Reachability-Based Planning for Safe Navigation Under Adversarial Pursuit

EEL4930- Formal Methods in Robotics and AI

Erik Bloomquist
Dept. of ECE
University of Florida
Gainesville, Florida, USA
erikbloomquist@ufl.edu

Adam Benali
Dept. of CISE
University of Florida
Gainesville, Florida, USA
adam.benali@ufl.edu

Rodrigo Bazan
Dept. of ECE
University of Florida
Gaineseville, Florida , USA
rodrigo.bazan@ufl.edu

***Abstract*—This work investigates formal-methods-based planning for an autonomous agent (Jerry) navigating a stochastic gridworld while avoiding an adversarial agent (Tom) and fixed environmental hazards. We express the task using the Linear Temporal Logic (LTL) specification $\varphi = \mathbf{G}\neg\text{unsafe} \wedge \mathbf{F}\,\text{goal}$ and enforce it through a reachability-MDP formulation in which unsafe states are modeled as zero-valued absorbing states and goal states as unit-valued absorbing states. Value iteration then computes the maximum probability of satisfying φ from every joint agent state.**

We compare two adversary models—uniform random movement and a Manhattan-distance heuristic pursuit model—and analyze their effects on Jerry's success probability, spatial occupancy distributions, and joint agent-state structure. The heuristic pursuit adversary substantially *reduces* Jerry's success probability, as Tom's directed motion constrains safe navigation routes despite being more predictable. Monte Carlo simulation further reveals concentrated spatial patterns and strong correlations in the joint state distribution that emerge only under heuristic pursuit. These results demonstrate how incorporating formal safety specifications and explicit adversarial modeling shapes optimal behavior and safety guarantees in stochastic multi-agent environments.

***Index Terms*—Markov Decision Processes, Temporal Logic, Safety Constraints, Stochastic Planning, Value Iteration.**

I. Introduction

In modern autonomous systems, it is vital that an agent reach target states without ever entering an unsafe state. However, these autonomous systems often exhibit stochastic dynamics, making it necessary to model this behavior in simulation before deployment. For example, autonomous vehicles, such as self-driving cars, often suffer incorrect movements due to sensor errors [1]. Typical reinforcement learning strategies aim to rectify this issue but fall short given their concern with only maximizing the cumulative discounted reward. These reinforcement learning strategies reward the agent for reaching its goal states and penalize the agent for entering an unsafe state. The problem that arises is that in the case of self-driving cars, it is imperative that the car *never* go on the other side of the road, even if it ultimately results in reaching the goal. In addition, self-driving cars need to keep track of other stochastic systems such as pedestrians that may enter the street that the car is driving on.

This project aims to investigate this issue by utilizing a Markov Decision Process (MDP) to model the system's stochastic behavior and encode goal regions and unsafe regions within this framework. Furthermore, the MDP will contain an antagonistic agent that follows various movement policies. This project aims to (1) model this sample system and (2) compute an optimal policy in an MDP that (3) maximizes the probability of reaching a goal state while (4) minimizing the probability of entering an unsafe state (including the state occupied by the opposing agent). Unsafe states are treated as zero-valued absorbing states, which enforces the safety requirement through boundary conditions rather than through reward shaping.

II. Preliminaries and Problem Statement

We consider a gridworld environment modeled as a Markov Decision Process (MDP) with stochastic transitions in which a mouse (Jerry) has the goal of navigating toward a piece of cheese while avoiding traps and an adversarial agent (Tom) [2]. Jerry's transition dynamics are subject to uncertainty and may cause him to enter unsafe states despite attempting to move safely. We experiment with two pursuit policies for Tom: a heuristic model in which Tom moves toward Jerry with probability 0.7 (and moves randomly otherwise), and a completely random movement model. As a result, Jerry must employ a control policy that maximizes his probability of reaching a piece of cheese while avoiding all hazards, including the cell currently occupied by Tom. To express these safety and goal constraints, we use the temporal-logic specification

$$\varphi = \mathbf{G}\neg\text{hazard} \ \wedge \ \mathbf{F}\,\text{cheese}.$$

Our central problem revolves around the pursuit of a policy where the constraints of "always stay safe" and "eventually reach the cheese" can be formulated for the gridworld MDP. The combination of Linear Temporal Logic and MDP principles will be necessary to answer our question: *Can we compute*

a policy such that Jerry avoids traps and Tom with maximal probability? If so, what is the probability that Jerry will reach a piece of cheese?

To model this problem in the grid, we begin by defining the different movements in the MDP. Both Tom and Jerry are able to move north, south, east, and west, allowing them to move into adjacent squares. The traps and cheese remain stationary throughout the entire simulation. At each time step, both Jerry and Tom move to their next location, with Jerry moving first at every step. Within the MDP, trap states and states in which Jerry and Tom occupy the same cell are treated as absorbing terminal states with value 0. This does not prevent Jerry from transitioning into these states due to stochastic movement, but any trajectory that enters them immediately terminates with zero success probability. Likewise, cheese states are treated as absorbing terminal states with value 1.

