) and DAPre_1^2 adds a state (s_2, X_1) to Z_1. The interesting observation here is that (s_2, X_1) is added because the actions b_1 and b_2 are subjectively rationalizable actions for P2, both of which lead to a state in $V \setminus C_0$.

Iteration 2 of `DASW`**.** The fixed-point of `DASW` algorithm is reached in this iteration with $Z_2 = \{(s_0, X_2), (s_1, X_2), (s_1, X_1), (s_2, X_1), (s_3, X_1)\}$. The states (s_2, X_1) and (s_3, X_1) are identified as the deceptive almost-sure winning states for P1.

Using intuition from Example (5) with the observation that $\mathsf{ASWin}_1(Act_1) \subseteq \mathsf{DASWin}_1 \lfloor_S$ holds for every hypergame $\mathcal{H}$ by definition, we formalize our first key result. It establishes that using action deception could be advantageous to P1.

Theorem 4-2. *Let* $\mathsf{DASWin}_1 \lfloor_S = \{s \in S \mid v \in \mathsf{DASWin}_1$ *and* $s = v \lfloor_S\}$ *be the set of projection of the deceptively almost-sure winning states onto the game state space. There exists a hypergame*

Algorithm 4-1 Deceptive almost-sure winning region for P1.

function DASW($\mathcal{H}$)
  $Z_0 = \mathsf{ASWin}_1(Act_1) \times \Gamma$
  repeat
    $C_k = \texttt{Safe-2}(V \setminus Z_k)$
    $Z_{k+1} = \texttt{Safe-1}(V \setminus C_k)$
  until $Z_{k+1} = Z_k$
  return $\mathsf{DASWin}_1 = Z_k$
end function

function SAFE-$i(U)$
  $Y_0 = U$
  repeat
    $W_1 = \mathsf{DAPre}_i^1(Y_k)$
    $W_2 = \mathsf{DAPre}_i^2(Y_k)$
    $Y_{k+1} = Y_k \cap (W_1 \cup W_2)$
  until $Y_{k+1} = Y_k$
  return Y_k
end function

$\mathcal{H}$ for which $\mathsf{ASWin}_1(Act_1)$ *is a strict subset of* $\mathsf{DASWin}_1 \downharpoonright_S$.

Next, we proceed to prove the correctness of Alg. 4-1 by showing that from every state in DASWin_1, we can construct a deceptive almost-sure winning strategy for P1 to ensure a visit to final states with probability one. We first prove two lemmas.

Lemma 4-2. *In the i-th iteration of Alg. 4-1, P1 has a strategy to restrict the game indefinitely within Z_i, for all states in Z_i.*

Proof. ($v \in V_2$). For a P2's state in Z_i, every state $v' = \Delta(v,b)$ for a subjectively rationalizable action $b \in \mu(v)$ of P2 is in Z_i, by definition of DAPre_1^2. Hence, no action of P2 at any state $v \in Z_i$ can lead the game state outside Z_i.

($v \in V_1$). For every P1's state in Z_i, there exists an action $a \in A$ such that the successor $v' = \Delta(v,a)$ is in Z_i, by definition of DAPre_1^1. Hence, P1 always has an action, consequently a strategy, to stay within Z_i. □

Lemma 4-3. *For every state $v \in Z_{i+1} \setminus Z_i$ added in the i-th iteration of Alg. 4-1. The, there exists an action that leads into Z_i.*

Proof. Given any state $v \in V$ at the beginning of the i-th iteration, observe that it would belong to either C_{i-1}, Z_i or $V \setminus (C_{i-1} \cup Z_i)$. We will prove the statement by showing that the every new state added to Z_{i+1} has at least one transition into Z_i.

Consider i-th iteration of Alg. 4-1. The sub-routine `Safe-2` will add a P1 state $v \in V_1 \setminus Z_i$ to C_i if all the actions of P1 stay within $V \setminus Z_i$. Similarly, `Safe-2` will include a P2 state $v \in V_2 \setminus Z_i$ in C_i if all subjectively rationalizable actions of P2 lead to a state within $V \setminus Z_i$. Therefore, a state that is not included in C_i must have at least one action leading outside $V \setminus Z_i$, *i.e.*entering Z_i. In the next step, the sub-routine `Safe-1` may add new states to Z_{i+1} from the set $V \setminus C_i$. But, all states in $V \setminus C_i$ have an action entering Z_i. Hence, all new states added to Z_{i+1} satisfy the statement. □

The following observation follows immediately from Lma. 4-3.

Corollary 3. *For every $i \geq 0$, we have $Z_i \subseteq Z_{i+1}$.*

From Lma. 4-3, it is easy to see that P1 has a strategy to reach Z_i from a state added to Z_{i+1} in one-step. However, this is not true for P2. From a P2 state in Z_{i+1}, there exists a positive probability to reach Z_i because of Assumption 5. In the next theorem, we prove a stronger statement which states that from every state in Z_{i+1}, P1 can not only reach Z_i with positive probability, but with probability one.

Theorem 4-3. *From every state $v \in \mathsf{DASWin}_1$, P1 has a strategy to satisfy φ with probability one.*

Proof. For any $v \in Z_i$, $i > 1$, P1 has a strategy to stay within Z_i indefinitely, by Lma. 4-2. Furthermore, by Lma. 4-3, the probability of reaching to a state $v' \in Z_{i-1}$ from v is strictly positive. Thus, given a run of infinite length, the probability of reaching Z_{i-1} from Z_i is one. By repeatedly applying this argument, the probability of reaching Z_0 from Z_i is one. □

The deceptive almost-sure winning strategy can be constructed based on the proof of Thm. 4-3. At a P1 state $v \in V_1$, if $i \geq 1$ is the smallest integer such that $v \in Z_i$, then $\pi(v) = \{a \in Act_1 \mid v' = \Delta(v,a) \text{ and } v' \in Z_{i-1}\}$ is the deceptive almost-sure winning strategy of P1 at v. Given $\pi(v)$ is a set, P1 can select any action from this set. We also state the following two important corollaries that follow from Thm. 4-2 and Lma. 4-3.

We conclude this section with the complexity analysis of our proposed algorithm.

Theorem 4-4. *The space and time required by Alg. 4-1 scales quadratically with the size of the hypergame $\mathcal{H}$.*

4.3 Case Study: Capture-the-Flag Game on Gridworld

In this section, we illustrate the advantages of using action deception using a simplified version of capture-the-flag game [89] played over a 5×5 gridworld, like the one shown in Fig. (4-4). The gridworld is partitioned into P1 (blue) and P2 (red) territories. P1's objective in the game is to capture both the flags from P2's territory, while that of P2 is to prevent P1 from capturing the flags. We restrict P2 to move only within its own territory. Under this setting, we are interested to determine the number of game states from which P1 has a deceptive sure (almost-sure) winning strategy and compare it with the sizes of the non-deceptive sure (almost-sure) winning regions. We introduce the following notion of *value of deception*, denoted by VoD to quantify the advantage gained by P1 by using deception.

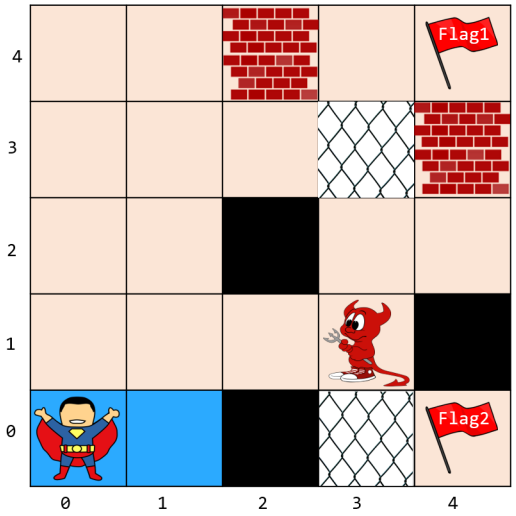


Figure 4-4. An example of capture-the-flag game between P1 (superman) and P2 (devil) played over a 5×5 grid world.

$$\mathsf{VoD} = \begin{cases} \frac{|\mathsf{DSWin}_1\downarrow_S| - |\mathsf{SWin}_1(Act_1)|}{|\mathsf{SWin}_2(Act_1)|} & \text{under deceptive sure winning condition} \\ \frac{|\mathsf{DASWin}_1\downarrow_S| - |\mathsf{ASWin}_1(Act_1)|}{|\mathsf{ASWin}_2(Act_1)|} & \text{under deceptive almost-sure winning condition} \\ 0 & \text{if } |\mathsf{ASWin}_2(Act_1)| = 0 \end{cases} \tag{4-2}$$

To understand Eq. (4-2), first, recall that P1 can win from any state in $\mathsf{ASWin}_1(Act_1)$ regardless of whether P1 uses deception or not. Thus, the benefit of deception can be quantified by counting the number of P2's winning states in the game with complete, symmetric information (*i.e.* in $\mathsf{ASWin}_2(Act_1)$) that P1 can win from by using deception. Notice that VoD takes a value between 0 and 1. $\mathsf{VoD} = 0$ represents the case when P1 gains no advantage by using deception. $\mathsf{VoD} = 1$ represents the case in which P1 gains maximum benefit that is possible by using deception, *i.e.*P1 can leverage P2's misperception to win from all of P2's winning states in $\mathsf{ASWin}_2(Act_1)$.

To demonstrate the applicability of our proposed approach to a broad range of reachability objectives, we specify P1's objective using a scLTL formula. We consider the following two scLTL objectives for P1 in this experiment.

1. P1 must capture both FLAG_1 and FLAG_2 in any order.

$$\underbrace{\Diamond \mathsf{FLAG}_1}_{\text{Eventually capture } \mathsf{FLAG}_1} \wedge \underbrace{\Diamond \mathsf{FLAG}_2}_{\text{Eventually capture } \mathsf{FLAG}_2} \tag{4-3}$$

2. P1 must first capture FLAG_1 and then capture FLAG_2. Until then, P1 must avoid colliding with P2.

$$\underbrace{(\neg \mathsf{FLAG}_2 \wedge \neg \mathsf{collide})\, \mathsf{U}\, \mathsf{FLAG}_1}_{\text{don't collide or collect } \mathsf{FLAG}_2 \text{ until } \mathsf{FLAG}_1 \text{ is collected}} \wedge \underbrace{\neg \mathsf{collide}\, \mathsf{U}\, \mathsf{FLAG}_2}_{\text{don't collide until } \mathsf{FLAG}_2 \text{ is collected}} \tag{4-4}$$

The dynamics of the capture-the-flag game are as follows. Both the players can move in 4

compass directions: `N, E, S, W`. P2 cannot enter any cell containing a wall or a fence, and *presumes this to be the case for P1 as well.* However, initially unknown to P2, P1 has the following special actions: `JumpN, JumpE, JumpS, JumpW` and `Cut`. Using the `Jump` action P1 can jump over a wall in a free cell (*i.e.*a cell not containing an obstacle, a wall or a fence) adjacent to the wall in the direction of the jump. Using the `Cut` action, P1 can convert a cell containing a fence into a free cell. Note that once a cell containing a fence becomes free, P2 can visit that cell.

Given the dynamics, we construct game and hypergame graphs. We define the game state (denoted by s) and hypergame state (denoted by v) as follows:

$$s : \big((\texttt{p1.x}, \texttt{p1.y}, \texttt{p2.x}, \texttt{p2.y}), (\texttt{f1.cut}, \texttt{f2.cut}), \texttt{turn}, \texttt{q}\big)$$
$$v : \big((\texttt{p1.x}, \texttt{p1.y}, \texttt{p2.x}, \texttt{p2.y}), (\texttt{f1.cut}, \texttt{f2.cut}), \texttt{turn}, \texttt{q}, \texttt{gamma}\big)$$

where

- `p1.x, p2.y, p1.x, p2.y` represents the position of P1 and P2 in gridworld;
- `f1.cut, f2.cut` represents whether fence 1 and fence 2 (cells $(0,3)$ and $(3,3)$ in Fig. (4-4)) are cut or intact;
- `turn` represents whether it is P1's or P2's turn at that state;
- `q` is the DFA state that encodes the progress P1 has made towards satisfying its scLTL objective;
- `gamma` is a subset of P1's action set known to P2.

For simplicity, we use the following indices to represent different subsets of P1's action sets. Hence, `gamma` takes values from 0 to 3, with $\texttt{gamma} = 3$ representing the game in which P2 has complete information.

$0 : \mathtt{N}, \mathtt{E}, \mathtt{S}, \mathtt{W},$

$1 : \mathtt{N}, \mathtt{E}, \mathtt{S}, \mathtt{W}, \mathtt{Cut},$

$2 : \mathtt{N}, \mathtt{E}, \mathtt{S}, \mathtt{W}, \mathtt{JumpN}, \mathtt{JumpE}, \mathtt{JumpS}, \mathtt{JumpW},$

$3 : \mathtt{N}, \mathtt{E}, \mathtt{S}, \mathtt{W}, \mathtt{JumpN}, \mathtt{JumpE}, \mathtt{JumpS}, \mathtt{JumpW}, \mathtt{Cut},$

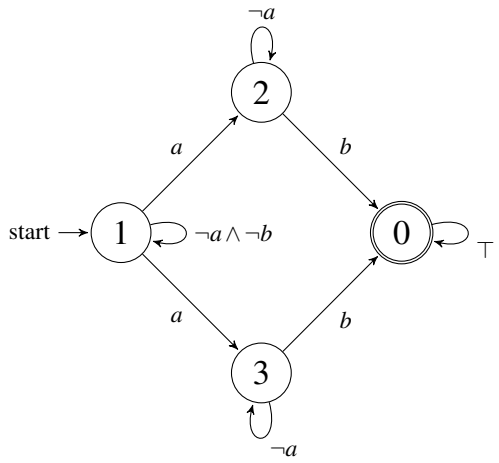
The game on graph $G(Act_1)$ is constructed using the product construction described in Ch. 2. The edges of hypergame graph follow from Def. 19. A game or hypergame state is marked as a final state whenever q is a final state in the DFA. Fig. (4-5) shows the DFAs corresponding to scLTL formulas in Eq. (4-3) and Eq. (4-4). In the figure, the final states of DFA are shown with two concentric circles.

The result of applying Alg. 4-1 on the game and hypergame graph for objective $\varphi_1 = \Diamond \mathsf{FLAG}_1 \wedge \Diamond \mathsf{FLAG}_2$ is tabulated in Table. 4-1 and that for objective $\varphi_2 = ((\neg \mathsf{FLAG}_2 \wedge \neg \mathsf{collide}) \,\mathsf{U}\, \mathsf{FLAG}_1) \wedge (\mathsf{collide} \,\mathsf{U}\, \mathsf{FLAG}_2)$ is tabulated in Table. 4-2.

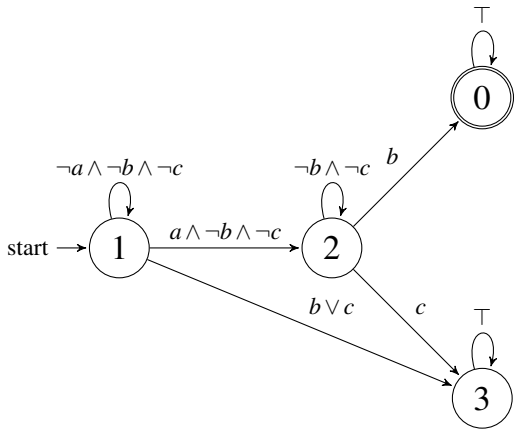
However, under the deceptive *almost*-sure winning condition, we observe that P1 can win from 9395 out of 9423 hypergame states. That is, P1 has a deceptive almost-sure winning strategy from 6370 out of 6388 game states, which is $6370 - 6133 = 237$ more states than the case when deception is *not* used. This results in $\mathsf{VoD} = 0.9294$. Similarly, for the second objective, where P1 has must capture flags in certain order and ensure that certain safety constraints are also satisfied, we observe that P1 can win from 6947 out of 6965 hypergame states. That is, P1 has a deceptive almost-sure winning strategy from 4868 out of 4880 game states which is $4868 - 4724 = 144$ more states than the number of states when deceptive mechanism is not used, thereby, resulting in $\mathsf{VoD} = 0.9230$.

Table 4-1. Comparison of deceptive and non-deceptive winning states under sure and almost-sure winning condition for P1's objective $\varphi_1 = \Diamond \mathsf{FLAG}_1 \wedge \Diamond \mathsf{FLAG}_2$.

	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert F\rvert$	$\lvert\mathsf{DASWin}_1\rvert$	$\lvert\mathsf{DASWin}_1 \downharpoonright_S\rvert$	ASWin_2	VoD
SW(G)	6388	15016	1686	-	6133	255	-
DASW($\mathcal{H}$)	9423	22181	2238	9395	6370	18	0.9294



(a) DFA of $\varphi_1 = \Diamond a \wedge \Diamond b$



(b) DFA of $\varphi_2 = ((\neg b \wedge \neg c)\,\mathsf{U}\,a) \wedge (c\,\mathsf{U}\,b)$

Figure 4-5. The sub-figure (a) shows the DFA equivalent to the scLTL formula given in Eq. (4-3) and sub-figure (b) shows the DFA equivalent to scLTL formula in Eq. (4-4). For brevity, we use $a = \mathsf{FLAG}_1$, $b = \mathsf{FLAG}_2$ and $c = \mathsf{collide}$ in the figure.

Table 4-2. Comparison of deceptive and non-deceptive winning states under sure and almost-sure winning condition for P1's objective $\varphi_2 = ((\neg \mathsf{FLAG}_2 \wedge \neg \mathsf{collide})\,\mathsf{U}\,a) \wedge (\mathsf{collide}\,\mathsf{U}\,\mathsf{FLAG}_2)$.

	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert F\rvert$	$\lvert\mathsf{DASWin}_1\rvert$	$\lvert\mathsf{DASWin}_1 \downharpoonright_S\rvert$	ASWin_2	VoD
SW(G)	4880	11449	1686	-	4724	156	-
DASW($\mathcal{H}$)	6965	16372	2238	6947	4868	12	0.9230

CHAPTER 5
SYNTHESIS WITH MISPERCEPTION OF SPECIFICATIONS

This chapter investigates the synthesis of deceptive winning strategies for the sub-class of games with incomplete information where P2 misperceives P1's true objective. We explore two approaches for analyzing these games. In the first section, we focus on characterizing the state space and synthesizing strategies when facing an ignorant or incapable P2, who does not update its perception during their interaction. In this setting, P2 is assumed to know a partial objective of P1, which it regards as P1's true objective. And, because of its ignorance or incapability, P2's perception of P1's objective remains constant during their interaction. We differentiate this case from situations where P1 deliberately prevents P2 from becoming aware of deception by labeling the synthesized strategy as *opportunistic*, as it capitalizes on the opportunities arising from P2's ignorance or incapability.

In the second section, we consider an informed P2, who is aware of its misperception of P1's objective and maintains a hypothesis set regarding the possible objectives of P1. In addition, we equip P2 with an inference mechanism, using which P2 updates its hypothesis by observing P1's behavior in the game as well as its counter-strategy.

5.1 Opportunistic Strategies in Games with Specification Misperception

5.1.1 Effect of Specification Misperception on Ignorant P2

Consider an interaction between P1 and P2 characterized by a deterministic two-player turn-based zero-sum game, $G = \langle S, Act, T, AP, L \rangle$, as defined in Def. 1. In this interaction, P1 aims to satisfy an scLTL formula φ comprising of a public component φ_1 and a private component φ_2, *i.e.* $\varphi := \varphi_1 \wedge \varphi_2$. The adversarial agent, P2, only knows the public component φ_1 and believes that P1's aim is to satisfy φ_1. Therefore, P2's objective is to prevent P1 from satisfying φ_1. Formally, the information structure in the interaction is characterized by the following assumption.

Assumption 6 (Information Structure)**.** P1 knows her complete objective $\varphi := \varphi_1 \wedge \varphi_2$. P2 knows only the public component of P1's objective, φ_1. The components S, Act, AP and L of the game arena G are commonly known to both the players.

As a result of Assumption 6, the interaction between P1 and P2 is a game with incomplete information about payoffs/specifications. Hence, P1 and P2 construct different games in their minds. Since P1 knows her true objective, she constructs a perceptual game as the product $G \otimes \mathcal{A}$, where $\mathcal{A}$ is the DFA representing the language of scLTL formula φ. On the other hand, P2 constructs his perceptual game as the product $G \otimes \mathcal{A}_1$, where $\mathcal{A}_1$ is the DFA representing the language of scLTL formula φ_1.

Notation 3. Given an scLTL formula φ, let $G(\varphi)$ denote the deterministic two-player turn-based game on a graph in which P1's objective is to satisfy φ.

Following the discussion in Sec. 2.3, the first-level hypergame representing the interaction between P1 and P2 is given by $H^1 = \langle G(\varphi), G(\varphi_1) \rangle$. Since P1 is aware that φ_2 is her private information, she is also aware that P2 misperceives her true objective. Therefore, their interaction is, in fact, a second-level hypergame.

$$H^2 = \langle H^1, G(\varphi_1) \rangle. \tag{5-1}$$

Given the hypergame H^2, we are interested to know whether P1 can exploit her superior knowledge to gain advantage over P2? Specifically, we want to determine if there exists an *opportunistic strategy* for P1 that exploits her superior knowledge to win from a state in G from which P1 cannot win using the standard sure winning strategy[1] as defined in Def. 3.

Problem 4. Given a game G and an objective $\varphi = \varphi_1 \wedge \varphi_2$ for P1 following Assumption 6, determine the set of states and the strategy using which P1 can satisfy $\varphi_1, \varphi_2, \varphi$ with a high likelihood by exploiting her superior knowledge about the private and public components of φ and P2's misperception.

Problem 4 poses two challenges to decision making. The first challenge is to identify those states in which P2's subjectively rationalizable strategy is different from his rational strategy due

1 The standard sure winning strategy is synthesized in the game $G(\varphi)$ since it does not exploit P1's superior knowledge.

to his misperception. From these states, there is a possibility that P1 might have a strategy to exploit the difference to enforce a win from an otherwise losing state. Secondly, it is possible that, from a state, either φ_1 or φ_2 is satisfiable but not both. In such a situation, which sub-formula should P1 satisfy?

5.1.2 Static Hypergame on Graph

We begin by defining a graphical model of the hypergame H^2 that incorporates the superior knowledge of P1. Using this model, we can compute P2's subjectively rationalizable strategy and use it to synthesize an *opportunistic* strategy for P1.

To construct the graphical model of hypergame H^2, observe that the language of φ is the same as the intersection of languages of φ_1 and φ_2. Hence, a DFA that represents the union of languages of φ_1 and φ_2 is sufficient to determine if a given word satisfies φ_1, φ_2 or φ. Let $\mathcal{A}_1 = \langle Q_1, \Sigma, \delta_1, q_{10}, F_1 \rangle$ and $\mathcal{A}_2 = \langle Q_2, \Sigma, \delta_2, q_{20}, F_2 \rangle$ be the DFA representing the languages of the scLTL formulas φ_1, φ_2. The DFA representing the language of φ is given by the intersection product of $\mathcal{A}_1$ and $\mathcal{A}_2$. We denote it by $\mathcal{A} = \mathcal{A}_1 \otimes \mathcal{A}_2 = \langle Q, \Sigma, \delta, q_0, F_{12} \rangle$, where $Q = Q \times Q_2$, $\delta((q_1, q_2), \sigma) = (\delta(q_1, \sigma), \delta(q_2, \sigma))$, $q_0 = (q_{10}, q_{20})$ and $F = F_1 \times F_2$.

Definition 23 (Hypergame on a Graph)**.** The hypergame on a graph representing the hypergame H^2 of Eq. (5-1) is a deterministic two-player turn-based game on a graph,

$$\mathcal{H} = \langle V, Act, \Delta, v_0, \mathcal{F} \rangle$$

where

- $V = S \times Q_1 \times Q_2$ is the set of states;
- Act is the set of P1 and P2 actions;
- $v_0 \in V$ is an initial state;
- $\Delta : (V_1 \times Act_1) \cup (V_2 \times Act_2) \to V$ is the deterministic transition function that maps a state $v = (s, q_1, q_2)$ and an action $a \in Act$ to a state $v' = (s', q_1', q_2') = \Delta(v, a)$, where

$s' = T(s,a),\ q_1' = \delta_1(q_1, L(s'))$ and $q_2' = \delta_2(q_2, L(s'))$;

- $\mathcal{F} = (S \times F_1 \times Q_2) \cup (S \times Q_2 \times F_2)$ is the set of final states.

The final states $\mathcal{F}$ can be partitioned into three parts: (i) $\mathcal{F}_1 = S \times F_1 \times (Q_2 \setminus F_2)$: the states that denote satisfaction of φ_1 but not φ_2, (ii) $\mathcal{F}_2 = S \times (Q_1 \setminus F_1) \times F_2$: the states that denote satisfaction of φ_2 but not φ_1, and (iii) $\mathcal{F}_{12} = S \times F_1 \times F_2$: the states that denote satisfaction of φ, that is, they satisfy φ_1 and φ_1. Note that the sets $\mathcal{F}_1, \mathcal{F}_2$ and $\mathcal{F}_{12}$ are mutually exclusive and exhaustive.

5.1.3 Characterization of State Space

The following proposition establishes the equivalence between the sure winning strategies in the games $G(\varphi_1), G(\varphi_2), G(\varphi)$ and the hypergame $\mathcal{H}$.

Proposition 9. *The following statements hold.*

1. *There exists a sure winning strategy to visit $\mathcal{F}_1$ from a state (s, q_1, q_2) in the hypergame $\mathcal{H}$ if and only if there exists a sure winning strategy to visit F_1 from the state (s, q_1) in the game $G(\varphi_1)$.*

2. *There exists a sure winning strategy to visit $\mathcal{F}_2$ from a state (s, q_1, q_2) in the hypergame $\mathcal{H}$ if and only if there exists a sure winning strategy to visit F_2 from the state (s, q_2) in the game $G(\varphi_2)$.*

3. *There exists a sure winning strategy to visit $\mathcal{F}_{12}$ from a state (s, q_1, q_2) in the hypergame $\mathcal{H}$ if and only if there exists a sure winning strategy to visit F from the state (s, q_1, q_2) in the game $G(\varphi)$.*

Since P2 is only aware of the public component φ_1 of P1's true objective φ, he would play an subjectively rationalizable strategy to prevent the game from reaching the final states $\mathcal{F}_1 \cup \mathcal{F}_{12} = S \times F_1 \times Q_2$ that denote satisfaction of φ_1. However, due to misperception, P2 is unaware that P1 may have a preference over visiting the subset $\mathcal{F}_{12}$ over visiting $\mathcal{F}_1$. As a result, intentionally P2's subjectively rationalizable strategy neither prevents P1 from satisfying φ_2 nor

does it restrict P1 from satisfying her more preferred outcome. But could it *unintentionally* prevent P2 from achieving φ_2 or φ?

To see how P2's unawareness affects the interaction, consider the partition of V induced by the sure winning regions of the three sub-games: $G(\varphi_1), G(\varphi_2)$ and $G(\varphi)$. Since visiting any state in $\mathcal{F}_1 \cup \mathcal{F}_{12}$ denotes satisfaction of φ_1, by Proposition 9, $\mathsf{SWin}_1(\mathcal{F}_1 \cup \mathcal{F}_{12})$ is the set of winning states in the game $G(\varphi_1)$. Similarly, $\mathsf{SWin}_1(\mathcal{F}_2 \cup \mathcal{F}_{12})$ is the set of winning states in the game $G(\varphi_2)$, and $\mathsf{SWin}_1(\mathcal{F}_{12})$ is the set of winning states in the game $G(\varphi)$. The containment relation among the sure winning regions follows immediately.

Proposition 10. $\mathsf{SWin}_1(\mathcal{F}_{12}) \subseteq \mathsf{SWin}_1(\mathcal{F}_1 \cup \mathcal{F}_{12})$ *and* $\mathsf{SWin}_1(\mathcal{F}_{12}) \subseteq \mathsf{SWin}_1(\mathcal{F}_2 \cup \mathcal{F}_{12})$.

Since the deterministic two-player zero-sum games are determined (Proposition 1), the containment relation between the sure winning regions of P2 follows from Proposition 10.

Corollary 4. $\mathsf{SWin}_2(\mathcal{F}_{12}) \supseteq \mathsf{SWin}_2(\mathcal{F}_1 \cup \mathcal{F}_{12})$ *and* $\mathsf{SWin}_2(\mathcal{F}_{12}) \supseteq \mathsf{SWin}_2(\mathcal{F}_2 \cup \mathcal{F}_{12})$.

Corollary 4 yields two key insights. The first insight, which solidifies our hypothesis, is that P2 can win from a smaller number of states using his subjectively rationalizable rationalizable strategy than he could if he had complete information about P1's objective. The second insight is that P2 could unintentionally prevent P1 from satisfying the private component of P1's objective, φ_2. This is because P2's subjectively rationalizable strategy prevents the game from reaching $\mathcal{F}_1 \cup \mathcal{F}_{12}$, which includes a subset of final states, $\mathcal{F}_{12}$, that denote satisfaction of φ_2. For example, consider 3 states $\{v_1, v_2, v_3\}$ such that s_1 is P2 state with actions a_1, a_2, a_3 that transitions the game to v_1, v_2, v_3, respectively. Suppose that $v_2 \in \mathcal{F}_1$ and $v_3 \in \mathcal{F}_{12}$. Then, P2's subjectively rationalizable strategy is to choose a_1 at state s_1 since both actions a_2, a_3 will lead to P1 satisfying φ_1. Therefore, unintentionally P2 prevents the game from satisfying φ_2 by marking action a_3 to be non-permissive.

We now characterize the state space of the hypergame $\mathcal{H}$ by labeling each state in V with a win-label.

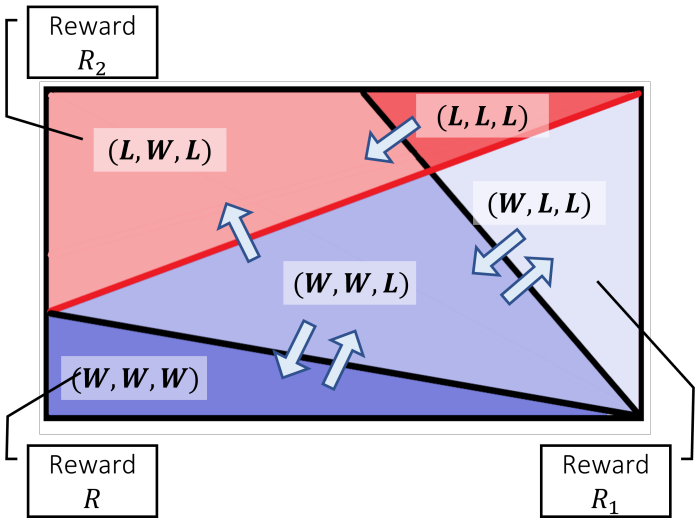


Figure 5-1. State space characterization. Arrows indicate whether going from one partition to another could be rational or not.

Definition 24 (Win-label)**.** the win-labeling function $\lambda : V \to \{0,1\}^3$ maps every state $v \in V$ in the hypergame $\mathcal{H}$ to an ordered 3-tuple denoting whether the state v is winning (1) or losing (0) for P1 in the games $G(\varphi_1)$, $G(\varphi_2)$, and $G(\varphi)$, respectively.

Intuitively, the win-label for a state $v \in V$ captures the perception and knowledge of players about whether they can win and whether their opponent can win from the state v . The first component of the win-label captures what P2 thinks whereas the whole 3-tuple is known by P1 given her superior knowledge. For example, if a state $v \in V$ is winning for P1 in the game $G(\varphi_1)$ and the game $G(\varphi_2)$, but losing in the game $G(\varphi)$, then its win-label is $\lambda(v) = \{1,1,0\}$.

The win-labeling function can assign to every state $v \in V$, a unique label from $2^3 = 8$ possible labels. We analyze each possible label separately to understand which of the objectives φ_1, φ_2 or φ should P1 try to satisfy from a state with a particular win-label.

Case I: $(\lambda(v) = (0,0,0))$ The state v is losing for P1 in the games $G(\varphi_1), G(\varphi_2)$ and $G(\varphi)$, *i.e.*P2 has an subjectively rationalizable sure winning strategy that prevents P2 from satisfying φ_1. Therefore, in this case, P1 can try to satisfy only φ_2, since she will not be able to satisfy either φ_1 or φ.

Case II: $(\lambda(v) = (0,1,0))$ The state is losing for P1 in the games $G(\varphi_1)$ and $G(\varphi)$, but winning

in game over φ_2. That is, P2 has an subjectively rationalizable sure winning strategy that prevents P2 from satisfying φ_1. Therefore, in this case, P1 must satisfy only φ_2. It cannot satisfy either φ_1 or φ.

Case III: ($\lambda(v) = (1,0,0)$) The state is losing for P1 in the games $G(\varphi_2)$ and $G(\varphi)$, but winning in game over φ_1. That is, P2 believes that it has lost the game. In this case, P1 is at least guaranteed to satisfy φ_1. But she may have an *opportunity* to satisfy either φ_2 or φ since P2's subjectively rationalizable strategy does not intentionally prevent her from doing so.

Case IV: ($\lambda(v) = (1,1,0)$) The state is winning for P1 in the games $G(\varphi_1)$ and $G(\varphi_2)$, but losing in game $G(\varphi)$. That is, P2 believes that it has lost the game. This case presents an interesting decision problem where P1 has to choose between satisfying φ_1 or φ_2 since it does not have a sure winning strategy to satisfy both. However, there may be an *opportunity* to satisfy φ.

Case V: ($\lambda(v) = (1,1,1)$) This is a trivial case, in which P1 can satisfy φ by following the standard sure winning strategy. That is, P1 satisfies φ regardless of the strategy and perception of P2.

Cases VI-VIII: ($\lambda(v) = (0,0,1)$, $(0,1,1)$, or $(1,0,1)$) These cases are not possible, because P1 must be winning in both the specifications, φ_1 and φ_2, to be winning in φ [77, Lma. 1].

5.1.4 Synthesis of Opportunistic Strategy

P2's subjectively rationalizable strategy. Since P2's aim is to prevent P1 from satisfying φ_1, P2's subjectively rationalizable strategy is a permissive strategy in his perceptual game. Conventionally, a permissive strategy is only defined at the winning states of a player. However, in many real-life situations, the interaction between the players does not terminate even if the state is sure losing for a player. Hence, without loss of generality, we assume that a player chooses every available action with a strictly positive probability at a losing state.

Therefore, P2's randomized subjectively rationalizable strategy at a state $v \in V$ is a probability distribution over the set $M(v)$ defined as follows:

$$M(v) = \begin{cases} \{a \in Act_2 \mid \Delta(v,a) \in \mathsf{SWin}_2(\mathcal{F}_1 \cup \mathcal{F}_{12})\} & \text{if } \lambda(v) = (0,\cdot,\cdot) \\ Act_2 & \text{otherwise} \end{cases}$$

P1's opportunistic strategy. By knowing P2's subjectively rationalizable strategy in the hypergame $\mathcal{H}$, P1 can compute her randomized opportunistic strategy. Intuitively, from a state $v \in V$, we expect an opportunistic strategy to yield at least as much payoff as the sure winning strategy. If possible, it should yield a higher payoff than P1's sure winning strategy π_{12} in $G(\varphi)$.

To formalize this idea, we define payoffs $r_1, r_2, r_{12} \in \mathbb{R}^+$ that P1 receives for satisfying $\varphi_1, \varphi_2, \varphi$, respectively. We assert the condition that $r_{12} \geq r_1 + r_2$ to capture that satisfying φ is strictly preferred to satisfying either φ_1 or φ_2 individually. Given that the interaction is zero-sum, P2 receives the payoff $-r_1$ if P1 satisfies φ_1 and a payoff r_1 otherwise. Owing to misperception, P2 incorrectly thinks his payoff when P1 satisfies $\varphi_2 \wedge \neg\varphi_1$ is also r_1.

The relation between r_1 and r_2 quantifies the preference of P1 to satisfy φ_1 and φ_2. When $r_1 = r_2$, then P1 is considered to be indifferent to satisfying either φ_1 or φ_2. Otherwise, if $r_1 > r_2$, then φ_1 is strictly preferred over φ_2, and vice versa.

As a result, the problem of synthesizing an opportunistic strategy is reduced to maximizing the payoff in the following MDP constructed by marginalizing the two player game with P2's randomized strategy μ.

Definition 25 (Hypergame MDP)**.** Given the hypergame on graph $\mathcal{H} = \langle V, Act, \Delta, \mathcal{F} \rangle$ and P2's subjectively rationalizable strategy μ given his perception, the opportunistic planning reduces to an MDP, defined by

$$\mathcal{H}^{\mu} = \langle V_1 \cup \{\mathsf{sink}_{12}, \mathsf{sink}_1, \mathsf{sink}_2\}, Act_1 \cup \{\mathsf{stop}\}, P, R \rangle,$$

where $V_1 = (S_1 \times Q_1 \times Q_2)$ is a set of states where P1 chooses an action. The states

sink_{12} and sink_1 are speial absorbing states which denote that P1 will thereafter follow the respective sure winning strategy in the games $G(\varphi)$ and $G(\varphi_1)$. The probabilistic transition function P and the payoff function R are defined based on the win-label of a state $v \in V_1$ as follows,

- $\lambda(v) \in (0,0,0)$: All feasible actions of P1 are enabled at v. The special action stop is not enabled. Given an action $a_1 \in Act_1$, the transition probability function is given by

$$P(v' \mid v, a_1) = \sum_{a_2 \in Act_2} \mathbf{1}_{\{v'\}}(\Delta(\Delta(v, a_1), a_2))) \cdot \mu(v)(a_2).$$

- $\lambda(v) \in (1,0,0)$: Only actions that have *zero* probability of reaching a state with win-label $(0,0,0)$ are enabled. In other words, P1 is guaranteed to remain within the winning region $\mathsf{SWin}_1(\mathcal{F}_1 \cup \mathcal{F}_{12})$. Therefore, φ_1 will at least be satisfied. The special action stop is enabled. Given an action $a_1 \in Act_1$, the transition probability function is given by

$$P(v' \mid v, a_1) = \sum_{a_2 \in Act_2} \mathbf{1}_{\{v'\}}(\Delta(\Delta(v, a_1), a_2))) \cdot \mu(v)(a_2).$$

  For action stop, the game transitions to the absorbing state sink_1 with probability one, *i.e.*

$$P(\mathsf{sink}_1 \mid v, \mathsf{stop}) = 1.$$

  after which P1 must switch to its winning strategy in $G(\varphi_1)$. The payoff for reaching the absorbing state sink_1 is defined as $R(\mathsf{sink}_1) = r_1$.

- $\lambda(v) \in (0,1,0)$: Only the special action stop is enabled. Using this action, the game transitions to the absorbing state sink_2 with probability one, *i.e.*

$$P(\mathsf{sink}_2 \mid v, \mathsf{stop}) = 1.$$

  The payoff for reaching the absorbing state sink_2 is defined as $R(\mathsf{sink}_2) = r_2$.

- $\lambda(v) \in (1,1,1)$: Only the special action stop is enabled. Using this action, the game transitions to the absorbing state sink_{12} with probability one, *i.e.*

$$P(\mathsf{sink}_{12} \mid v, \mathsf{stop}) = 1.$$

The payoff for reaching the absorbing state sink_2 is defined as $R(\mathsf{sink}_2) = r_2$.

- $\lambda(v) \in (1,1,0)$: Any action that does not lead into partition $(0,0,0)$ is enabled. The special action stop is also enabled. Given an action $a_1 \in Act_1$, the transition probability function is given by

$$P(v' \mid v, a_1) = \sum_{a_2 \in Act_2} \mathbf{1}_{\{v'\}}(\Delta(\Delta(v, a_1), a_2))) \cdot \mu(v)(a_2).$$

For the action stop, P1 transitions to the absorbing state sink_1 if $r_1 \geq r_2$ and to sink_2 otherwise. The payoff received by P1 on reaching the absorbing state sink_1 is $R(\mathsf{sink}_1) = r_1$ and on reaching the absorbing state sink_2 is $R(\mathsf{sink}_2) = r_2$.

The optimal opportunistic strategy π for P1 is the one that solves

$$\max_{\pi} \mathbb{E}\left[\sum_{t=1}^{T} R(v_t)\right], \tag{5-2}$$

where T is the first time when an absorbing state is reached. The rationale behind defining absorbing states is to provide P1 with a mechanism to decide whether it wants to explore the state space to find an opportunity or settle for a sub-optimal payoff by satisfying a sub-specification. We define the set of states $\{v \in V \mid \lambda(v) \in \{(0,1,0),(1,1,1)\}\} \cup \{\mathsf{sink}_1, \mathsf{sink}_{12}\}$ as absorbing in the hypergame MDP, $\mathcal{H}^{\mu}$.

Theorem 5-1. *There may exist an opportunistic strategy, which satisfies Eq. (5-2), using which P1 can satisfy φ from a state $v \in V \setminus \mathsf{SWin}_1(\mathcal{F}_{12})$.*

Intuitively, Thm. 5-1 proves our hypothesis that P1 may have a strategy that leverages P2's misperception to satisfy φ from a state that is sure-losing for P1 to satisfy φ, if P2 knew P1's true

objective. Naturally, the opportunistic strategy need not exist from all sure-losing states. The following theorem establishes that the time and space complexity of computing an opportunistic strategy is the same as that of reactive synthesis.

Theorem 5-2. *The time and space required to synthesize an opportunistic strategy scales linearly with the size of hypergame $\mathcal{H}$.*

It is also noted that the opportunistic synthesis computes a strategy to satisfy φ, φ_1, and φ_2 in order of preference given by the reward function.