Rather than defining a reward function, we solve a reachability problem: cheese cells are assigned $V = 1$, hazard states are assigned $V = 0$, and all other states are initialized with $V = 0$ and updated using the value iteration algorithm until convergence of the Bellman equation. This dynamic programming method updates the value of each state until the system converges to the maximal probability of eventually reaching a goal state without hitting any hazard.

Although we do not build an explicit automaton product, this setup implements the standard formal-methods reduction of the LTL formula φ to a probabilistic reachability problem. Assigning $V = 1$ to goal states and $V = 0$ to hazards encodes the semantics of $\mathbf{G}\neg\text{hazard} \wedge \mathbf{F}\,\text{cheese}$, and the Bellman operator computes the maximal satisfaction probability over all policies. Thus, our solution constitutes formal LTL-constrained policy synthesis for an MDP.

III. Methods

A. Approach and Problem Formulation

We formulate Jerry's navigation problem as a reachability analysis over a joint MDP with state space $\mathcal{S} = J \times T$ (625 states), where J and T denote Jerry's and Tom's positions on the 5×5 grid. The objective encodes the LTL specification

$$\varphi = \mathbf{G}\neg\text{hazard} \ \wedge \ \mathbf{F}\,\text{goal},$$

which we implement through boundary conditions: goal states (cheese cells) are assigned $V(s) = 1$, while hazard states (traps or $j = t$) are assigned $V(s) = 0$. Both hazard and cheese states are treated as absorbing terminal states. All remaining (neutral) states are initialized with $V(s) = 0$ but are *not* terminal; their values are updated iteratively by the Bellman equation until convergence.

We solve the undiscounted Bellman reachability equation

$$V(s) = \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P(s' \mid s, a)\, V(s')$$

using synchronous value iteration until $\|\Delta V\|_\infty < 10^{-7}$. The resulting value function $V(s)$ gives the probability of eventually reaching a cheese cell from state s without ever entering a hazard.

This formulation enforces hard safety constraints: any trajectory that reaches a hazard state has success probability zero. Rather than penalizing hazards with negative rewards, we directly encode the LTL constraint by modeling hazards as absorbing zero-valued states, eliminating the need for reward shaping or tuning safety–performance tradeoffs.

B. Agent Dynamic and Transition Model

Jerry selects actions from $\mathcal{A} = \{\text{North, South, East, West}\}$ and moves stochastically: with probability 0.8 in the intended direction, 0.1 perpendicular-left, and 0.1 perpendicular-right. Attempted moves off the grid leave the agent in place.

Tom also moves each step, and the joint state $s = (j, t)$ is terminal if j is a trap cell or if $j = t$ (collision). We consider two Tom policies: (1) *Random surveillance*, where Tom selects uniformly among the four actions, and (2) *Heuristic pursuit* with parameter $p_{chase} = 0.7$, where Tom moves to reduce Manhattan distance to Jerry with probability 0.7 and moves randomly otherwise. Ties in the greedy heuristic are broken uniformly.

Transitions factorize as

$$P((j', t') \mid (j, t), a) = P_J(j' \mid j, a)\, P_T(t' \mid t, j),$$

allowing transition probabilities to be computed on-the-fly. This factorization captures Jerry's action-dependent uncertainty and Tom's reactive or random behavior while avoiding dense storage of the full 625×625 transition matrix.

C. Solution Method

We initialize $V(s) = 1$ for all goal states (those with j in the cheese set), $V(s) = 0$ for all hazard states (those with j in a trap or $j = t$), and $V(s) = 0$ for all other states prior to iteration. At each iteration k, we update all non-terminal states using the Bellman operator

$$V^{(k+1)}(s) \leftarrow \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P(s' \mid s, a)\, V^{(k)}(s'),$$

computing transition probabilities on-the-fly by enumerating Jerry's stochastic motion and Tom's policy-dependent moves. We iterate until $\|V^{(k+1)} - V^{(k)}\|_\infty < \varepsilon = 10^{-7}$, which occurs in 76–88 iterations depending on Tom's policy.

Once converged, we extract the greedy policy

$$\pi^*(s) = \arg\max_{a \in \mathcal{A}} \sum_{s'} P(s' \mid s, a)\, V(s'),$$

with ties broken arbitrarily.

Because the optimal policy depends on the full joint state (j, t), we do not present a marginalized single-agent policy visualization, as such summaries can obscure critical safety structure in the value function.