5.1.5 Case Study: Robot Motion Planning

We illustrate our approach using a gridworld example as shown in Fig. (5-2). P1 (blue agent) is controllable, whereas P2 (red agent) is the adversary. P1's objective is to visit two regions, A (green cell) and B, while avoiding obstacles O (black cells). P2's objective is to prevent R2D2 from completing her task. However, P2 only knows that P1 wants to visit A. He is unaware that P1's objective includes visiting B as well. Hence, letting $\varphi_1 = \neg O \,\mathcal{U}\, A$ and $\varphi_2 = \neg O \,\mathcal{U}\, B$, the objective of P1 is $\varphi = \varphi_1 \wedge \varphi_2$, whereas P2's objective is to prevent P1 from satisfying φ_1. This defines the information asymmetry in the interaction. The action sets of P1 and P2 are as follows:

$$Act_1 = \{\texttt{N}, \texttt{S}, \texttt{E}, \texttt{1}, \texttt{NE}, \texttt{N1}, \texttt{SE}, \texttt{SW}\},$$
$$Act_2 = \{\texttt{N}, \texttt{S}, \texttt{E}, \texttt{1}, \texttt{STAY}\},$$

where `N, E, ..., SW` denote the standard actions to move in 8-neighborhood in a gridworld. The action `STAY` allows P2 to remain in the same cell.

Given the 20 obstacle-free cells of gridworld in Fig. (6-2) and the action sets, we construct the transition system with $20 \times 20 \times 2 = 800$ states. The automaton equivalent to $\neg O \,\mathcal{U}\, X$ for $X = A, B$ is shown in the Fig. (5-3). We prune unsafe actions that result in P1 to visiting an obstacle and thus exclude the transitions labeled O and the state 2 in computing the transition system. Therefore, the hypergame $\mathcal{H}$ has $800 \times 2 \times 2 = 3200$ states, where we keep track of both sub-specification using two automata. Consequently, each sub-game, $G(\varphi_1)$ and $G(\varphi_2)$, has

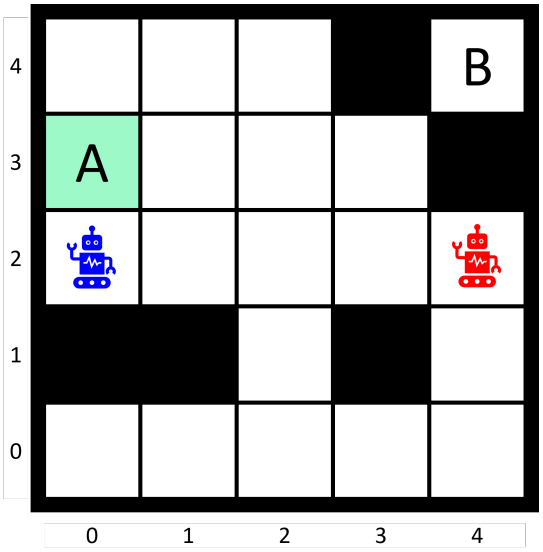


Figure 5-2. Game arena.

$800 \times 2 \times 1 = 1600$ final states and the game $G(\varphi)$ has 800 final states. Applying Alg. 2-1 for each of the three games generates the winning regions with sizes: $|\mathsf{SWin}_1(\mathcal{F}_1 \cup \mathcal{F}_{12})| = 2491$, $|\mathsf{SWin}_1(\mathcal{F}_2 \cup \mathcal{F}_{12})| = 2527$, and $|\mathsf{SWin}_1(\mathcal{F}_{12})| = 1831$.

Given the three winning regions, we first validate that the state-space is indeed partitioned in five regions as discussed in Sec. 5.1.3. For every state in the $\mathcal{H}$, we assign a win-label to it by determining the winning regions in which the state appears. The result is tabulated in 5-1. We observe that the state-space is partitioned into exactly five regions.

Table 5-1. Partition of game state-space due to information asymmetry.

Partition	**Number of States**
(1, 1, 1)	1831
(1, 1, 0)	181
(1, 0, 0)	479
(0, 1, 0)	515
(0, 0, 0)	194
(1, 0, 1)	0
(0, 1, 1)	0
(0, 0, 1)	0

Using the five partitions, we construct the hypergame MDP as defined in Def. 25. We define the randomized strategies for P1 as follows: for every state with win-label of $(0, \cdot, \cdot)$, we assume μ to be a uniform distribution over all safe actions; *i.e.* the actions that, with probability *one*, lead to

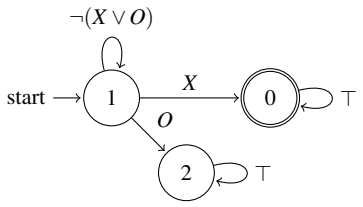


Figure 5-3. The automaton for $\neg O \,\mathcal{U}\, X$, where $X \in \{A, B\}$.

a state with a win-label of type $(0,\cdot,\cdot)$. We define μ by assigning an arbitrary distribution over all feasible actions from a state within partitions $(1,\cdot,\cdot)$. Given the hypergame MDP states and P2's strategy μ, the transition probabilities are determined based on win-label of the state and the corresponding expression for $P(v' \mid v,a)$ is provided in Def. 25. We compute the value function and opportunistic strategy using the value iteration algorithm [90].

Next, we illustrate the decision process in the hypergame MDP. Let the initial configuration be such that P1 is at the cell $(0,2)$, and P2 is at $(4,2)$ as shown in Fig. (6-2). Therefore, the initial state in the hypergame MDP is $v_0 = (((0,2),(4,2),0),1,1)$. We define the payoff for visiting A as $r_1 = 200$ and that of visiting B as $r_2 = 100$. With this initial configuration we simulate the interaction between P1 and P2, where P1 uses the opportunistic strategy π and P2 uses the strategy μ. We run the simulation for 100 times. We highlight a part of one of the runs obtained

Table 5-2. A decision table for state $(((0,2),(4,2),0),1,1)$ with value 285.03 and strategy to choose action `N`.

Act	Next State	Partition	Prob	Value
	(((0, 3), (4, 2), 0), 0, 1)	(1, 0, 0)	0.03	288.99
N	(((0, 3), (3, 2), 0), 0, 1)	(1, 0, 0)	0.36	290.20
	(((0, 3), (4, 1), 0), 0, 1)	(1, 1, 1)	0.61	288.99
	(((1, 2), (4, 1), 0), 0, 1)	(1, 1, 0)	0.25	0
E	(((1, 2), (3, 2), 0), 1, 1)	(1, 0, 0)	0.73	297.41
	(((1, 2), (4, 2), 0), 1, 1)	(1, 1, 0)	0.02	0
	(((1, 3), (3, 2), 0), 1, 1)	(1, 0, 0)	0.38	259.42
NE	(((1, 3), (4, 2), 0), 1, 1)	(1, 1, 0)	0.18	285.03
	(((1, 3), (4, 1), 0), 1, 1)	(1, 1, 0)	0.44	299.25

from simulation.

$$v_0 = (((0,2),(4,2),0),1,1) \text{ with } \lambda(v_0) = (1,0,0)$$
$$v_1 = (((0,3),(3,2),0),0,1) \text{ with } \lambda(v_1) = (1,0,0)$$
$$v_2 = (((1,2),(2,2),0),0,1) \text{ with } \lambda(v_2) = (1,1,1)$$

Table. 5-2 provides an insight into P1's decision process. It shows the enabled actions, possible next states and their respective partitions, the probability of reaching those states and the value of those states. Based on the value iteration, the value of initial state v_0 is 285.03, while the optimal strategy is to select action `N`, which has a high likelihood to reach a $(1,1,1)$ state. Note that by choosing action `E`, if P1 reaches a state with value 0, then it chooses to settle for sub-optimal payoff of $r_1 = 200$ by satisfying only φ_1. Hence, the action `N` is preferred over `E`. A similar argument can be given for the action `NE`.

We now point out the key advantage of the opportunistic synthesis over reactive synthesis as highlighted in Thm. 5-1. Observe that the initial state is losing in the game $G(\varphi)$ for P1. Therefore, if P1 uses reactive synthesis approach, it will give up instantaneously and get no payoff. On the contrary, with opportunistic synthesis, P1 could leverage the misperception of P2 to start from a losing state in $G(\varphi)$ and satisfy φ.

We highlight that the construction of hypergame MDP is such that P1 behaves rationally and tries to maximize the payoff. Given the initial state in partition $(1,0,0)$, it could have chosen the stop action and switched to the winning strategy in $G(\varphi_1)$ to get a payoff of $r_1 = 200$. Instead, P1 continues to explore to find an opportunity to get a payoff of $r = r_1 + r_2 = 300$.

We conclude this section by counting the number of states with opportunities. This is done by counting the number of hypergame MDP states with non-zero value. Recall that we label the absorbing states in the hypergame MDP as absorbing with a fixed payoff. Therefore, they always have fixed value of *one*. We find that there are a total of 1245 absorbing states and 312 states with opportunities. This implies that out of 1600 total states, there are $1600 - (1245 + 312) = 43$ states

with no opportunities. In other words, not all losing states of P1 in the reactive game $G(\varphi)$ have opportunities.

5.2 Deceptive Strategies under Specification Misperception

5.2.1 Effect of Specification Misperception on Informed P2

In this section, we consider the case when P2 has incomplete information about P1's temporal logic objective and is aware of it. We introduce a *hypothesis space* for P2, denoted by X. The set X can be discrete and finite. The set X can be a finite set of scLTL formulas that P2 believes that P1's true objective is one of these. The hypothesis space X can also be continuous. For example, each $x \in X$ is a distribution over a subset of scLTL formulas Φ so that $x(\varphi)$ is the probability that P2 believes $\varphi \in \Phi$ to be P1's true objective. For the time being, we do not restrict the set X. In practice, a hypothesis space can be constructed from observations of any previous interactions or from the threat modeling [91] given P2's understanding of their interaction and potential objectives of an adversary.

Assumption 7 (Information Structure)**.** The asymmetrical information between players is introduced as follows:

- P1's objective is φ_1.
- P2 does not know φ_1 but has an initial hypothesis x_0 and a hypothesis space X about P1's objective.

The assumption describes scenarios commonly encountered in practice for both cooperative and adversarial interactions. For example, in a contested search and rescue mission, a search team has a sequence of waypoints that need to be visited according to a temporal order. The opponent may know the set of waypoints but is unclear about the team's temporal objective. The problem we aim to solve is stated informally as follows.

Problem 5. Given an adversarial encounter between P1 and P2 under information asymmetry as defined by Assumption 7, how to compute a strategy for P1 that maximizes the probability of satisfying φ_1 while a rational P2 responds optimally given P2's knowledge of the game?

Next, we introduce the modeling framework of hypergames and present a solution concept for a class of hypergames to solve P1's strategy.

5.2.2 Dynamic Hypergame on Graph

To characterize P2's evolving perceptual game, we introduce an inference function.

Definition 26 (Inference)**.** Assuming P2 has complete observation on the game plays, a *perfect recall inference* function $\eta : X \times \mathsf{PrefPaths} \to X$ maps a hypothesis $x \in X$ and an observation (a history) $h \in \mathsf{PrefPaths}$ to a new hypothesis $x' = \eta(x,h) \in X$.

Anticipating that P2 will respond with evolving hypothesis, P1 must calculate its moves to steer P2's inferred hypothesis and the resulting strategy. For the time being, we assume that P1 knows P2's inference mechanism and initial hypothesis, and study how P1 can exploit P2's incomplete knowledge and inference mechanism for strategic advantage.

We introduce a *transition system of P1's level-1 hypergame* to simultaneously capture the changes in game states given players' actions and the evolving perceptual game of P2.

Definition 27 (Transition System of P1's Level-1 Hypergame)**.** Given the transition system $TS = \langle S, A, P, s_0, \mathcal{AP}, L \rangle$, the DFA $\mathcal{A} = \langle Q, \Sigma, \delta, \iota, F \rangle$ that corresponds to P1's scLTL specification φ_1, and P2's hypothesis space X, the transition system of P1's level-1 hypergame is a tuple

$$\mathcal{H} = \langle V, A, \Delta, (s_0, h_0, q_0, x_0), \mathcal{F} \rangle,$$

where the components of hypergame transition system are defined as follows.

- $V = S \times \mathsf{PrefPaths} \times Q \times X$ is the set of states. Every state $v = (s,h,q,x) \in V$ has four components:
  - s is the state.
  - $h \in \mathsf{PrefPaths}$ is a history terminating in state $s \in S$.
  - $q \in Q$ is the automaton state for keeping track of P1's progress towards satisfying φ_1.

- $x \in X$ represents the hypothesis of P2 given the history h.

- A is the set of joint actions.

- $\Delta\colon V \times A \to \mathcal{D}V$ is a probabilistic transition function defined as follows. Consider $v = (s,h,q,x)$ and $v' = (s',has',q',x')$, where has' is the history h appended with the new action a and state s',

$$\begin{aligned}\Delta(v' \mid v,a) =& P(s' \mid s,a)\mathbf{1}(\delta(q,L(s')) = q') \\ & \cdot \mathbf{1}(\eta(x,has') = x'),\end{aligned}$$

  where $\mathbf{1}(E)$ is the indicator function that returns 1 if the statement E is true, and 0 otherwise.

- (s_0,h_0,q_0,x_0) is the initial state that includes the initial state in the transition system TS, the current history that consists of the initial state only, *i.e.*, $h_0 = s_0$, $q_0 = \delta(\iota,L(s_0))$, and P2's initial hypothesis x_0.

- $\mathcal{F} = S \times \mathsf{PrefPaths} \times F \times X$ is the set of final states for P1.

The transition function is understood as follows: Given a history h ending in the current state s and a joint action $a \in A$, the probability of reaching the next state s' is determined by $P(s' \mid s,a)$ in the transition system. Upon reaching s', P2 updates its hypothesis to $x' = \eta(x,has')$ (here we assume the entire history is used for this update). Also, the transition in the specification automaton is triggered to reach state q' from state q given the label of the new state s'.

It is observed that the hypergame transition system in Def. 27 captures the dynamic evolution of P2's viewpoint. The history has time indices implicitly encoded. For example, a history $s_0a_0s_1a_1\ldots s_t$ is a history up to time step t.

Given P2's perceptual game evolving given the history and the inference function, P2 employs a *Behaviorally Subjectively Rationalizable (BSR)* strategy, defined as follows.

Definition 28 (Behaviorally Subjectively Rationalizable Strategy)**.** A strategy $\pi_2^{B,2}\colon \mathsf{PrefPaths} \to \mathcal{D}A_2$ is behaviorally subjectively rationalizable for P2 if

$$\pi_2^{B,2}(h) = \pi_2^{*,x,2}(h),$$

where $x = \eta(x_0, h)$, and $\pi_2^{*,x,2}\colon \mathsf{PrefPaths} \to \mathcal{D}A_2$ is a subjectively rationalizable strategy for P2 in the hypergame $H^2(x)$.

Intuitively, playing a BSR strategy means that for any history h, P2 plays the subjectively rationalizable strategy corresponding to its hypothesis constructed from the history h and its initial hypothesis. It is noted that the BSR strategy for P2 always exists in the class of hypergames studied herein. In [43], the author states the condition for the existence of the subjectively rationalizable strategy as follows: P1 never excludes an action from P2's action set in P1's own perceptual game, where P1 thinks in P2's perceptual game P2 believes this action is rationalizable [92] to P2. In the class of hypergames considered, P2's subjectively rationalizable strategy exists as P2's perceptual game is a zero-sum game.

When X is finite, the hypergame transition system has a countably infinite set of states. This is because a history can be of a finite but unbounded length. The entire history is maintained as a part of the state due to the general definition of the inference mechanism. In the next section, we show for some special cases of the interactions, a state aggregation can be performed in the hypergame transition system to reduce the infinite state space to a finite state space.

5.2.3 Synthesis of Deceptive Strategy

Given that P2 uses a BSR strategy, P1 can play deceptively by influencing P2's hypothesis so that P2's actions given P2's hypothesis can be advantageous for P1. To make P1's planning problem tractable, we introduce inference-equivalent histories so as to aggregate the countably infinite states of the transition system $\mathcal{H}$ of P1's level-1 hypergame into a finite state set.

Definition 29 (Inference-equivalent Histories)**.** Given an inference function $\eta : X \times \mathsf{PrefPaths} \to X$ and a hypothesis x, two histories h_1 and h_2 are said to be (η, x)-equivalent

if $\eta(x,h_1) = \eta(x,h_2)$ and for any $h' \in (A \times S)^+$, $\eta(x,h_1h') = \eta(x,h_2h')$. The set of histories equivalent to $h \in \mathsf{PrefPaths}$ given hypothesis x is denoted by $[\![h]\!]_x$. If the equivalence between histories can be defined to be independent of the current hypothesis, that is, *for any* pair of hypotheses $x,x' \in X$, if h_1,h_2 are (η,x)-equivalent, then h_1,h_2 are also (η,x')-equivalent, then we say that the two histories h_1 and h_2 are η*-equivalent*. The set of histories η-equivalent to $h \in \mathsf{PrefPaths}$ is denoted by $[\![h]\!]$.

We consider a subset of dynamic hypergames which satisfy the following assumption.

Assumption 8.

1. The hypothesis space X is discrete and finite.
2. The inference function η has a finite domain. That is, the set of histories are grouped into a finite set of *inference-equivalent classes* (see Def. 29).
3. For any $x \in X$, P2 selects a quantal response strategy in the zero-sum game $G(x)$ with a response parameter known to P1[2].
4. For any $x \in X$, P2's strategy in game $G(x)$ is memoryless.

Assumption 8-1) and 8-2) ensure the planning state space in $\mathcal{H}$ can be aggregated into a finite set. Assumption 8-3) enables us to only need to consider one SR strategy for P2 in game $G(x)$, for each $x \in X$. It is noted that if P2 takes the deterministic SR strategy instead of the quantal response, there may be multiple strategies. There are two possible approaches to deal with multiple equilibra. The first one is that P1 must learn from online interaction about which SR strategy is employed by P2 and adapt P1's deceptive strategy. However, this adaptive deception requires further study of online optimization and regret analysis. The second one is that the deceptive planning algorithm should be robust for a range of possible equilibria strategies used by P2. Adaptive and robust deceptive planning are future extensions for this work.

Next, we formally state the deceptive planning problem for a subclass of dynamic hypergames.

2 At each state, the quantal response strategy selects an action that is proportional to the exponential of λ-times the expected future payoffs from that state given the chosen action. The parameter λ is called the response parameter [93].

Problem 6. Given Assumptions 7 and 8, compute the optimal deceptive strategy for P1 in the dynamic hypergame $\mathcal{H}$, provided that P2 follows a BSR strategy.

We leverage the hierarchy of reasoning in level-2 hypergames and develop a two-step approach: Firstly, we construct P2's BSR strategy according to Def. 28: for each $x \in X$, we solve P2's subjectively rationalizable strategy $\pi_2^{*,x,2}$ in the static hypergame $H^2(x)$. P2's BSR strategy is computed from the set of subjectively rationalizable strategies given P2's evolving hypothesis (see Def. 28). Secondly, we incorporate P2's BSR strategies into the transition system in Def. 27 to reduce P1's planning problem into an MDP with a reachability objective, stated next.

Definition 30. Under Assumption 8, the dynamic hypergame $\mathcal{H} = \langle V, A, \Delta, (s_0, h_0, q_0, x_0), \mathcal{F} \rangle$ reduces to a finite-state MDP with a reachability objective for P1,

$$\widetilde{\mathcal{H}} = \langle \widetilde{V}, A_1, \widetilde{\Delta}, (s_0, [\![h_0]\!]_{x_0}, q_0, x_0), \widetilde{\mathcal{F}} \rangle,$$

where

- $\widetilde{V}$ is a finite and discrete set of states. Each state $\widetilde{v} = (s, [\![h]\!]_x, q, x)$ consists of a state s, an inference-equivalent class given the (η, x)-equivalent relation, a state q in the DFA, and a hypothesis x of P2.

- $\widetilde{\Delta}$: $\widetilde{V} \times A_1 \to \mathcal{D}(\widetilde{V})$ is defined as follows: For any state $\widetilde{v} = (s, [\![h]\!]_x, q, x)$, if $q = q_{\mathsf{sink}}$ — the sink state in the DFA $\mathcal{A}$, then state $\widetilde{v}$ is a sink state.

  Given $\widetilde{v}_1 = (s_1, [\![h_1]\!]_{x_1}, q_1, x_1)$ with $q_1 \neq q_{\mathsf{sink}}$, $a^1 \in A_1$, and $\widetilde{v}_2 = (s_2, [\![h_2]\!]_{x_2}, q_2, x_2)$ and $h_1(a^1, a^2)s_2 \in [\![h_2]\!]_{x_2}$, then

$$\widetilde{\Delta}(\widetilde{v}_2 \mid \widetilde{v}_1, a^1) = \sum_{a^2 \in A_2} \pi_2^{*,x_1,2}(a^2 \mid s_1) \cdot P(s_2 \mid s_1, (a^1, a^2)) \mathbf{1}(\delta(q_1, L(s_2)) = q_2).$$

where $\pi_2^{*,x_1,2}(a^2 \mid s_1)$ is the probability of P2 selecting action a^2 given its current hypothesis x_1 and the current state s_1. That is, P2 uses a BSR strategy.

- $(s_0, [\![h_0]\!]_{x_0}, q_0, x_0) \in \widetilde{V}$ is the initial state, given (s_0, h_0, q_0, x_0) is the initial state in the transition system $\mathcal{H}$.
- $\widetilde{\mathcal{F}} = \{(s, [\![h]\!]_x, q, x) \in \widetilde{V} \mid q \in F\}$ is the set of final states for P1, where F is the set of final states of DFA $\mathcal{A}$. P1's goal is to maximize the probability of reaching $\widetilde{\mathcal{F}}$.

By construction, if a path ρ in the MDP visits $\widetilde{\mathcal{F}}$, then P1 satisfies the scLTL formula φ_1. Thus, maximizing the probability of satisfying P1's specification is equivalent to maximizing the probability of reaching the set $\widetilde{\mathcal{F}}$. The optimal policy for P1 in $\widetilde{\mathcal{H}}$ is deceptive because by optimally planning in this MDP, P1 will select actions to influence P2's belief so that P2 takes actions that are advantageous for P1 to achieve its goal. We can employ dynamic programming to solve the optimal value function $\mathcal{V} : \widetilde{\mathcal{V}} \to \mathbb{R}$ which satisfies the Bellman optimality condition:

$$\mathcal{V}(\widetilde{v}) = \max_{a \in A_1} \sum_{\widetilde{v}' \in \widetilde{V}} \widetilde{\Delta}(\widetilde{v}' \mid \widetilde{v}, a) \mathcal{V}(\widetilde{v}'), \forall \widetilde{v} \notin \widetilde{\mathcal{F}}, \tag{5-3}$$

and

$$\mathcal{V}(\widetilde{v}) = 1, \forall \widetilde{v} \in \widetilde{\mathcal{F}}.$$

where $\{\mathcal{V}(\widetilde{v}) \mid \widetilde{v} \in \widetilde{V}\}$ is the set of decision variables. The optimal policy $\widetilde{\pi}_1^*$ is computed from the optimal value function:

$$\widetilde{\pi}_1^*(\widetilde{v}) = \arg\max_{a \in A_1} \sum_{\widetilde{v}' \in \widetilde{V}} \widetilde{\Delta}(\widetilde{v}' \mid \widetilde{v}, a) \mathcal{V}(\widetilde{v}'), \forall \widetilde{v} \notin \widetilde{\mathcal{F}}.$$

The time complexity for solving MDPs with reachability objectives is polynomial in the size of state space and action space. Here, the size of state space in the MDP is $O(|S| \times N \times |Q| \times |X|)$, where N is the number of (η, x)-equivalent classes of histories in the game. The size of the action space in the MDP is $|A_1|$. Besides using dynamic programming, an MDP with a reachability

objective can be solved using probabilistic model checking algorithms ([78, Chapter 10.1.1], [94]) and existing PRISM toolbox [95].

Remark 2. Given the problem can be of large scale, approximate dynamic programming (ADP) solutions of MDP can be used to reduce the number of decision variables [96]. For example, value function approximation in ADP uses a function approximator (such as a neural network) to approximate the value function, where the decision variables are coefficients of the value function. In the problem of large scale, it is often the case that the number of coefficients of the value function approximator is much smaller than the number of states.

To this end, we include Alg. 5-1 to describe how to compute P1's subjectively rationalizable strategy in the dynamic hypergames with temporal logic objectives.

Theorem 5-3. *Assuming P1's knowledge about η is correct, the optimal strategy $\widetilde{\pi}_1^*: \widetilde{V} \to \mathcal{D}A_1$ in the MDP $\widetilde{\mathcal{H}}$ is P1's subjectively rationalizable strategy in the dynamic hypergame given P2's evolving knowledge.*

Proof. The construction of $\widetilde{\mathcal{H}}$ is achieved through marginalizing out P2's actions given that P2 follows the BSR strategy in the dynamic hypergame $\mathcal{H}$. Thus, optimal planning in $\widetilde{\mathcal{H}}$ computes the best response strategy for P1 against P2's BSR strategy. Any deviation from this best response strategy will not gain P1 a better outcome. □

Remark 3. Assumption 8-4) is not necessary. If P2's SR strategy $\pi_2^{*,x,2}$ is not memoryless in the game $G(x)$ but represented using a finite-state controller (also known as a finite-memory policy), then we can augment the states in the hypergame transition system in Def. 27 with the states in the finite-state controller and planning in the augmented state space.

Definition 31 (Value of Deceit)**.** Given the dynamic hypergame $\mathcal{H} = \langle V, A, \Delta, (s_0, h_0, q_0, x_0), \mathcal{F} \rangle$, the value of deceit is defined by

$$\mathsf{VoD} = \frac{\Pr^{\widetilde{\mathcal{H}}, \widetilde{\pi}_1^*}(s_0 h' \models \varphi_1)}{u_1(s_0, \pi_1^*, \pi_2^*, \varphi_1)},$$

Algorithm 5-1 Computation of P1's subjectively rationalizable strategy.

1: Construct P1's level-1 hypergame $\mathcal{H}$ with TS, $\mathcal{A}$, X, and η.
2: **for** $x \in X$ **do**
3: Compute P2's SR strategy $\pi_2^{*,x,2}$ from game $G(x)$.
4: **end for**
5: Construct $\widetilde{\mathcal{H}}$ with $\{\pi_2^{*,x,2} \mid x \in X\}$ and $\mathcal{H}$.
6: $\widetilde{\pi}_1^*, \mathcal{V} \leftarrow$ Solve MDP $\widetilde{\mathcal{H}}$.
7: **return** $\widetilde{\pi}_1^*$.

where $\Pr^{\widetilde{\mathcal{H}},\widetilde{\pi}_1^*}(s_0 h' \models \varphi_1)$ is the probability of satisfying the given P1's task φ_1 in the Markov chain induced from $\widetilde{\mathcal{H}}$ under the optimal policy $\widetilde{\pi}_1^*$, and $u_1(s_0, \pi_1^*, \pi_2^*, \varphi_1)$ is the value of the zero-sum game with complete information given P1's task φ_1.

Note that we have $\Pr^{\widetilde{\mathcal{H}},\widetilde{\pi}_1^*}(s_0 h' \models \varphi_1) = \mathcal{V}(s_0, [\![h_0]\!]_{x_0}, q_0, x_0)$. In words, the value of deceit is the ratio between P1's probability of satisfying the scLTL objective using the solution of the dynamic hypergame and P1's probability of satisfying the same objective when both players have complete information. Based on the definition, P1 will only gain advantage with deception when the value of deceit is greater than one.

5.2.4 Case study: Robot Motion Planning

In this section, we present a robot motion planning example to illustrate the proposed deceptive planning method. This case study includes an inference function for P2 based on the sliding-window change detection, introduced next.

5.2.4.1 Inference with Sliding-Window Change Detection

We introduce a class of inference algorithms based on change detection in Markov chain (MC) [97]. Given P2's finite hypothesis space X, P2 can construct a set of games $\{G(x) \mid x \in X\}$. For each game $G(x)$, it is assumed that there is a unique equilibrium $\langle \pi_1^{*,x}, \pi_2^{*,x} \rangle$, where $\pi_i^{*,x}$: $\mathsf{PrefPaths} \to \mathcal{D}A_i$ is a mixed strategy for player i given the hypothesis x. This equilibrium induces a probability measure $\Pr^{\langle \pi_1^{*,x}, \pi_2^{*,x} \rangle}$ over histories in $G(x)$. For simplicity in notation, we denote $\Pr^{\langle \pi_1^{*,x}, \pi_2^{*,x} \rangle}$ as $\Pr^x$.

When P2's current hypothesis is x, P2 can detect a change from x to some $x' \in X$ using a sliding-window change detection algorithm based on the Cumulative SUM (CUSUM)

statistic [98]. First, we are given a data point in forms of history $h = s_0 a_0 s_1 \dots s_n$ and a nominal model x_0. We denote the interval of a time window of size $m+1$ as $[k, k+m]$, and the history within this time window is $s_k a_k \dots s_{k+m} a_{k+m} s_{k+m+1}$. Second, we denote the i-th observation of the transitions *within the time window* as $y_i = (a_{k+i-1}, s_{k+i})$ for $1 \leq i \leq m+1$. When $i = 0$, the 0-th observation within the window is $y_0 = (s_k)$. Intuitively, given a data and a nominal model x_0, the sliding-window change detection algorithm uses a subsequence of history over a time window and detects if a change has occurred in the model that generates the data during this time window. Specifically, for each hypothesis $x \in X$ and a nominal model x_0, the algorithm computes the log-likelihood ratio, for $1 \leq j \leq m+1$,

$$R_j^x = \sum_{i=0}^{j} r_i^x,$$

where $r_i^x = \ln \frac{\Pr^x(y_i)}{\Pr^{x_0}(y_i)}$, and $\Pr^x(y_i)$ (resp. $\Pr^{x_0}(y_i)$) is the probability of observing the transition given the probability measure $\Pr^x$ (resp. $\Pr^{x_0}$).

The change detection lies in the difference between the log-likelihood ratio and its current minimum value. The CUSUM score is given by,

$$Z_l^x = R_l^x - \min_{1 \leq j \leq l} R_j^x, \text{ for } 1 \leq l \leq m+1.$$

Recursively, the CUSUM score is updated for each hypothesis $x \in X$ as

$$Z_l^x = \max\{0, Z_{l-1}^x + \ln \frac{\Pr^x(y_l)}{\Pr^{x_0}(y_l)}\}, \tag{5-4}$$

where $Z_0^x = 0$.

A change is detected at time t when the score of at least one model, say Z_l^x, exceeds a user-defined constant threshold $c > 0$. Formally, the *time of change* is given by

$$t = \min\{l \mid \exists x \in X,\ Z_l^x \geq c\}.$$

Once a change is detected, the algorithm sets the nominal model to be the current predicted model, disregards the history until the change, and keeps running the online change detection given new observations from the change point onwards. In the case when multiple models maintain similar CUSUM scores, we select one model based on some domain-specific heuristics or at uniformly random.

Lemma 5-1. *Given a sliding-window change detection inference* $\eta : X \times \mathsf{PrefPaths} \to X$ *with window size* $m+1$ *and a finite hypothesis space* X*, two histories* h_1, h_2 *are* (η, x)-equivalent *if they share the same suffix* [3] *of length* $m+1$.

Proof. The proof is based on the property of the change detection and thus omitted. □

5.2.4.2 Deceptive Planning with a Temporal Logic Objective

We consider two examples inspired by security games, referred to $world_1$ (Fig. (5-4a)) and $world_2$ (Fig. (5-4b)). In both worlds, a robot (P1) is to visit several regions of interest (labeled A, B, C and colored in red) according to a temporal ordering, and an observer (P2) can reallocate traps in cells colored in blue. Both games are concurrent: When P1 selects an action to move, P2 simultaneously chooses an action to reallocate the traps. When P1 enters the cell where P2 allocates the trap to that cell, we say that P1 is trapped. The game terminates in two ways: a) P1 is trapped; b) P1 completes its task.

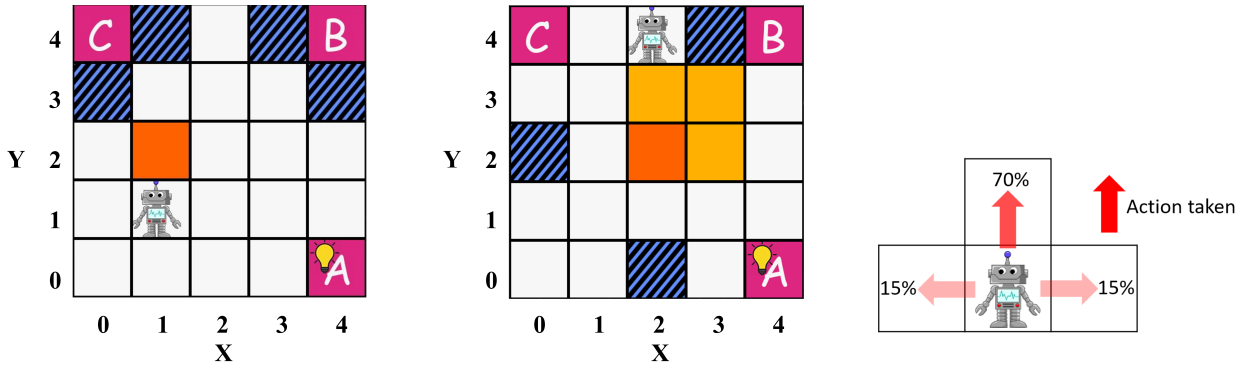
Formally, we describe P1's task by the formula as follows:

$$\varphi_1 = (\neg \text{obs} \,\mathsf{U}\, A) \wedge (\neg (B \vee \text{obs}) \,\mathsf{U}\, C).$$

That is, the robot needs to visit A and C without reaching obstacles. Before visiting C, the robot cannot visit B. The corresponding DFA is drawn in Fig. (5-5).

P1 can move in four compass directions, and P1's dynamics is plotted in Fig. (5-4c). The grid world is surrounded by a bouncing wall, *i.e.*, if P1 hits the wall, then P1 gets bounced back to P1's previous cell. The orange cell in the grid world is a static obstacle, labeled by obs.

3 For a word $w = \sigma_1 \sigma_2 \ldots \sigma_n$, a suffix of w is a word v of the form $\sigma_i \sigma_{i+1} \ldots \sigma_n$, where $1 \leq i \leq n$.



(a) (b) (c)

Figure 5-4. (a): $world_1$'s initial configuration for P1 and P2. (b): $world_2$'s initial configuration for P1 and P2. Cells colored in yellow are walls. Bulbs indicate initial P2's predictions. (c): Robot's dynamics when action "up" is taken.

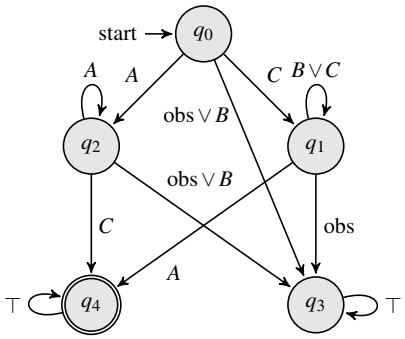


Figure 5-5. The task automaton with 5 states and 12 edges corresponds to φ_1, where $Q = \{q_i \mid i = 0,1,2,3,4\}$.

P2 can reallocate the traps (*i.e.*, dynamic obstacles) to a subset of cells colored in blue in $world_1$ and $world_2$. P2 can only use ℓ traps with n possible trap locations. Thus, the number of actions for P2 is $\binom{n}{\ell}$, *i.e.*, choose ℓ out of n. Every time P2 resets the location of any trap, it must wait at least k time steps to be able to reallocate any trap again. In the example of $world_1$, we select $n = 4$, $\ell = 1$, and let $k = 0$; In the example of $world_2$, we select $n = 3$, $\ell = 1$, and let k to be a variable.

In both examples: $world_1$ and $world_2$, the asymmetrical information is as follows:

- P1 knows the complete task φ_1.
- P2 does not know the complete task φ_1.

We refer to this situation as *asymmetric information* case. On the other side, if P2 knows P1's complete task, then we refer to that as *symmetric information* case. In the *asymmetric information* case, P2 has a hypothesis space $X = \{\neg\text{obs}\,\mathsf{U}\,\phi \mid \phi \in \{A,B,C\}\}$.

Different behaviors under asymmetric and symmetric information cases in $world_1$. We compare P1's task completion rates between asymmetric information case and symmetric information case.

In the asymmetric information case, for each $x \in X$, P2 solves a Stackelberg/leader-follower game and decides a trap configuration against the best response of P1 in game $G(x)$. Let A_2 be the set of different configurations of traps. The strategy of P2 is obtained as follows:

$$\pi_2^{*,x,2}(s) = \arg\min_{a^2 \in A_2} \max_{\pi_1} \Pr^{\pi_1}(hh' \models x \mid s, a^2),$$
$$\forall s \in S,$$

where $\Pr^{\pi_1}(hh' \models x \mid a^2)$ is the probability of P1 satisfying the formula x given P2's action (trap configuration) a^2. For instance, if $x = \neg\text{obs}\,\mathsf{U}\,B$ and robot is at the $(2,4)$, then P2's optimal action is to allocate the trap to the blue cell right to robot, that is $(3,4)$. For each hypothesis $x \in X$ and state $s \in S$, P2 solves the optimal trap allocation action a^2 and also computes the best response of P1 that achieves the maximum probability of satisfying x from the state s. The joint strategy profiles for different hypotheses $x \in X$ also enable P2 to infer the subgoal of P1: P2 observes the behavior of P1 given the current trap location a^2 and then infer, for which x, P1's behavior matches with the best response given x and a^2 using the sliding-window change detection.

For the configuration of $world_1$, we evaluate different window sizes and find sliding-window size $m+1 = 2$ and user-defined threshold $c = 0.12$ achieve a good trade-off between space complexity and accuracy in prediction in this example. In the symmetric information case, P1 and P2 both have exact knowledge of task specification φ_1, and P1 wants to maximize the P1's probability of finishing the task; P2 wants to minimize the P1's probability of

finishing the task. We denote the Nash Equilibrium strategy profile by $\langle \pi_1^*, \pi_2^* \rangle$, where the Nash Equilibrium strategy profile is obtained as follows:

$$\langle \pi_1^*, \pi_2^* \rangle = \arg \min_{\pi_2 \in \Pi_2} \max_{\pi_1 \in \Pi_1} \overset{\langle \pi_1, \pi_2 \rangle}{\Pr} (hh' \models \varphi_1).$$

Table 5-3. The completion rates for P1 in asymmetric information case and symmetric information case in $world_1$.

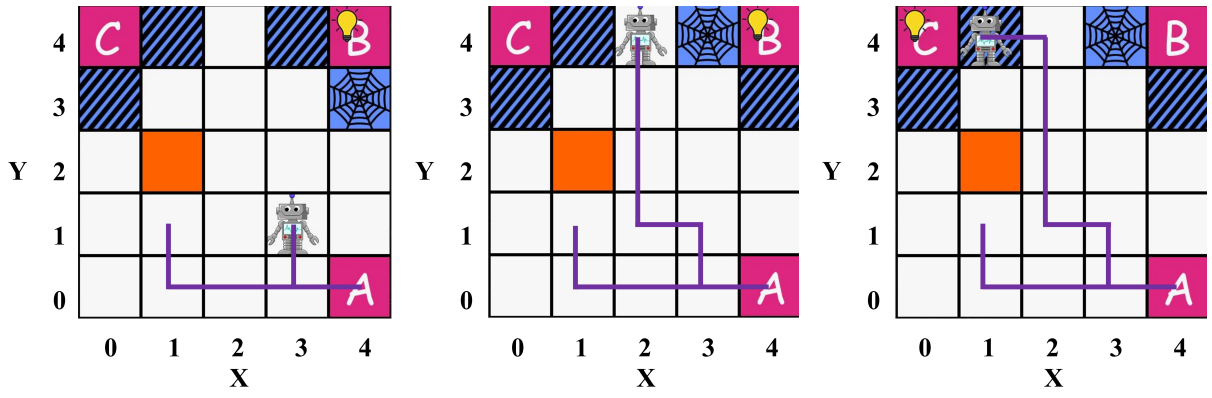
Info	P1 Policy	P2 Policy	Completion rate (P1)
Asymmetric	$\widetilde{\pi}_1^*$	$\pi_2^{B,2}$	66.96%
Symmetric	π_1^*	π_2^*	29.69%

In Table. 5-3, we list P1's completion rates for its task specification: one for asymmetric information case and one for symmetric information case. From Table. 5-3, it indicates that under asymmetrical information, by following the deceptive strategy given P2 plays BSR strategy, P1 has a higher probability of satisfying the specification than that of the case by following the Nash Equilibrium strategy profile. The *value of deceit* in $world_1$ is $\mathsf{VoD} = \frac{66.96\%}{29.69\%} = 2.26$.

Note that in this case, P2 can only place traps near B and C but not A. We plot three key steps during the simulation in Fig. (5-6). The solid lines denote the robot's trajectories. In Fig. (5-6) (a), P2 predicts that P1 is to reach B after observing that the robot goes up. The prediction does not change until the robot reaches $(1,4)$ in Fig. (5-6) (c). When the robot reaches $(2,4)$, P2 still predicts B (see Fig.5-6 (b)) and places the trap at $(3,4)$ (see Fig.5-6 (c)). When the robot reaches $(1,4)$, P2 correctly predicts C. But it is too late, and P2 cannot prevent the robot from reaching C. The deceptive strategy leverages this information asymmetry to lead P1 to achieve a higher probability of finishing its task. We provide a short video [4] to demonstrate the difference between P2's behaviors in the cases with asymmetric information and symmetric information, respectively.