D. Experimental Process

We solve the joint MDP twice—once with `tom_policy = "random"` and once with `tom_policy = "heuristic"` (with $p_{\text{chase}} = 0.7$)—to obtain the value functions V_{random} and $V_{\text{heuristic}}$. For each solution, we aggregate Jerry's start-cell success probabilities by marginalizing V

over a uniform prior on Tom's initial position, producing two 5×5 grids of success probabilities. These are visualized side-by-side along with a difference heatmap

$$\Delta = \text{grid}(V_{\text{random}}) - \text{grid}(V_{\text{heuristic}}).$$

To analyze state-visitation patterns, we simulate 10,000 episodes for each Tom policy. Each episode begins from a uniformly sampled non-terminal joint state (j, t) and proceeds for at most 100 steps or until a terminal state is reached. At each step, Jerry's next position is sampled from P_J and Tom's from P_T according to the corresponding policy, and we record visit counts for both agents. These counts are normalized into probability distributions over the 25 grid cells and visualized as 2D heatmaps.

Additionally, we estimate the joint distribution $P(j, t)$ by tallying visits to each of the 625 joint states across all episodes with uniformly random spawns. After normalization, the resulting 25×25 joint occupancy matrix is rendered as a heatmap, revealing correlations between the agents' locations.

This experimental design isolates the effect of Tom's behavior on Jerry's optimal strategy and success probability while providing complementary views of the resulting equilibrium dynamics.

IV. Contributions

Erik wrote the initial code for the Tom and Jerry simulation using a random movement policy for Tom. He ran experiments and provided preliminary results for the slide presentation. Rodrigo focused on implementing the heuristic pursuit policy for Tom. He developed the basis for visualizing the data via MatPlotLib and gathered results on the random vs. heuristic policy for Tom. With simulations completed, Erik collected data, analyzed results, and created figures for the report.

Adam was responsible for writing the "Introduction" and "Preliminaries and Problem Statement" sections of the report. Rodrigo composed the "Methods" section, and Erik authored the "Results and Analysis" and "Discussion and Conclusions" portions.

V. Results and Analysis

A. Optimal Success Probabilities Under Each Tom Policy

We first computed the reachability value function for both adversary models. Figures 1 and 2 show the optimal success probability

$$V^*(s) \;=\; \max_{a \in \mathcal{A}} \mathbb{E}[\, V^*(s') \,],$$

for Jerry under Random Tom and Heuristic Tom, respectively.

Across most of the grid, Jerry maintains a high probability of eventually reaching a cheese cell while avoiding traps or collision with Tom. However, averaging over possible initial states reveals a clear difference between the two adversaries:

- **Average Jerry success under Random Tom:** 0.865
- **Average Jerry success under Heuristic Pursuit Tom:** 0.841

As one might expect from a less intelligent adversary, Jerry performs better when Tom moves randomly. The heuristic policy causes Tom to move toward Jerry more aggressively, shrinking safe regions and reducing the number of viable escape routes. As a result, Jerry's optimal success probability decreases under the pursuit model.

This trend is confirmed by the difference heatmap in Fig. 3, which plots

$$\Delta(j) \;=\; V_{\text{random}}(j) - V_{\text{heuristic}}(j).$$

Most cells exhibit positive values, indicating that for the majority of starting locations, Jerry has a higher success probability when Tom follows the Random Surveillance policy.

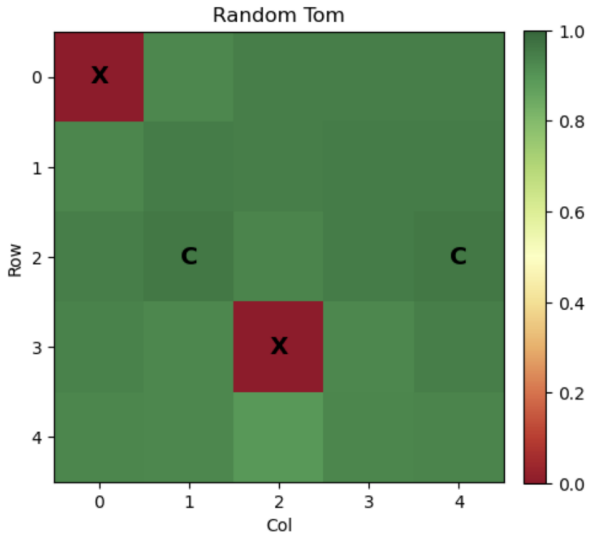

Fig. 1. Optimal success probability for Jerry under Random Tom.