Next, we investigate how delays in reallocating traps for P2 would affect the completion rate of P1. However, in the $world_1$ example, we observed in experiments that any delay on

4 https://www.dropbox.com/s/i98ka56gdhdvxgq/video_10_09_2021.mp4?dl=0



(a) (b) (c)

Figure 5-6. Three key steps of deception in the simulation. (a) P2 predicts P1 is to reach B. (b) P2 reallocates the trap given P1's position. (c) P2 predicts that P1 is to reach C but it is too late for P2 to respond.

reallocation could easily lead P1 to complete its task. Based on this observation, we construct another example $world_2$, and evaluate the completion rates for every k steps of delay and effectiveness of model mismatch in this example $world_2$.

Reallocation every k steps of delay in $world_2$. In this example, we assume that P2 is restricted to only reallocate the trap after k steps since the last reallocation, where k is an integer. P1 is aware of P2's delay k and synthesizes the deceptive strategy. Fig. (5-7) shows the completion rate of task (values of P1 at initial state $(2,4)$ in Fig. (5-4b)) under different steps of delay up to $k = 3$. The results indicate that with the increase of steps of delay, the probability of completing the task increases, and P1 exploits P2's delay and lack of information.

Detection of model mismatch in $world_2$. We use experiments in the configuration $world_2$ to demonstrate the effectiveness of the detection mechanism, that is, to identify whether there is a deviation from the predicted opponent model of P2. We set the significance level $\alpha = 0.05$. If the likelihood of observed action sequences is smaller than or equal to 0.05, we reject the null hypothesis: the data is generated by our predicted model of P2.

We consider a case that P1 follows policy $\widetilde{\pi}_1^*$, and P2 plays the policy predicted by P1 for the first four steps. After the first four steps, we let P2 play a random policy π_2^R, *i.e.*,

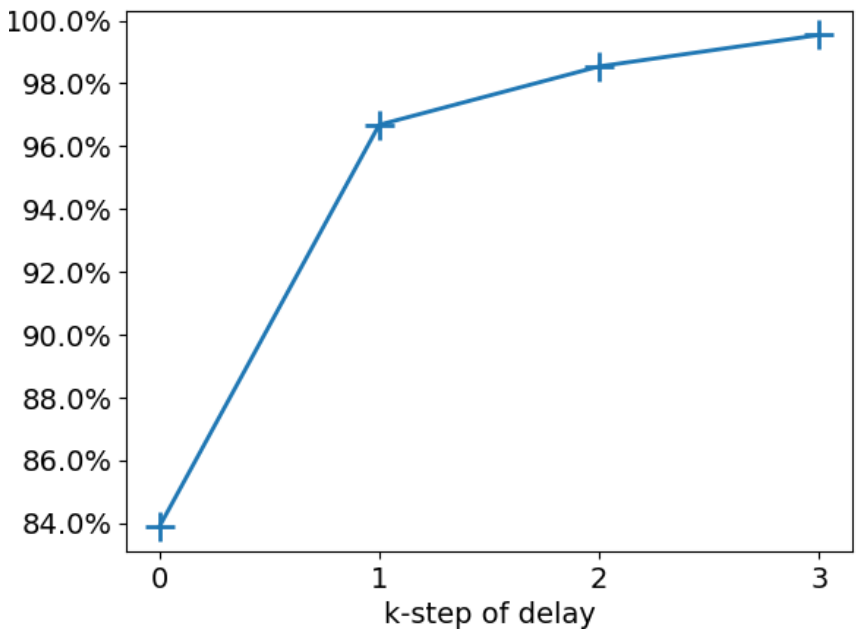


Figure 5-7. The task completion rates of P1 given P2 with k-step delay in reallocating traps, for $k = 0, 1, 2, 3$.

$\pi_2^R(a \mid s) = \frac{1}{|A(s)|}$, for all $a \in A(s)$. The mismatch is detected at the 7-th step of the online interaction, and P1 is alerted that P2 deviates from the predicted policy. We compute λ after each step and plot it in Fig. (5-8), where we also plot the χ^2. (The reason predicted $\lambda = 0$ is because the predicted policy $\pi_2^{B,2}$ is deterministic.) From Fig. (5-8), we see that at the 7-th step of online interaction, we have $\lambda > \chi^2$, so we reject the null hypothesis H_0. The degree of freedom in the Chi-square detector is the number of the state-action pairs.

Complexity. Our realization of the proposed framework in examples includes three major components: a) Inference with sliding-window change detection, b) Equilibrium solving of Stackelberg games, c) MDP planning for deceptive planning. The inference with sliding-window change detection has an $O(m)$ time complexity, where $m+1$ is the window size. It is noted that P2's BSR strategies are computed using a set of leader policies computed offline based on solving a set of Stackelberg games, one for each hypothesis. Given P2's BSR policy, we can reduce solving P1's optimal deceptive strategy problem into an MDP planning problem, which can be

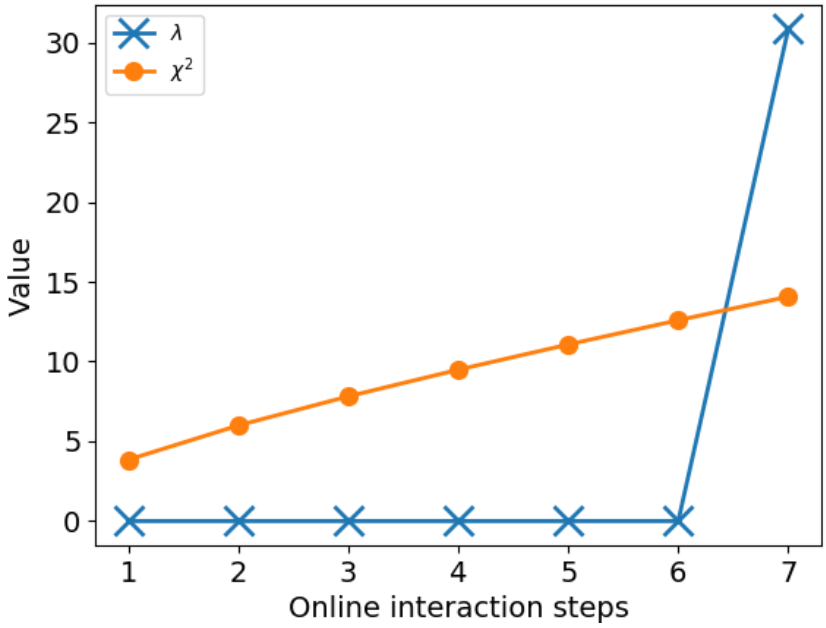


Figure 5-8. The likelihood ratio λ for online interaction between P1 and P2.

solved in polynomial time in the size of the states and actions [99], where the state space is the product of the states in the game, the set of inference-equivalent histories, the DFA states, and a set of hypotheses. We solve the equilibrium of Stackelberg games and solve the MDP with the value iteration algorithm. We run algorithms on a Windows 10 machine with AMD Ryzen 9 5900X CPU and 16 GB RAM. The computational time of equilibrium solving of Stackelberg games are about 5 s, and the computational time of MDP planning is 140 s.

Finally, it is remarked that the deceptive planner can use different components given different inference algorithms and solutions of P2's BSR strategies. This analysis of complexity may not generalize to other classes of hypergames.

CHAPTER 6
PLANNING WITH INCOMPLETE PREFERENCES OVER TEMPORAL GOALS

This chapter investigates the problem of planning with incomplete preferences over temporal goals. We introduce a novel automata-theoretic approach to qualitative planning in MDPs with incomplete preferences over temporal logic objectives. Our approach consists of a language `PrefScLTL` to specify preferences over ω-regular reachability objectives, a procedure to construct an automaton representation of the preference model defined by the `PrefScLTL` formula, and a synthesis algorithm to construct a maximal preference satisfying strategy.

6.1 `PrefScLTL`: A Language to Specify Preferences over Temporal Objectives

In this section, we introduce a new language to express preferences over scLTL formulas.

Definition 32 (Preference formula)**.** Let φ be an scLTL formula. A preference formula is defined inductively as follows.

$$\alpha := \varphi \triangleright \varphi \mid \varphi \approx \varphi \mid \varphi \bowtie \varphi \mid \alpha \wedge \alpha.$$

Given two scLTL formulas φ_1 and φ_2, the formula $\varphi_1 \triangleright \varphi_2$ represents that satisfying φ_1 is *strictly preferred* to satisfying φ_2. The formula $\varphi_1 \approx \varphi_2$ represents that satisfying φ_1 is *indifferent* to satisfying φ_2. The formula $\varphi_1 \bowtie \varphi_2$ represents that satisfying φ_1 is *incomparable* to satisfying φ_2. The formula $\alpha_1 \wedge \alpha_2$ represents that both the preference formulas α_1 *and* α_2 should be satisfied.

The formula $\varphi_1 \trianglerighteq \varphi_2$ represents that satisfying φ_1 is *weakly preferred* to satisfying φ_2. The operator $\trianglerighteq$ is a derived operator. Given a formula α that contains weak preference operator, a preference formula α' containing only $\triangleright, \approx, \bowtie, \wedge$ can be constructed based on [100, Ch. 2]: If $\varphi_1 \trianglerighteq \varphi_2$ appears in α but $\varphi_2 \trianglerighteq \varphi_1$ does not, then α' contains $\varphi_1 \triangleright \varphi_2$. If $\varphi_2 \trianglerighteq \varphi_1$ appears in α but $\varphi_1 \trianglerighteq \varphi_2$ does not, then α' contains $\varphi_2 \triangleright \varphi_1$. If $\varphi_1 \trianglerighteq \varphi_2$ and $\varphi_2 \trianglerighteq \varphi_1$ appear in α, then α' contains $\varphi_1 \approx \varphi_2$. If neither $\varphi_1 \trianglerighteq \varphi_2$ not $\varphi_2 \trianglerighteq \varphi_1$ appears in α, then α' contains $\varphi_1 \bowtie \varphi_2$.

The formulas $\varphi_1 \triangleright \varphi_2$, $\varphi_1 \trianglerighteq \varphi_2$, $\varphi_1 \approx \varphi_2$, and $\varphi_1 \bowtie \varphi_2$ are called *atomic* preference formulas. The formulas containing $\wedge$-operator are called general preference formulas.

A preference formula is interpreted using the preference model they induce over the set Σ^ω (formalized in Definition 36. The preference model determines, for a pair $w, w' \in \Sigma^\omega$ of words, whether w is preferred/indifferent/incomparable to w'.

Definition 33 (Preference Model)**.** A preference model is a tuple $\mathcal{P} = \langle U, \succeq \rangle$, where U is a countable set of outcomes and $\succeq$ is a reflexive and transitive binary relation, *i.e.*, a partial order, on U.

We recall that a binary relation $\succeq$ on U is reflexive if every element $u \in U$ is related to itself, *i.e.*, $u \succeq u$. It is transitive if $u_1 \succeq u_2$ and $u_2 \succeq u_3$ then $u_1 \succeq u_3$ is true for any $u_1, u_2, u_3 \in U$. When $\succeq$ is a partial order, $u_1 \succeq u_2$ and $u_2 \succeq u_1$ implies $u_1 \approx u_2$. An antisymmetric partial order, in which $u_1 \succeq u_2$ and $u_2 \succeq u_1$ implies $u_1 = u_2$, is called a preorder.

The preference model over Σ^ω induced by a preference formula α is understood based on the preference model induced by α over the set of scLTL formulas appearing in α. The following definition describes how to construct the preference model from a preference formula.

Definition 34. The preference model induced by α over the set of scLTL formula appearing in φ is the tuple

$$\mathcal{P} = \langle \mathbb{F}, \trianglerighteq \rangle,$$

where

- $\mathbb{F} = \{\varphi_0, \varphi_1, \ldots, \varphi_n\}$ where $\varphi_1, \ldots, \varphi_n$ is the set of scLTL formula appearing in α and $\varphi_0 = \bigwedge_{i=1\ldots n} \neg\varphi_i$. φ_0 is not included if $\varphi_0 = \bot$;
- $\trianglerighteq$ is the transitive closure of the set $\{(\varphi_i, \varphi_j) \mid 0 < i, j \leq n : \varphi_i \trianglerighteq \varphi_j$ or $\varphi_i \triangleright \varphi_j$ or $\varphi_i \approx \varphi_j$ appears in $\alpha\} \cup \{(\varphi_i, \varphi_0) \mid 0 < i, j \leq n\} \cup \{(\varphi_i, \varphi_i) \mid i = 0 \leq i, j \leq n\}$.

The set $\mathbb{F}$ containing φ_0 is said to be the *completion* of the set of scLTL formulas $\{\varphi_1, \ldots, \varphi_n\}$ appearing in α since it ensures that, for every word $w \in \Sigma^\omega$, there exists a formula $\varphi_i \in \mathbb{F}$, $i = 0 \ldots n$, such that $w \models \alpha$. It is also noted that, by construction, $\trianglerighteq$ is a partial order.

Remark 4. In Definition 34, we follow the common assumption [55] that satisfying some outcome in $\mathbb{F}$ is strictly preferred to satisfying none of them, *i.e.*, $\varphi_i \triangleright \varphi_0$ for all $i = 1 \dots n$.

Combinative preferences. The model $\langle \mathbb{F}, \trianglerighteq \rangle$ is a combinative preference model, as opposed to an exclusionary one. This is because we do not assert the exclusivity condition that the languages of any two formulas φ_1, φ_2 in $\mathbb{F}$ have empty intersection. This allows us to represent a preference such as $(\Diamond a \wedge \Diamond b) \trianglerighteq \Diamond a$, *i.e.*, "Visiting A and B is preferred to visiting A," where the less preferred outcome must be satisfied first in order to satisfy the more preferred outcome. In literature, it is common to study exclusionary preference models (see [54, 55] and the references within) because of their simplicity [65]. However, we focus on planning with combinative preferences since they are more expressive than the exclusionary ones [101]. In fact, every exclusionary preference model can be transformed into a combinative one, but the opposite is not true.

When a combinative preference model is interpreted over infinite plays, the agent needs a way to compare the subsets of formulas in $\mathbb{F}$ satisfied by two plays. For instance, consider the preference formula $(\Diamond a \wedge \Diamond b) \trianglerighteq \Diamond a$. Let ρ_1, ρ_2 be two plays. Suppose that ρ_1 visits both A and B, and ρ_2 visits A only. Therefore, $\rho_1 \models \Diamond a \wedge \Diamond b$, whereas $\rho_2 \models \Diamond a$. To determine the preference between the two plays, the agent compares the set $\{\Diamond a, \Diamond b\}$ with $\{\Diamond a\}$ to conclude that the ρ_1 is preferred over ρ_2. However, suppose the given preference formula is $\Diamond a \trianglerighteq \Diamond b$. Then, the two plays would be indifferent since both satisfy the more preferred objective of visiting A. In this case, the less preferred objective of visiting B would not influence the comparison of the sets. To formalize this notion, we define the notion of most-preferred outcomes.

Given a non-empty subset $\mathbb{X} \subseteq \mathbb{F}$, let $\mathsf{MP}(\mathbb{X}) \triangleq \{R \in \mathbb{X} \mid \nexists R' \in \mathbb{X} : R' \triangleright R\}$ denote the set of most-preferred outcomes in $\mathbb{X}$.

Definition 35. Given a preference model $\langle \mathbb{F}, \trianglerighteq \rangle$ and a word $w \in \Sigma^\omega$, the set of most-preferred outcomes satisfied by w is given by $\mathsf{MP}(w) \triangleq \mathsf{MP}(\{\varphi \in \mathbb{F} \mid w \models \varphi\})$.

By definition, there is no outcome included in $\mathsf{MP}(w)$ that is preferred to any other outcome

in $\mathsf{MP}(w)$. Thus, we have the following result.

Lemma 6-1. *For any word $w \in \mathsf{Plays}(M)$, every pair of outcomes in $\mathsf{MP}(w)$ are incomparable to each other.*

Now, we formally define the interpretation of $\langle \mathbb{F}, \trianglerighteq \rangle$ in terms of the preference relation it induces on Σ^ω.

Definition 36 (Semantics)**.** Given a preference formula φ, let $\langle \Sigma^\omega, \succeq \rangle$ be the preference model induced by φ over Σ^ω. Then, for any $w_1, w_2 \in \Sigma^\omega$, we have

- $w_1 \succ w_2$, *i.e.*, w_1 is strictly preferred to w_2, if and only if there exist a pair of outcomes $\alpha \in \mathsf{MP}(w_1)$ and $\alpha' \in \mathsf{MP}(w_2)$ such that $\alpha \triangleright \alpha'$, and there does not exist a pair of outcomes $\alpha \in \mathsf{MP}(w_1)$ and $\alpha' \in \mathsf{MP}(w_2)$ such that $\alpha' \triangleright \alpha$.
- $w_1 \sim w_2$, *i.e.*, w_1 is indifferent to w_2, if and only if $\mathsf{MP}(w_1) = \mathsf{MP}(w_2)$.
- $w_1 \nparallel w_2$, , *i.e.*, w_1 is incomparable to w_2, otherwise.

6.2 Preference Automaton

In this section, we introduce a novel computational model called a Deterministic Finite-state Preference Automaton (DFPA), which encodes the preference model $\langle \Sigma^\omega, \succeq \rangle$ into an automaton. We present a procedure to construct a Deterministic Finite-State Preference Automaton (DFPA) given a preference model $\mathcal{P} = \langle \mathbb{F}, \trianglerighteq \rangle$ and prove its correctness with respect to the interpretation in Definition 36.

Definition 37. A deterministic finite-state preference automaton (DFPA) is a tuple,

$$\mathcal{B} = \langle Q, \Sigma, \delta, \iota, G \rangle,$$

where Q, Σ, δ, ι are the finite set of states, the alphabet, the deterministic transition function, and an initial state, similar to these components in a DFA. The last component $G = (\mathcal{X}, E)$ is called a preference graph, where the set of nodes $\mathcal{X} \subseteq 2^Q$ represents a partition of Q and $E \subseteq \mathcal{X} \times \mathcal{X}$ is a set of directed edges.

Algorithm 6-1 Construction of preference graph

function PREFGRAPH($\langle \mathbb{F}, \trianglerighteq \rangle, \langle Q, \Sigma, \delta, \iota \rangle$)
  Initialize $\mathcal{X} = \emptyset, E = \emptyset$.
  Let $\Lambda \leftarrow \{\mathsf{Maximal}(\vec{q}) \mid \vec{q} \in Q\}$.
  $\mathcal{X} \leftarrow \{\{\vec{q} \in Q \mid \mathsf{Maximal}(\vec{q}) = \lambda\} \mid \lambda \in \Lambda\}$ is the set of nodes of preference graph.
  for all $(X, X') \in \mathcal{X} \times \mathcal{X}$ **do**
    Let $\vec{q}, \vec{q}'$ be two arbitrary states in X, X', respectively.
    Initialize $\texttt{Cond}_{1a} \leftarrow \texttt{false}$ and $\texttt{Cond}_{1b} \leftarrow \texttt{true}$.
    for all $(\alpha, \alpha') \in \mathsf{Maximal}(\vec{q}) \times \mathsf{Maximal}(\vec{q}')$ **do**
      if $\alpha \triangleright \alpha'$ **then**
        $\texttt{Cond}_{1a} \leftarrow \texttt{true}$.
      end if
      if $\alpha' \triangleright \alpha$ **then**
        $\texttt{Cond}_{1b} \leftarrow \texttt{false}$.
      end if
    end for
    if $\texttt{Cond}_{1a} = \texttt{true} \wedge \texttt{Cond}_{1b} = \texttt{true}$ **then**
      Add (X', X) to the set of edges E.
    end if
  end for
  return $G = \langle \mathcal{X}, E \rangle$
end function

Given a word $w = \sigma_0 \sigma_1 \ldots \in \Sigma^\omega$, the *path* induced by w in the DFPA is the sequence of states $q_0 q_1 \ldots \in Q^\omega$ such that $q_0 = \iota$ and for any integer $k \geq 0$, we have $q_{k+1} = \delta(q_k, \sigma_k)$. The preference graph G defines a preference model over Q as follows: Each preference node $X \in \mathcal{X}$ represents an equivalence class of states in Q such that any two states $q, q' \in X$ are indifferent to one another. Each edge $(X, X') \in E$ represents a strict preference that any state in X' is strictly preferred to any state in X and an absence of an edge between two nodes $X, X' \in \mathcal{X}$ represents that any state in X is incomparable to any state in X'.

Next, we describe the construction of DFPA given a preference model $\mathcal{P} = \langle \mathbb{F}, \trianglerighteq \rangle$ induced by φ. The construction involves two steps, namely, the construction of the underlying graph of DFPA and the construction of the preference graph.

Definition 38. Let $\mathcal{A}_i = \langle Q_i, \Sigma, \delta_i, \iota_i, F_i \rangle$ be the *complete* DFA representing the languages of α_i for

all $i = 0 \ldots n$. The underlying graph of the DFPA representing $\mathcal{P}$ is the tuple,

$$\langle Q, \Sigma, \delta, \iota \rangle$$

where $Q = \times_{i=0}^{n} Q_i$ is the set of states in DFPA. We represent each state in Q as a vector $\vec{q}$ and the i-th component of $\vec{q}$, denoted as $\vec{q}[i]$, is the state in Q_i. $\Sigma = \wp(\mathcal{AP})$ is a set of symbols. $\delta : Q \times \Sigma \to Q$ is the transition function is defined as $\delta(\vec{q}, \sigma) = (\delta_i(\vec{q}[i], \sigma))_{i=0}^{n}$ for any state $\vec{q} \in Q$ and any symbol $\sigma \in \Sigma$; and the initial state is $\vec{\iota} = (\iota_0, \ldots, \iota_n)$ where the state $\vec{\iota}[i] \in Q_i$ is the initial state of the DFA $\mathcal{A}_i$ for any integer $i = 0 \ldots n$.

Notice that the underlying graph of the DFPA is identical to the underlying graph of the union product of the DFAs [102] corresponding to the outcomes $\{\varphi_0, \ldots, \varphi_n\}$. The DFPA replaces the final states in the union product with a preference graph which can be used to determine the preference relation between two arbitrary words by comparing the sets of final states visited by their paths in the DFPA.

Algorithm 6-1 describes a procedure to construct the preference graph. Given the preference model and the underlying graph of the DFPA, the lines 3–4 of Algorithm 6-1 construct the set of nodes $\mathcal{X}$ by grouping together the states in Q that represent satisfaction of the same set of most-preferred outcomes. These most-preferred outcomes for a state $\vec{q}$ are determined based on the subset of its components $\vec{q}[i], i = 1 \ldots n$, that are final states in the respective DFAs as follows: Let

$$\begin{aligned} \mathsf{Outcomes}(\vec{q}) \triangleq & \{\varphi_i \in \mathbb{F} \mid \vec{q}[i] \in F_i, i = 1 \ldots n\} \cup \\ & \{\varphi_0 \mid \forall i \in \{0 \ldots n\} : \vec{q}[i] \notin F_i\} \end{aligned} \tag{6-1}$$

denote the set of outcomes satisfied by any word in Σ^ω with a good prefix whose last state is $\vec{q} \in Q$. Clearly, $\mathsf{Outcomes}(\vec{q}) = \{\varphi_0\}$ if and only if the word has no prefix that satisfies any of the outcomes in $\{\varphi_1, \ldots, \varphi_n\}$. Then, the set of most-preferred outcomes for $\vec{q}$ is defined as $\mathsf{MP}(\vec{q}) \triangleq \mathsf{MP}(\mathsf{Outcomes}(\vec{q}))$.

The lines 5-7 of Algorithm 6-1 define the edges of the preference graph. An edge from X' to X is added to E if the conditions $\texttt{Cond}_{1a}$ and $\texttt{Cond}_{1b}$ are both `true` at the end of the `for`-loop. The variable $\texttt{Cond}_{1a}$ represents the condition (1a) from Definition 36 that there exists a pair of most-preferred outcomes $\alpha \in \mathsf{MP}(\vec{q})$ and $\alpha' \in \mathsf{MP}(\vec{q}')$ satisfied by any state $\vec{q} \in X$ and $\vec{q}' \in X'$, such that $\alpha \triangleright \alpha'$. The variable $\texttt{Cond}_{1b}$ represents the (1b) from Definition 36. To ensure that $\alpha' \not\triangleright \alpha$ holds for all pairs of most-preferred outcomes $\alpha \in \mathsf{MP}(\vec{q})$ and $\alpha' \in \mathsf{MP}(\vec{q}')$ satisfied by any state $\vec{q} \in X$ and $\vec{q}' \in X'$, the variable $\texttt{Cond}_{1b}$ is initialized to `true` and, whenever a violation is witnessed (lines 11), it is set to `false`.

Proposition 11. *Let $\mathcal{X}$ be the set of nodes constructed by Algorithm 6-1. Then, every state $\vec{q} \in Q$ belongs to a unique node in $\mathcal{X}$, i.e., $\mathcal{X}$ partitions Q.*

Proof. Consider any state $\vec{q}$. We will show that $\vec{q}$ must be contained in some node in $\mathcal{X}$ and it cannot be contained in more than one node. To see that it must be contained in some node, observe that, by construction on line 3, there must exist $\lambda^* \in \Lambda$ such that $\mathsf{MP}(\vec{q}) = \lambda^*$. By construction on line 4, each node in $\mathcal{X}$ corresponds to a unique $\lambda \in \Lambda$. Therefore, $\vec{q}$ must be included in the node corresponding to λ^*. Since the most-preferred set of any subset of $\mathbb{F}$ is unique, the condition $\mathsf{MP}(\vec{q}) = \lambda$ holds for exactly one $\lambda \in \Lambda$, which is λ^*. Therefore, $\vec{q}$ must be included in a unique node $X \in \mathcal{X}$. □

We conclude this section by showing that the DFPA $\mathcal{B} = \langle Q, \Sigma, \delta, \iota, G = \langle \mathcal{X}, E \rangle \rangle$ constructed using Definition 38 and Algorithm 6-1 indeed encodes the preference model $\mathcal{P}$. First, we note that the set of most-preferred outcomes satisfied by the states in the path of any preference graph word satisfies the following property.

Lemma 6-2. *Given any word $w = \sigma_0\sigma_1 \ldots \in \Sigma^\omega$, let $\vec{q}_0\vec{q}_1 \ldots \in Q^\omega$ be the path induced by w in the DFPA. Then, there exists a finite integer $k \geq 0$ such that $\mathsf{Outcomes}(\vec{q}_k) = \mathsf{Outcomes}(w)$. Moreover, for any integer $j > k$, we have $\mathsf{Outcomes}(\vec{q}_j) = \mathsf{Outcomes}(\vec{q}_k)$.*

Proof. Without loss of generality, let $\mathsf{Outcomes}(w) = \{\varphi_1, \ldots, \varphi_m\}$, $0 < m \leq n$, be the subset of outcomes satisfied by the word w. Then, for every integer $i = 1 \ldots m$, there exists an integer $k_i \geq 0$

such that the prefix $\sigma_0 \dots \sigma_{k_i}$ is a good prefix for the scLTL formula φ_i. Choose k to be the largest integer from the set $\{k_1, \dots, k_m\}$. Then, the prefix $\sigma_0 \dots \sigma_k$ is a good prefix for every outcome in $\mathsf{Outcomes}(w)$ because every finite extension of a good prefix is also a good prefix. Since $\delta_i(\vec{q}_k[i], \sigma_0 \dots \sigma_k) \in F_i$ for any good prefix $\sigma_0 \dots \sigma_k$, we have $\mathsf{Outcomes}(\vec{q}_k) = \mathsf{Outcomes}(w)$. □

Because any two states that have identical most-preferred sets are represented by the same node in $\mathcal{X}$, we have the following result.

Corollary 5. *In Lemma 6-2, let $k \geq 0$ be an integer such that $\mathsf{Outcomes}(\vec{q}_k) = \mathsf{Outcomes}(w)$. Then, there exists a unique node $X \in \mathcal{X}$ such that $\vec{q}_j, \vec{q}_k \in X$, for all $j \geq k$.*

Given a word $w \in \Sigma^\omega$, the node $X \in \mathcal{X}$ that satisfies Corollary 5 is called a *terminal node* visited by w.

Theorem 6-1. *Given two words $w, w' \in \Sigma^\omega$, let $\vec{q}_0 \vec{q}_1 \dots \in Q^\omega$ and $\vec{q}'_0 \vec{q}'_1 \dots \in Q^\omega$ be the paths induced by w, w' in DFPA, respectively. Then, for any integer $k \geq 0$ such that $\mathsf{MP}(\vec{q}_k) = \mathsf{MP}(w)$ and $\mathsf{MP}(\vec{q}'_k) = \mathsf{MP}(w')$, the following conditions hold:*

1. *An edge $(X'_k, X_k) \in E$ if and only if $w \succ w'$.*
2. *$X_k = X'_k$ if and only if $w \sim w'$.*
3. *X_k and X'_k are disconnected in G if and only if $w \nparallel w'$.*

where $X_k, X'_k \in \mathcal{X}$ are the nodes that contain $\vec{q}_k, \vec{q}'_k$, respectively.

Proof. (1). Let $k \geq 0$ be an integer such that $\mathsf{MP}(\vec{q}_k) = \mathsf{MP}(w)$ and $\mathsf{MP}(\vec{q}'_k) = \mathsf{MP}(w')$. From Algorithm 6-1, we know that an edge $(X'_k, X_k) \in E$ exists if and only if the following conditions hold: (a) there exists $\alpha, \alpha' \in \mathbb{F}$ such that $\alpha \in \mathsf{MP}(\vec{q}_k), \alpha' \in \mathsf{MP}(\vec{q}'_k)$ and $\alpha \triangleright \alpha'$, and (b) for all $\alpha, \alpha' \in \mathbb{F}$ such that $\alpha \in \mathsf{MP}(\vec{q}_k), \alpha' \in \mathsf{MP}(\vec{q}'_k)$, we have $\alpha' \not\triangleright \alpha$. Since $\mathsf{MP}(\vec{q}_k) = \mathsf{MP}(w)$ and $\mathsf{MP}(\vec{q}'_k) = \mathsf{MP}(w')$ is known, the conditions (a) and (b) reduce to the condition (1a) and (1b) from Definition 34. Finally, the statement (1) follows by Corollary 5. The proofs of (2), (3) follow similarly. □

6.3 Solution Concepts

In preference-based planning, the agent is to choose its next action given a finite prefix $\nu \in \mathsf{PrefPaths}(M)$ in order to satisfy the given preference relation on a set of outcomes. A naïve approach to this problem is to follow the strategy to satisfy a most-preferred outcome from the set of almost-surely achievable outcomes given ν. However, this is not sufficient, as illustrated by the following example.

Example 6. Consider the toy MDP shown in Fig. (6-1). The exact probabilities are omitted because we analyze the MDP qualitatively. The transitions are understood as follows: Given action a at state s_0, it is possible to reach both s_5 and s_1 with positive probabilities.

Let $F_1 = \{s_1, s_5\}, F_2 = \{s_2, s_4\}$ and $F_3 = \{s_3\}$ be three sets of final states. Let preference formula be $\lozenge F_2 \rhd \lozenge F_1 \wedge \lozenge F_3 \rhd \lozenge F_1$. Clearly, $\lozenge F_2$ and $\lozenge F_3$ are incomparable. Therefore, the play $\rho_1 = s_0 s_3^\omega$, which satisfies $\lozenge F_3$, is strictly preferred to the play $\rho_2 = s_0 s_5 s_1^\omega$, which satisfies $\lozenge F_1$. Whereas, ρ_1 is incomparable to the play $\rho_3 = s_0 s_4^\omega$ because it satisfies $\lozenge F_2$.

Consider the state s_0 at which the agent is to choose its next action. From s_0, the agent can visit F_1 almost surely by choosing a. It, however, does not have an almost sure winning strategy to visit either F_2 or F_3, individually. But, by choosing b at s_0, the agent almost surely visits either F_2 or F_3 and achieves a strictly better outcome than F_1.

The example highlights that the almost sure winning solution concept is not suitable for preference-based planning because it reasons about exactly one outcome at a time. As a result, the agent cannot reason about opportunities to achieve a better outcome that may become available due to stochasticity in the environment.

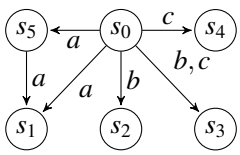


Figure 6-1. Toy example to illustrate the limitation of almost-sure winning solution concept for preference-based planning. The states with no outgoing transitions are sink states (the self-loops are omitted for clarity).

In the sequel, we introduce two new solution concepts for probabilistic planning under incomplete preferences interpreted over infinite plays. Our solution concepts are based upon the notion of an *improvement* that generalizes the idea of *improving flip* [103] which is defined for propositional preferences. An improving flip compares two outcomes representable as propositional logic formulas to determine which is more preferred. Analogously, an improvement compares two prefixes of a play to determine which one can yield a more preferred outcome with probability one.

Given a prefix ν, let $\mathsf{Outcomes}(\nu) = \{\varphi \in \mathbb{F} \mid \exists \pi \in \Pi, \forall \rho \in \mathsf{Cone}(M, \nu, \pi) : \rho \models \varphi\}$ be the set of outcomes, each of which can be achieved almost-surely under some strategy. Note that different outcomes may require different policies to achieve them.

Definition 39. Given a play $\rho \in \mathsf{Plays}(M)$ and two of its prefixes $\nu, \nu' \in \mathsf{Pref}(\rho)$ such that $|\nu'| > |\nu|$, ν' is said to be an *improvement* of ν if there exists a pair of outcomes $R \in \mathsf{MP}(\mathsf{Outcomes}(\nu))$ and $R' \in \mathsf{MP}(\mathsf{Outcomes}(\nu'))$ such that $R' \triangleright R$. And, ν' is said to be a *weakening* of ν if there exists a pair of outcomes $R \in \mathsf{MP}(\mathsf{Outcomes}(\nu))$ and $R' \in \mathsf{MP}(\mathsf{Outcomes}(\nu'))$ such that $R \triangleright R'$.

Given a prefix $s_0 s_1 \ldots s_k \in \mathsf{PrefPaths}(M)$, the transition from s_{k-1} to s_k is said to be an *improving transition* if the prefix $s_0 s_1 \ldots s_{k-1} s_k$ is an improvement over $s_0 s_1 \ldots s_{k-1}$. A play that contains an improving transition is called an *improving play*. It is noted that a prefix ν' can simultaneously be an improvement and a weakening of a prefix ν.

Next, we define the two solution concepts that, while avoiding any weakening, induce improvements either with positive probability or with probability one.

Definition 40 (SPI/SASI Strategy)**.** Given a prefix $\nu = s_0 s_1 \ldots s_k \in \mathsf{PrefPaths}(M)$, a strategy $\pi : S^+ \to 2^A$ is said to be *safe and positively (resp., safe and almost-surely) improving* for ν if the following conditions hold:

1. (Safety) For all $\rho \in \mathsf{Cone}(M, \nu, \pi)$, the play $\nu\rho$ satisfies that $s_0 s_1 \ldots s_j$ is not a weakening of $s_0 s_1 \ldots s_k$ for any integer $j > k$.

2. (Improvement) There exists (resp., for any) $\rho \in \mathsf{Cone}(M, \nu, \pi)$, the play $\nu\rho$ satisfies the condition that there exists an integer $j > k$ such that $s_0 s_1 \dots s_j$ is an improvement over $s_0 s_1 \dots s_k$.

We now state our problem statement.

Problem 7. Given an MDP M and a preference model $\langle \mathbb{F}, \trianglerighteq \rangle$, design an algorithm to synthesize an SPI and a SASI strategy.

6.4 Synthesis of Opportunistic Preference Satisfying Strategies

In this section, we show how to synthesize the positive and almost-surely preference satisfying strategies in MDP given the DFPA corresponding to a preference formula φ. We begin by constructing a product of an MDP and a DFPA that allows us to reason simultaneously about the stochastic environment and the preference model.

Definition 41. Given an MDP $M = \langle S, A, T, \mathcal{AP}, L \rangle$ and a DFPA $\mathcal{B} = \langle Q, \Sigma, \delta, \iota, G = (\mathcal{X}, E) \rangle$, the product of the MDP and DFPA is the tuple,

$$\mathcal{M} \triangleq \langle V, A, \Delta, \mathcal{G} \triangleq \langle \widetilde{\mathcal{X}}, \mathcal{E} \rangle \rangle,$$

where $V := S \times Q$ is the finite set of states. A is the same set of actions as M. The transition function $\Delta : V \times A \to \mathcal{D}(V)$ is defined as follows: for any states $(s, \vec{q}), (s', \vec{q}') \in V$ and any action $a \in A$, $\Delta((s', \vec{q}') \mid (s, \vec{q}), a) = P(s' \mid s, a)$ if $\vec{q}' \in \delta(\vec{q}, L(s'))$ and 0 otherwise. The component $\mathcal{G} = (\widetilde{\mathcal{X}}, \mathcal{E})$ is a graph where $\widetilde{\mathcal{X}} \triangleq \{S \times X \mid X \in \mathcal{X}\}$ is the set of nodes and $\mathcal{E}$ is a set of edges such that, for any $\widetilde{X}_i = S \times X_i$ and $\widetilde{X}_j = S \times X_j$, $(\widetilde{X}_i, \widetilde{X}_j) \in \mathcal{E}$ if and only if $(X_i, X_j) \in E$.

A path in the product MDP is an infinite sequence of states $\rho = v_0 v_1 \dots \in V^\omega$ such that there exists an action $a \in A$ such that $\Delta(v_{i+1} \mid v_i, a) > 0$ holds for all $i \geq 0$. Letting $v_i = (s_i, \vec{q}_i)$ for all $i \geq 0$, we define the projection of a path $\rho \in V^\omega$ onto the DFPA $\mathcal{B}$ as the path $\rho = \rho \downarrow_{\mathcal{B}} \triangleq \vec{q}_0 \vec{q}_1 \dots \in Q^\omega$ in $\mathcal{B}$. The projection maps a path in the product MDP to the corresponding path in the DFPA. Given a path $\rho \in V^\omega$, we denote the set of outcomes satisfied by

ρ by $\mathsf{Outcomes}(\rho)$. The following result, which is a consequence of [104, Prop. 1], states that an outcome $\alpha \in \mathbb{F}$ is satisfied by ρ if and only if its projection $\rho \restriction_{\mathcal{B}}$ satisfies φ.

Proposition 12. *For any path ρ in $\mathcal{M}$,* $\mathsf{Outcomes}(\rho) = \mathsf{Outcomes}(\rho \restriction_{\mathcal{B}})$ *and* $\mathsf{MP}(\rho) = \mathsf{MP}(\rho \restriction_{\mathcal{B}})$.

Due to Proposition 12, the preference between two paths in the product MDP can be determined by comparing their projections onto the DFPA and using Theorem 6-1. But recall that, Problem 7 asks us to design a positive (resp., almost-sure) preference satisfying strategy that achieves no worse outcome than that possible by any other strategy with positive probability (probability one).

Our approach to synthesize SPI and SASI strategies distinguishes between *opportunistic* states, *i.e.*, the states from which an improvement could be made, and *non-opportunistic* states. We now introduce a new model called *an improvement MDP* to synthesize the SPI and SASI strategies.

To facilitate the definition, we slightly abuse the notation and let $\mathsf{MP}((s,\vec{q})) \triangleq \mathsf{MP}(\{\varphi \in \mathbb{F} \mid \varphi \in \mathsf{Outcomes}(\vec{q})\})$ be the set of outcomes almost surely achievable from state v in $\mathcal{M}$.

Definition 42 (Improvement MDP)**.** Given a product MDP $\mathcal{M}$, an *improvement MDP* is the tuple,

$$\widetilde{\mathcal{M}} = \langle \widetilde{V}, A, \widetilde{\Delta}, v_0, \widetilde{\mathcal{F}} \rangle,$$

where $\widetilde{V} = V \times \{0,1\}$ is the set of states, A is the same set of actions as $\mathcal{M}$, $\widetilde{v}_0 = (v_0, 0)$ is the initial state, and $\mathcal{F} = \{(v,1) \mid v \in V\}$ is a set of final states that can only be reached by making an improvement. The transition function $\Delta : \widetilde{V} \times A \to \mathcal{D}(\widetilde{V})$ is defined as follows: For any states $\widetilde{v} = (v,m), \widetilde{v}' = (v',m') \in \widetilde{V}$ such that $v \in \widetilde{X}$ and $v' \in \widetilde{X}'$ and for any action $a \in A$, $\Delta(\widetilde{v}, a, \widetilde{v}') > 0$ holds if and only if the following conditions hold: $T(v,a,v') > 0$ and either $(\widetilde{X}, \widetilde{X}') \in \mathcal{E}$ and $m' = 1$ holds or $\widetilde{X} = \widetilde{X}'$ and $m' = 0$ holds.

Every play $\rho = v_0 v_1 \ldots \in \mathsf{Plays}(\mathcal{M})$ induces a play $\rho = \widetilde{v}_0 \widetilde{v}_1 \ldots$ in $\widetilde{\mathcal{M}}$ such that for all $i = 0, 1, \ldots$, $\widetilde{v}_i = (v_i, m_i)$ where $m_i \in \{0, 1\}$ represents a memory element such that $m_i = 1$ if and only if the transition from v_{i-1} to v_i is improving. The following proposition highlights important features of the improvement MDP. Before that, we note the following fact to prove Proposition 13.

Lemma 6-3. *For every prefix $\nu = v_0 v_1 \ldots v_k \in \mathsf{PrefPaths}(\mathcal{M})$, it holds that $\mathsf{Outcomes}(\nu) = \mathsf{Outcomes}(v_k)$ and thus $\mathsf{MP}(\mathsf{Outcomes}(\nu)) = \mathsf{MP}(\mathsf{Outcomes}(v_k))$.*

The proof follows from the fact that memoryless strategies are sufficient to ensure the satisfaction of reachability objectives in MDPs [20]. In other words, if an outcome is almost surely achievable given a prefix $\nu = v_0 v_1 \ldots v_k$, then it is almost surely achievable given v_k.

For convenience, we write $\mathsf{MP}(v) = \mathsf{MP}(\mathsf{Outcomes}(v))$ to denote the set of most preferred outcomes satisfiable/achievable with some strategy from a state $v \in V$.

Proposition 13. *For any play $\rho = \widetilde{v}_0 \widetilde{v}_1 \ldots \in \mathsf{Plays}(\widetilde{\mathcal{M}})$ such that $\widetilde{v}_i = (v_i, m_i)$ for all $i = 0, 1 \ldots$, the following statements hold.*

1. *(Safety). For every prefix $\widetilde{v}_0 \widetilde{v}_1 \ldots \widetilde{v}_j \in \mathsf{PrefPaths}(\rho)$, $v_0 v_1 \ldots v_j$ is not a weakening of $v_0 v_1 \ldots v_i$ for any $0 \leq i < j$.*
2. *(Improvement). For every integer $k > 0$ such that $v_k \in \mathcal{F}$, the prefix $v_0 v_1 \ldots v_k$ is an improvement of $v_0 v_1 \ldots v_{k-1}$.*

Proof (Sketch). For statement (1) to hold, it must be the case that $R \not\triangleright R'$ holds for all pairs of outcomes $R \in \mathsf{MP}(s_i)$ and $R' \in \mathsf{MP}(s_j)$. This is true because of Lma. 6-3 and the fact that every transition from $\widetilde{v}_i$ to $\widetilde{v}_{i+1}$, $j < i \leq k$, that violates the condition is disabled by Def. 42.

To see why statement (2) holds, consider an integer $k > 0$ such that $\widetilde{v}_k \in \mathcal{F}$. Then, by construction, there exists a pair $R \in \mathsf{MP}(v_{k-1})$ and $R' \in \mathsf{MP}(v_k)$ such that $R' \triangleright R$. □

In words, the improvement MDP guarantees by construction that no play in $\mathsf{Plays}(\widetilde{\mathcal{M}})$ violates the safety condition of Def. 40. Moreover, it helps identify the opportunistic states as the ones that have an outgoing transition into $\widetilde{\mathcal{F}}$.

Corollary 6. *A play* $\rho \in \mathsf{Plays}(\widetilde{\mathcal{M}})$ *is improving if and only if* $\mathsf{Occ}(\rho) \cap \widetilde{\mathcal{F}} \neq \emptyset$.

As a result, the problem of determining whether an improvement is possible from a state $\widetilde{v} \in \widetilde{V}$ reduces to checking whether a state in $\widetilde{\mathcal{F}}$ can be reached from $\widetilde{v}$ with a positive probability (in case of SPI strategy) or with probability one (in case of SASI strategy).

Theorem 6-2. *The following statements hold:*

1. *Any positive winning strategy* $\pi^{\mathsf{PWin}(\widetilde{\mathcal{F}})}$ *in* $\widetilde{\mathcal{M}}$ *is an SPI strategy.*
2. *Any almost-sure winning strategy* $\pi^{\mathsf{ASWin}(\widetilde{\mathcal{F}})}$ *in* $\widetilde{\mathcal{M}}$ *is an SASI strategy.*

The proof follows from the fact that there exists a (resp., every) play $\rho \in \mathsf{Cone}(\widetilde{\mathcal{M}}, \widetilde{v}_0, \pi)$ induced by any positive (resp., almost-sure) winning strategy π visits $\widetilde{\mathcal{F}}$ with positive probability (resp., probability one) [105]. Therefore, Thm. 6-2 establishes that by following $\pi^{\mathsf{PWin}(\widetilde{\mathcal{F}})}$ (resp., $\pi^{\mathsf{ASWin}(\widetilde{\mathcal{F}})}$), the agent is ensured to make an improvement with a positive probability (resp., with probability one). It is noted that an SPI (resp., SASI) strategy exists if and only if the corresponding positive (resp., almost-sure) winning strategy exists in $\widetilde{\mathcal{M}}$.

The SPI and SASI strategies from Thm. 6-2 guarantee that at least one improvement will occur with positive probability or with probability one. Next, we present Alg. 6-2, using which we can determine the maximum number of improvements that can *almost surely* be made from a given state in $\widetilde{\mathcal{M}}$. The algorithm to determine the maximum number of improvements possible from a given state in $\widetilde{\mathcal{M}}$ with *a positive probability* and its properties are similar to Alg. 6-2.

First, note the following properties of the improvement MDP which follow from the construction of MDP.

Proposition 14. *Consider two states* $(v,0),(v,1) \in \widetilde{V}$*, it holds that for any action* $a \in A$*, we have* $\mathsf{Supp}(\widetilde{\Delta}((v,0),a)) = \mathsf{Supp}(\widetilde{\Delta}((v,1),a))$.

The proof is straightforward because given $(v,0),(v,1)$, for any action $a \in A$, if a transition from v to v' given a is improving, then $\widetilde{\Delta}((v,0),a,(v',1)) > 0$ and $\widetilde{\Delta}((v,1),a,(v',1)) > 0$. Else, $\widetilde{\Delta}((v,0),a,(v',0)) > 0$ and $\widetilde{\Delta}((v,1),a,(v',0)) > 0$.

Algorithm 6-2 Level set for constructing safe and almost-surely improving strategy.

Inputs: Improvement MDP, $\widetilde{\mathcal{M}}$.
Outputs: Level set, $\mathcal{W}$.

$i \leftarrow 0$
$R_i \leftarrow \mathcal{F}$
while R_i is not empty **do**
    $W_{i+1} \leftarrow \mathsf{ASWin}(R_i)$
    $R_{i+1} \leftarrow \{(v,1) \in \mathcal{F} \mid (v,0) \in W_{i+1}\}$
    if $i = 0$ **then**
        Add $\widetilde{V} \setminus W_{i+1}$ to level 0 in $\mathcal{W}$.
    end if
    Add W_{i+1} to level $i+1$ in $\mathcal{W}$.
    $i \leftarrow i+1$
end while
return $\mathcal{W}$

Corollary 7. *The final states $\widetilde{\mathcal{F}}$ can be visited* again *from a state $(v,1) \in \widetilde{V}$ with a positive probability (resp., with probability one) if and only if $\widetilde{\mathcal{F}}$ can be visited from $(v,0)$ with a positive probability (resp., with probability one).*

Proof. Let π be a positive winning strategy to visit $\widetilde{\mathcal{F}}$ from $(v,0)$. Let $Y = \mathsf{Supp}(\widetilde{\Delta}((v,0),a))$ for some $a \in \pi((v,0))$. By the property of a positive winning strategy, a state in $\widetilde{\mathcal{F}}$ is reached with positive probability by following π from any state in Y. By Proposition 14, $Y = \mathsf{Supp}(\widetilde{\Delta}((v,1),a))$. Therefore, by choosing a at $(v,1)$ and then following π, a state in $\widetilde{\mathcal{F}}$ is visited with positive probability from $(v,1)$. The proof for almost-sure winning is similar. □

Intuitively, Alg. 6-2 constructs a set $\mathcal{W}$ of level sets such that from any state that appears at k-th level in $\mathcal{W}$, at least k visits to $\widetilde{\mathcal{F}}$ are guaranteed and, thereby, at least k improvements can be made.

For this purpose, it iteratively computes the almost-sure winning region to visit the states in $R_i \subseteq \widetilde{\mathcal{F}}$, from which $\widetilde{\mathcal{F}}$ can be visited at least i times. We denote by W_i the i-th level set. The level-0 of $\mathcal{W}$ contains the states $\widetilde{V} \setminus \mathsf{ASWin}(\widetilde{\mathcal{F}})$ from which $\widetilde{\mathcal{F}}$ cannot be visited again with probability one. That is, 0-visits to $\widetilde{\mathcal{F}}$ are guaranteed from any state in level-0 of $\mathcal{W}$. Every state in level-1 of $\mathcal{W}$ is almost surely winning to visit $\widetilde{\mathcal{F}}$. Hence, at least one visit to $\widetilde{\mathcal{F}}$ is guaranteed.

Now, consider the subset $R_1 = \{(v,1) \in \mathcal{F} \mid (v,0) \in W_1\}$ of final states $\widetilde{\mathcal{F}}$. By Corollary 7, because $(v,0) \in W_1 = \mathsf{ASWin}(\widetilde{\mathcal{F}})$, there exists a strategy from every state in R_1 to visit $\widetilde{\mathcal{F}}$ with probability one. Therefore, from any state $(v,0) \in W_2 = \mathsf{ASWin}(R_1)$ at least two improvements are guaranteed—first, when visiting $(v',1) \in R_1$ and, second, when visiting $R_0 = \widetilde{\mathcal{F}}$ by following the almost-sure winning strategy at $(v',1)$. Repeating a similar argument, it follows that at least k-visits are guaranteed almost surely from states at k-th level in $\mathcal{W}$.

The largest integer $k \geq 0$ such that the state $(v,0) \in \widetilde{V}$ appears at k-th level of $\mathcal{W}$ is called the rank of the states $(v,0)$ and $(v,1)$, denoted as $\mathsf{rank}(v,0) = \mathsf{rank}(v,1) = k$.

Proposition 15. *From any state $\widetilde{v} = (v,m) \in \widetilde{V}$, $m \in \{0,1\}$, there exists a strategy to visit $\widetilde{\mathcal{F}}$ at least $\mathsf{rank}(\widetilde{v})$-many times.*

Proof. We construct the strategy that achieves $\mathsf{rank}(\widetilde{v})$ improvements: First, if $\mathsf{rank}(\widetilde{v}) = k$, then by construction it is in $\mathsf{ASWin}(R_{k-1})$. Following the almost-sure winning strategy a state in R_{k-1} can be reached with probability one and thus the first improvement is made. Upon reaching a state, say $(v',1)$, in R_{k-1}, we have $(v',0) \in W_{k-1}$. Because $W_{k-1} = \mathsf{ASWin}(R_{k-2})$, an almost-sure winning strategy exists to reach R_{k-2} and hence the second improvement. Repeating similar steps, eventually, R_0 will be reached after the k-th improvement. □

Corollary 8. *From any state $\widetilde{v} = (v,m) \in \widetilde{V}$ at most $\mathsf{rank}(\widetilde{v})$-many visits to $\widetilde{\mathcal{F}}$ are almost surely guaranteed.*

Proof (Sketch). By contradiction. Suppose that $\mathsf{rank}(\widetilde{v}) = k$ but $k+1$ visits to $\widetilde{\mathcal{F}}$ are possible from $\widetilde{v}$. Since $k+1$ visits are possible from $\widetilde{v}$, by definition of $\mathcal{W}$, it must be the case that $\widetilde{v} \in W_{k+1}$. If $\widetilde{v}$ is at $(k+1)$-th level in $\mathcal{W}$ then its rank must be at least $k+1$—a contradiction. □

The proof of Proposition 15 defines the strategy that allows the agent to make $\mathsf{rank}(v)$ improvements from any state v.

Complexity. Alg. 6-2 runs in polynomial time with respect to the size of $\widetilde{\mathcal{M}}$ since the while loop can run no more than $|\widetilde{V}|$ times and the complexity of ASWin is quadratic in the size of $\widetilde{\mathcal{M}}$ [78].

6.5 Example: Robot Motion Planning in Stochastic Gridworld

We illustrate our approach using a motion planning problem for a robot in a 5×5 gridworld as shown in Fig. (6-2). The gridworld environment consists of seven regions: $\{A : (0,0), B : (2,0), C : (4,0), D : (2,4), E : (4,4), F : (1,2)\}$ from which the robot must pick up an item. There is a charging station at cell $(4,2)$. Each cell is denoted using the convention `(row, col)`. The robot can choose among four actions `N, S, E, W` to deterministically move north, east, south, and west by one cell. The actions `E, W` are disabled in the cells $(4,2)$ and $(2,2)$. The cells $(1,1), (3,1), (1,3), (3,3)$ are slippery; that is, whenever the robot moves into any of these cells, say $(1,1)$, it may non-deterministically end up in either the same cell $(1,1)$, or the cell north to it $(2,1)$, or south to it $(0,1)$. In any cell, if applying an action results in a cell that is outside the gridworld or contains an obstacle, the robot returns to the same cell. The robot has a limited battery of 8 units, which it may recharge by visiting the charging station. The robot spends 1 unit to execute each action.

Initially, only the items at A, B, and C are available for pickup. That is, if the robot visits the charging station or regions D, E, F, then neither its battery will be recharged nor will it be able to pick up items D, E, F. When the robot picks up an item at A or B, the charging station and the items at D, E become available. When the robot picks up an item at C, the charging station and the items at E, F become available. The following preference about picking up the items is given to the robot:

$$(\lozenge D \triangleright \lozenge A) \wedge (\lozenge E \triangleright \lozenge A) \wedge (\lozenge D \triangleright \lozenge B) \wedge (\lozenge E \triangleright \lozenge B) \wedge (\lozenge E \triangleright \lozenge C) \wedge (\lozenge F \triangleright \lozenge C).$$

By default, picking up any item is preferred to not picking up any item.

	SASI	SPI
Rank-1	768	926
Rank-2	98	167

Table 6-1. Number of states from which the robot has a safe and positively improving and safe and almost-surely improving strategies to make at least 1 or at least 2 improvements.

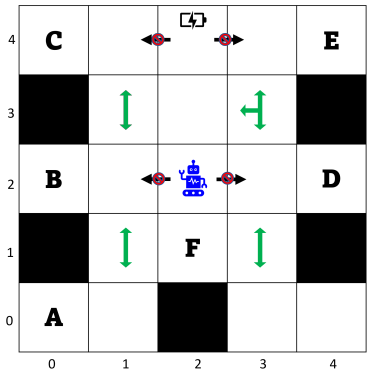


Figure 6-2. A gridworld example in which the black arrows with no-entry symbol denote the disabled actions from that state and green arrows show the random outcomes on entering the cell.

Note that the preference model given to the robot is incomplete as well as combinative. It is incomplete because picking up items A, B, C are mutually incomparable outcomes. Similarly, picking up items D, E, F are mutually incomparable. It is combinative because, for instance, any play in which robot picks up an item from D or E is considered preferred to a play in which robot only picks an item from A or B, even though to pick an item from D or E an item from A or B must be picked first.

We implemented the example in Python 3.9 on a Windows 10 machine with a core i7, 2.80GHz CPU, and 32GB memory. The SPI and SASI strategies are computed using set-based positive and almost-sure winning algorithms implemented in https://github.com/abhibp1993/ggsolver/. We discuss a few noteworthy observations next. The improvement MDP for this case has 3600 states and 18496 transitions, whereas the improvement MDP has 7200 states and 35524 transitions. The time required for constructing the improvement MDP is 9.47 seconds which includes time required to solve for almost-sure winning regions to visit A-F independently. Whereas, the construction of SASI and SPI strategies took 6.54 seconds and 7.23 seconds, respectively.

Consider the initial state $s_0 = (2, 2, 8, (1, 1, 1, 0, 0, 0, 0))$ in which the robot is at cell $(2, 2)$ with 8 units of battery. The fourth component of the state denotes which items are available for pickup, with the last element of the tuple reserved for the availability of the charging station. In

this state, the robot has no almost-sure winning strategy to visit any of A, B, or C. This is because to visit, say, A; the robot must visit the slippery cell $(1,1)$. But whenever $(1,1)$ is visited, the robot may reach $(2,1)$ with a positive probability. Hence, $\mathsf{MP}(\mathsf{Outcomes}(s_0)) = \emptyset$.

When we use the SASI concept, the rank of the state $(s_0,1)$ is 2, indicating that two improvements are almost surely guaranteed. This is understood by observing the SASI strategy which chooses action `N` at $(s_0,0)$ to reach $s_1 = (3,2,7,(1,1,1,0,0,0,0))$. At $(s_1,0)$ the strategy selects `W` and visits either B or C with probability one. Since a pickup from B and C are incomparable, both actions `N` and `S` are deemed valid under SASI strategy at $(3,1,6,(1,1,1,0,0,0,0))$. On visiting either B or C, the SASI strategy follows the almost-sure winning strategy to visit either D or E to make a second improvement. Since visiting cell $(3,3)$ may result in returning back to cell $(3,2)$ with a positive probability, the robot can recharge itself until a successful visit to E or D is made.

The SASI strategy at $(s_0,0)$ does not select `S` because a second improvement cannot be guaranteed with probability one after visiting A since the robot may remain at the cell $(0,1)$ until its battery runs out. However, we observe that the SPI strategy at $(s_0,0)$ allows the selection of both actions `N`, `S` at $(s_0,0)$ since in both cases, two improvements are possible with positive probability.

We conclude with Table. 6-1 that shows the number of states from which the robot has an SPI and SASI strategies to make at least 1 or 2 improvements, since the maximum number of improvements possible under given preference model is 2. We note that the states from which a SASI strategy exists are a subset of states from which an SPI strategy exists.

CHAPTER 7
CONCLUSION AND PERSPECTIVES

In this dissertation we have studied the synthesis of winning strategies in games on graphs with two kinds of incomplete information: exteroceptive and interoceptive. We studied three fundamental classes of two-player games on graphs with one-sided exteroceptive incomplete information and the synthesis of preference satisfying strategies in single-player stochastic games with interoceptive incomplete information.

7.1 Achievements and Perspectives

The key achievements of this dissertation are as follows.

Hypergame theory for games on graphs. This dissertation lays the foundations for studying hypergame theory for games on graphs. We define the categorization of hypergames in two ways based on whether the perceptions of players remain static or evolve during the interaction. We introduce a static hypergame on graph to model an interaction where perceptions of players remain constant throughout the interaction. We study static hypergames on graphs under two settings. First, when P2's perception remains constant because of its ignorance or its incapability to update its perception based on observations. In this setting, we show that P1 can synthesize opportunistic strategies, which capitalize on P2's misperception to enforce a win from an otherwise losing state (*i.e.*, a losing state in the game with perfect and complete information). Second, when P2 has the capability of updating its perception, but P1 intentionally prevents it by only selecting actions that are subjectively rationalizable for P2. We formalize this idea by introducing the solution concepts of stealthy deceptive sure winning and stealthy deceptive almost-sure winning. We introduce a *dynamic hypergame* to model situations where P2's perception could evolve during the game. A dynamic hypergame captures the evolution of P2's subjectively rationalizable strategies with respect to changes in its perception. For these models, we introduce the solution concepts of deceptive sure winning and deceptive almost-sure winning. Both these concepts are not stealthy since the model permits the perceptions of players to evolve. For the three fundamental classes of misperceptions possible in games on graphs, this dissertation

investigates the important properties of hypergame on graphs and presents the algorithms to synthesize winning strategies for P1 and P2 under the introduced solution concepts.

From a high-level perspective, the hypergame-theoretic approach used to studying games on graphs with incomplete information enables us to model and analyze the rational behavior of the players who may be unaware of their misperception and may have an ability to update their perceptions by observing the history of their interaction. This not only pushes the boundary of the state-of-the-art in sequential decision-making in infinite-duration interaction but also in significantly advances the field of games with incomplete information. Most importantly, we have employed a reductionist approach wherever possible: Except for the class of action misperception, we successfully reduce the synthesis problem for a game on graph with incomplete information to that of synthesizing a winning strategy for a game on graph with perfect and complete information. In static hypergames, we observed that the reduction only adds a polynomial-time overhead. Thus, overall, the synthesis procedure completes within polynomial-time. This observation is particularly important because the conventional Bayesian games approach would have first transformed the game with incomplete information to that with imperfect information. And, most algorithms games on graphs with imperfect information require at least exponential-time to synthesize winning strategies.

Automata-theoretic approach to preference-based planning. This dissertation introduces an automata-theoretic approach to synthesizing strategy given incomplete preferences over temporal goals. Following a declarative paradigm, we define a language to specify a preference over a set of scLTL specifications, a procedure to translate the specification into a computational model, *i.e.*, a newly introduced preference automaton, and design a procedure to use the preference automaton to synthesize a preference satisfying strategy in a stochastic environment under two solution concepts: safe and positively improving, and safe and almost-surely improving.

Our solution makes a fundamental contribution to addressing the problem of sequential decision-making under combinative, incomplete preferences, which may require the agent to

choose between incomparable outcomes. This is mainly because the classical approaches to decision theory that rely upon dominance principle fail in this situation.

Computation tools. Finally, all the algorithms introduced in this dissertation were implemented in a unified framework for solving games on graphs available at www.akulkarni.me/software.

7.2 Future Work

The approaches presented in this thesis open up a set of directions for future work.

Hypergames on graphs. Our development of hypergames is still in its early stages. We enlist three directions for potential investigation. First, we only consider up to level-2 hypergames in this work. A level-2 hypergame can model situations where at least one player knows that its opponent misperceives certain component of the game. But the opponent does not know that the player is aware of this fact. However, average humans are known to reason upto six levels (think of yourself playing a card game like Bridge or not-at-home), whereas some advanced poker or chess players can reason upto eight levels. In this regard, for the autonomous agents to interact effectively with humans, they should at least be able to reason at eight levels, if not more.

Second, the solution concepts introduced in this work are based on the concept of subjective rationalizability in normal-form hypergames. As discussed in the introduction, there are several solution concepts for hypergames that provide insights into rational behavior of agents under incomplete information. For example, the Fraser-Hipel equilibrium investigates the behavior of agents when a subset of them can impose "sanctions" on a player who might unilaterally deviate from an equilibrium point. At present, it is unclear whether these solution concepts are related to any of the known solution concepts for games on graphs. Investigating these connections could lead to deeper insights into sequential decision-making under incomplete information, especially in multi-agent settings.

Lastly, the algorithms presented in this dissertation were designed with the aim of being intuitive and easy to prove their correctness. Hence, there may exist more efficient algorithms to

solve the synthesis problem for these classes. It would be worthwhile to investigate and establish the lower-bound on the complexity of solving these class of problems.

Planning with incomplete preferences. We propose two future directions. First, it would be useful to investigate various ways to define semantics of the preference language. It is non-trivial to define a "good" way to interpret a preference language when incompleteness is permitted. To compare two outcomes (*i.e.*, infinite paths), one must compare the sets of temporal logic formulas satisfied by those outcomes. There are numerous ways to define this comparison. It is also clear to us that no one way is the correct way! Instead, the usefulness of semantics is application driven.

Second, the present work considers single-player stochastic games on graphs with incomplete preferences. It would be interesting to investigate the synthesis problem for two or more player games on graphs where each player plays to maximally satisfy its preference relation. Such games are of great interest to domains such as social choice theory, economics, and database systems apart from game theory.

LIST OF REFERENCES

[1] Z. Aslanyan, F. Nielson, and D. Parker, "Quantitative verification and synthesis of attack-defence scenarios," in *2016 IEEE 29th Computer Security Foundations Symposium (CSF)*, pp. 105–119, IEEE, 2016.

[2] S. Jha, O. Sheyner, and J. Wing, "Two formal analyses of attack graphs," in *Proceedings 15th IEEE Computer Security Foundations Workshop. CSFW-15*, pp. 49–63, IEEE, 2002.

[3] R. R. Hansen, P. G. Jensen, K. G. Larsen, A. Legay, and D. B. Poulsen, "Quantitative evaluation of attack defense trees using stochastic timed automata," in *International Workshop on Graphical Models for Security*, pp. 75–90, Springer, 2017.

[4] A. N. Kulkarni, J. Fu, H. Luo, C. A. Kamhoua, and N. O. Leslie, "Decoy allocation games on graphs with temporal logic objectives," in *International Conference on Decision and Game Theory for Security*, pp. 168–187, Springer, 2020.

[5] G. E. Fainekos, A. Girard, H. Kress-Gazit, and G. J. Pappas, "Temporal logic motion planning for dynamic robots," *Automatica*, vol. 45, no. 2, pp. 343–352, 2009.

[6] H. Kress-Gazit, G. Fainekos, and G. J. Pappas, "Temporal-logic-based reactive mission and motion planning," *IEEE Transactions on Robotics*, vol. 25, pp. 1370–1381, 2009.

[7] P. J. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proceedings of the IEEE*, vol. 77, no. 1, pp. 81–98, 1989.

[8] A. Puri, "Theory of hybrid systems and discrete event systems," 1996.

[9] L. De Alfaro, T. A. Henzinger, and O. Kupferman, "Concurrent reachability games," *Theoretical Computer Science*, vol. 386, no. 3, pp. 188–217, 2007.

[10] O. Kupferman and M. Y. Vardi, "Model checking of safety properties," *Formal Methods in System Design*, vol. 19, no. 3, pp. 291–314, 2001.

[11] J. C. Harsanyi, "Games with incomplete information played by "bayesian" players, i–iii part i. the basic model," *Management science*, vol. 14, no. 3, pp. 159–182, 1967.

[12] J. H. Reif, "The complexity of two-player games of incomplete information," *J. Comput. Syst. Sci.*, vol. 29, pp. 274–301, 1984.

[13] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artif. Intell.*, vol. 101, pp. 99–134, 1998.

[14] E. A. Hansen, D. S. Bernstein, and S. Zilberstein, "Dynamic programming for partially observable stochastic games," in *AAAI Conference on Artificial Intelligence*, 2004.

[15] J. D. Levin, "Dynamic games with incomplete information," 2002.

[16] N. Bertrand, B. Genest, and H. Gimbert, "Qualitative determinacy and decidability of stochastic games with signals," *2009 24th Annual IEEE Symposium on Logic In Computer Science*, pp. 319–328, 2009.

[17] D. A. Martin, "Borel determinacy," *Annals of Mathematics*, vol. 102, no. 2, pp. 363–371, 1975.

[18] D. A. Martin, "The determinacy of blackwell games," *The Journal of Symbolic Logic*, vol. 63, no. 4, pp. 1565–1581, 1998.

[19] L. De Alfaro and T. A. Henzinger, "Concurrent omega-regular games," in *Proceedings Fifteenth Annual IEEE Symposium on Logic in Computer Science (Cat. No. 99CB36332)*, pp. 141–154, IEEE, 2000.

[20] L. de Alfaro and R. Majumdar, "Quantitative solution of omega-regular games," in *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pp. 675–683, 2001.

[21] W. Zielonka, "Infinite games on finitely coloured graphs with applications to automata on infinite trees," *Theoretical Computer Science*, vol. 200, no. 1-2, pp. 135–183, 1998.

[22] J. H. Reif, "Universal games of incomplete information," in *Proceedings of the eleventh annual ACM symposium on theory of computing*, pp. 288–308, 1979.

[23] K. Chatterjee, L. Doyen, H. Gimbert, and T. A. Henzinger, "Randomness for free," in *Mathematical Foundations of Computer Science 2010: 35th International Symposium, MFCS 2010, Brno, Czech Republic, August 23-27, 2010. Proceedings 35*, pp. 246–257, Springer, 2010.

[24] K. Chatterjee, L. Doyen, S. Nain, and M. Y. Vardi, "The complexity of partial-observation stochastic parity games with finite-memory strategies," in *International Conference on Foundations of Software Science and Computation Structures*, pp. 242–257, Springer, 2014.

[25] S. Nain and M. Y. Vardi, "Solving partial-information stochastic parity games," in *2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science*, pp. 341–348, IEEE, 2013.

[26] V. Gripon and O. Serre, "Qualitative concurrent stochastic games with imperfect information," in *International Colloquium on Automata, Languages, and Programming*, pp. 200–211, Springer, 2009.

[27] P. G. Bennett, "Toward a theory of hypergames," *Omega*, vol. 5, no. 6, pp. 749–751, 1977.

[28] J. C. Harsanyi, "Games with incomplete information played by "bayesian" players, i–iii part i. the basic model," *Management Science*, vol. 14, no. 3, pp. 159–182, 1967.

[29] J. F. Mertens and S. Zamir, "Formulation of bayesian analysis for games with incomplete information," *International journal of game theory*, vol. 14, pp. 1–29, 1985.

[30] R. J. Aumann, M. Maschler, and R. E. Stearns, *Repeated games with incomplete information*. MIT press, 1995.

[31] J. C. Harsanyi, "Games with incomplete information played by "bayesian" players part ii. bayesian equilibrium points," *Management Science*, vol. 14, no. 5, pp. 320–334, 1968.

[32] G. Bonanno, "Agm-consistency and perfect bayesian equilibrium. part i: definition and properties," *International Journal of Game Theory*, vol. 42, pp. 567–592, 2013.

[33] M. Wang, K. W. Hipel, and N. M. Fraser, "Solution concepts in hypergames," *Applied Mathematics and Computation*, vol. 34, no. 3, pp. 147–171, 1989.

[34] R. R. Vane and P. E. Lehner, "Using hypergames to increase planned payoff and reduce risk," *Autonomous Agents and Multi-Agent Systems*, vol. 5, pp. 365–380, 2002.

[35] N. S. Kovach and G. B. Lamont, "Trust and deception in hypergame theory," in *2019 IEEE National Aerospace and Electronics Conference (NAECON)*, pp. 262–268, IEEE, 2019.

[36] D. M. Kilgour, K. W. Hipel, and L. Fang, "The graph model for conflicts," *Autom.*, vol. 23, pp. 41–55, 1987.

[37] J. T. House and G. V. Cybenko, "Hypergame theory applied to cyber attack and defense," in *Defense + Commercial Sensing*, 2010.

[38] B. L. Slantchev, "The principle of convergence in wartime negotiations," *American Political Science Review*, vol. 97, pp. 621 – 632, 2003.

[39] C. N. Gutierrez, S. Bagchi, H. Mohammed, and J. Avery, "Modeling deception in information security as a hypergame–a primer," in *Proceedings of the 16th Annual Information Security Symposium*, p. 41, CERIAS-Purdue University, 2015.

[40] N. S. Kovach, "A temporal framework for hypergame analysis of cyber physical systems in contested environments," 2016.

[41] B. Gharesifard and J. Cortes, "Evolution of Players' Misperceptions in Hypergames Under Perfect Observations," *IEEE Transactions on Automatic Control*, vol. 57, pp. 1627–1640, July 2012.

[42] B. Gharesifard and J. Cortés, "Stealthy deception in hypergames under informational asymmetry," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 6, pp. 785–795, 2013.

[43] Y. Sasaki, *Subjective rationalizability in hypergames*. Hindawi Publishing Corporation, 2014.

[44] S. Zamir, *Bayesian games: Games with incomplete information*. Springer, 2020.

[45] S. Morris, "The common prior assumption in economic theory," *Economics and Philosophy*, vol. 11, pp. 227 – 253, 1995.

[46] J. Y. Halpern, "Characterizing the common prior assumption," *Microeconomic Theory eJournal*, 1998.

[47] F. Araujo, K. W. Hamlen, S. Biedermann, and S. Katzenbeisser, "From patches to honey-patches: Lightweight attacker misdirection, deception, and disinformation," in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pp. 942–953, 2014.

[48] Y. Sasaki and K. Kijima, "Hierarchical hypergames and bayesian games: A generalization of the theoretical comparison of hypergames and bayesian games considering hierarchy of perceptions," *Journal of Systems Science and Complexity*, vol. 29, pp. 187–201, 2016.

[49] N. M. Fraser and K. W. Hipel, "Solving complex conflicts," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 12, pp. 805–816, 1979.

[50] K. W. Hipel and N. Fraser, *Conflict analysis*. North-Holland, 1990.

[51] K. Kijima, "Intelligent poly-agent learning model and its application," *Information and Systems Engineering*, vol. 2, pp. 47–61, 1996.

[52] Y. Sasaki, N. Kobayashi, and K. Kijima, "Mixed extension of hypergames and its applications to inspection games," in *Proceedings of the 51st Annual Meeting of the ISSS-2007, Tokyo, Japan*, 2007.

[53] J. Dubra, F. Maccheroni, and E. A. Ok, "Expected utility theory without the completeness axiom," *Journal of Economic Theory*, vol. 115, no. 1, pp. 118–133, 2004.

[54] J. A. Baier and S. A. McIlraith, "Planning with preferences," *AI Mag.*, vol. 29, pp. 25–36, 2008.

[55] M. Bienvenu, C. Fritz, and S. A. McIlraith, "Specifying and computing preferred plans," *Artificial Intelligence*, vol. 175, no. 7-8, pp. 1308–1345, 2011.

[56] J. Tumova, G. C. Hall, S. Karaman, E. Frazzoli, and D. Rus, "Least-violating control strategy synthesis with safety rules," in *Proceedings of the 16th international conference on Hybrid systems: computation and control*, pp. 1–10, 2013.

[57] T. Wongpiromsarn, K. Slutsky, E. Frazzoli, and U. Topcu, "Minimum-violation planning for autonomous systems: Theoretical and practical considerations," in *2021 American Control Conference (ACC)*, pp. 4866–4872, IEEE, 2021.

[58] H. Rahmani and J. M. O'Kane, "What to do when you can't do it all: Temporal logic planning with soft temporal logic constraints," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6619–6626, IEEE, 2020.

[59] N. Mehdipour, C.-I. Vasile, and C. Belta, "Specifying user preferences using weighted signal temporal logic," *IEEE Control Systems Letters*, vol. 5, no. 6, pp. 2006–2011, 2020.

[60] M. Lahijanian and M. Kwiatkowska, "Specification revision for markov decision processes with optimal trade-off," *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 7411–7418, 2016.

[61] M. Li, A. Turrini, E. M. Hahn, Z. She, and L. Zhang, "Probabilistic preference planning problem for markov decision processes," *IEEE transactions on software engineering*, 2020.

[62] J. Fu, "Probabilistic planning with preferences over temporal goals," *2021 American Control Conference (ACC)*, pp. 4854–4859, 2021.

[63] R. Nau, "The shape of incomplete preferences," 2006.

[64] E. A. Ok *et al.*, "Utility representation of an incomplete preference relation," *Journal of Economic Theory*, vol. 104, no. 2, pp. 429–449, 2002.

[65] S. O. Hansson and T. Grüne-Yanoff, "Preferences," in *The Stanford Encyclopedia of Philosophy* (E. N. Zalta, ed.), Metaphysics Research Lab, Stanford University, Spring 2022 ed., 2022.

[66] A. Sen, "Maximization and the act of choice," *Econometrica*, vol. 65, 1997.

[67] J. J. Thomson, "Killing, letting die, and the trolley problem.," *The Monist*, vol. 59 2, pp. 204–17, 1976.

[68] A. N. Kulkarni, H. Luo, N. O. Leslie, C. A. Kamhoua, and J. Fu, "Deceptive labeling: hypergames on graphs for stealthy deception," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 977–982, 2020.

[69] A. N. Kulkarni, M. S. Cohen, C. A. Kamhoua, and J. Fu, "Integrated resource allocation and strategy synthesis in safety games on graphs with deception," 2023.

[70] A. N. Kulkarni and J. Fu, "Synthesis of deceptive strategies in reachability games with action misperception," 2020.

[71] A. Kulkarni and J. Fu, "Opportunistic synthesis in reactive games under information asymmetry," *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 5323–5329, 2019.

[72] L. Li, H. Ma, A. N. Kulkarni, and J. Fu, "Dynamic hypergames for synthesis of deceptive strategies with temporal logic objectives (under review)," 2020.

[73] A. Kulkarni and J. Fu, "Opportunistic qualitative planning in stochastic systems with incomplete preferences over reachability objectives," *2023 American Control Conference (ACC)*, pp. 3541–3547, 2022.

[74] B. Gharesifard and J. Cortés, "Stealthy Deception in Hypergames Under Informational Asymmetry," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, pp. 785–795, June 2014.

[75] K. Chatterjee, L. Doyen, T. A. Henzinger, and J.-F. Raskin, "Algorithms for omega-regular games with imperfect information," *Logical Methods in Computer Science*, vol. 3, 2007.

[76] E. Filiot, N. Jin, and J.-F. Raskin, "Antichains and compositional algorithms for ltl synthesis," *Formal Methods in System Design*, vol. 39, pp. 261–296, 2011.

[77] A. N. Kulkarni and J. Fu, "A Compositional Approach to Reactive Games under Temporal Logic Specifications," in *American Control Conference*, pp. 2356–2362, IEEE, 2018.

[78] C. Baier and J.-P. Katoen, *Principles of model checking*. MIT press, 2008.

[79] Z. Manna and A. Pnueli, "A hierarchy of temporal properties (invited paper, 1989)," in *Proceedings of the ninth annual ACM symposium on Principles of distributed computing*, pp. 377–410, 1990.

[80] R. Píbil, V. Lisỳ, C. Kiekintveld, B. Bošanskỳ, and M. Pěchouček, "Game theoretic model of strategic honeypot selection in computer networks," in *International Conference on Decision and Game Theory for Security*, pp. 201–220, Springer, 2012.

[81] C. Kiekintveld, M. Jain, J. Tsai, J. Pita, F. Ordóñez, and M. Tambe, "Computing optimal randomized resource allocations for massive security games," in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pp. 689–696, 2009.

[82] K. E. Heckman, F. J. Stech, R. K. Thomas, B. Schmoker, and A. W. Tsow, "Cyber denial, deception and counter deception," *Advances in Information Security*, vol. 64, 2015.

[83] W. Bai and J. Bilmes, "Greed is still good: maximizing monotone submodular+ supermodular (bp) functions," in *International Conference on Machine Learning*, pp. 304–313, PMLR, 2018.

[84] R. A. Rosenbaum, "Sub-additive functions," *Duke Mathematical Journal*, vol. 17, no. 3, pp. 227 – 247, 1950.

[85] E. Hille and R. S. Phillips, *Functional analysis and semi-groups*, vol. 31. American Mathematical Soc., 1996.

[86] V. V. Vazirani, "Approximation algorithms," *Approximation Algorithms*, 2001.

[87] S. Jajodia, V. Subrahmanian, V. Swarup, and C. Wang, *Cyber Deception*. Springer, 2016.

[88] J. Bernet, D. Janin, and I. Walukiewicz, "Permissive strategies: from parity games to safety games," *RAIRO-Theoretical Informatics and Applications-Informatique Théorique et Applications*, vol. 36, no. 3, pp. 261–275, 2002.

[89] V. Švábenský, P. Čeleda, J. Vykopal, and S. Brišáková, "Cybersecurity knowledge and skills taught in capture the flag challenges," *Computers & Security*, vol. 102, p. 102154, 2021.

[90] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons Inc., 2005.

[91] S. Myagmar, A. J. Lee, and W. Yurcik, "Threat modeling as a basis for security requirements," in *Symposium on requirements engineering for information security (SREIS)*, vol. 2005, pp. 1–8, Citeseer, 2005.

[92] D. Fudenberg and J. Tirole, *Game theory*. 1991.

[93] J. K. Goeree, C. A. Holt, and T. R. Palfrey, "Stochastic game theory for social science: A primer on quantal response equilibrium," *Handbook of Experimental Game Theory*, pp. 8–47, 2020.

[94] M. Kwiatkowska, G. Norman, and D. Parker, "Stochastic model checking," in *Formal Methods for the Design of Computer, Communication and Software Systems: Performance Evaluation (SFM'07)* (M. Bernardo and J. Hillston, eds.), vol. 4486 of *LNCS (Tutorial Volume)*, pp. 220–270, Springer, 2007.

[95] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM 4.0: Verification of probabilistic real-time systems," in *Proc. 23rd International Conference on Computer Aided Verification (CAV'11)* (G. Gopalakrishnan and S. Qadeer, eds.), vol. 6806 of *LNCS*, pp. 585–591, Springer, 2011.

[96] L. Li and J. Fu, "Topological approximate dynamic programming under temporal logic constraints," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 5330–5337, Dec 2019.

[97] A. M. Polansky, "Detecting change-points in markov chains," *Computational statistics & data analysis*, vol. 51, no. 12, pp. 6013–6026, 2007.

[98] M. Basseville, I. V. Nikiforov, *et al.*, *Detection of abrupt changes: theory and application*, vol. 104. Prentice Hall Englewood Cliffs, 1993.

[99] M. L. Littman, T. L. Dean, and L. P. Kaelbling, "On the complexity of solving markov decision problems," *arXiv preprint arXiv:1302.4971*, 2013.

[100] D. Bouyssou, D. Dubois, and M. Pirlot, *Concepts & Methods of Decision-Making*. John Wiley & Sons Inc., 2009.

[101] S. O. Hansson, *The structure of values and norms*. Cambridge University Press, 2001.

[102] J. E. Hopcroft, R. Motwani, and J. D. Ullman, "Introduction to automata theory, languages, and computation," *Acm Sigact News*, vol. 32, no. 1, pp. 60–65, 2001.

[103] G. R. Santhanam, S. Basu, and V. Honavar, *Representing and reasoning with qualitative preferences: Tools and applications*. Springer, 2016.

[104] M. Kloetzer and C. Belta, "A fully automated framework for control of linear systems from temporal logic specifications," *IEEE Transactions on Automatic Control*, vol. 53, no. 1, pp. 287–297, 2008.

[105] K. Chatterjee and T. A. Henzinger, "A survey of stochastic ω-regular games," *Journal of Computer and System Sciences*, vol. 78, no. 2, pp. 394–413, 2012.

BIOGRAPHICAL SKETCH

Abhishek Ninad Kulkarni received his bachelor's degree in electrical engineering from Vishwakarma Institute of Technology, Pune, India, in 2012 and Master of Science in robotics engineering from Worcester Polytechnic Institute, Worcester, MA, USA, in 2021. He received his Ph.D. degree from the ECE Department at the University of Florida in 2023. His research interests include game theory, formal methods with applications to sequential decision-making in robotics, and cyber-physical systems security.