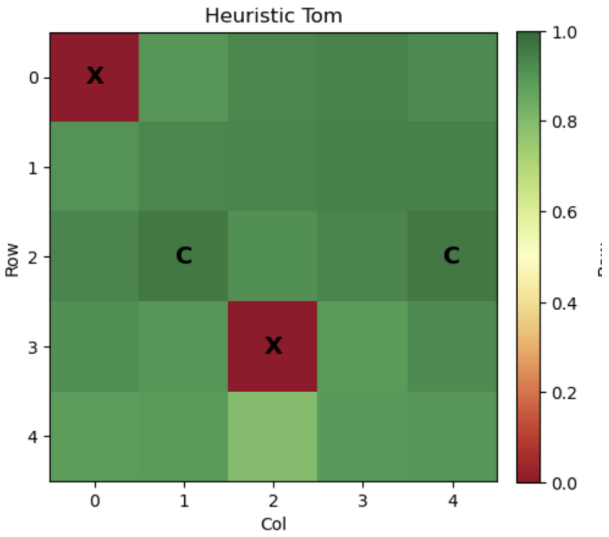

Fig. 2. Optimal success probability for Jerry under Heuristic Tom ($p_{\text{chase}} = 0.7$).

B. Marginal Spatial Behavior of Jerry and Tom

To understand how the optimal controller behaves under different adversary models, we examined marginal state–

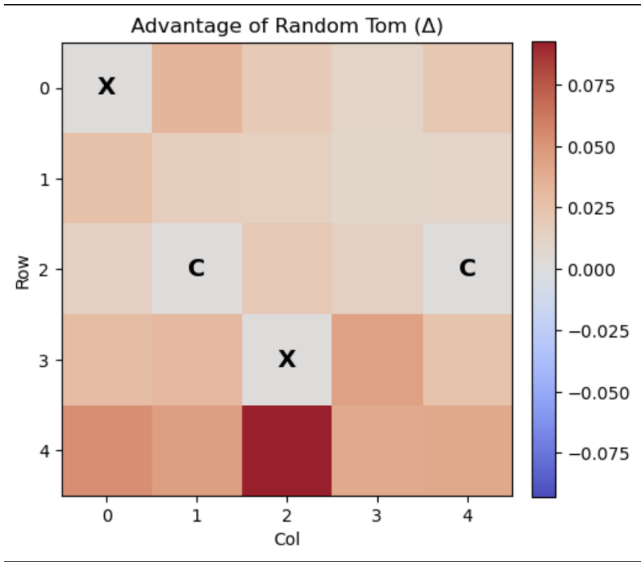

Fig. 3. Difference in success probability: $V_{\text{random}} - V_{\text{heuristic}}$. Larger values indicate states where random movement benefits Tom more.

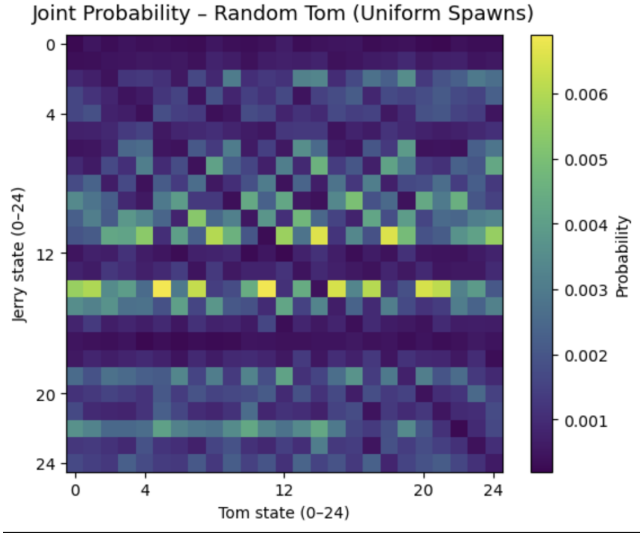

Fig. 4. Joint probability heatmap $P(J = j, T = t)$ under Random Tom.

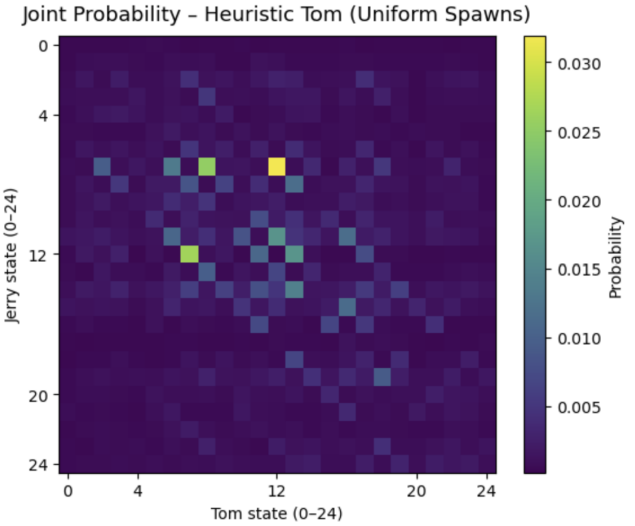

Fig. 5. Joint probability heatmap $P(J = j, T = t)$ under Heuristic Tom.

occupancy distributions for both agents across 10,000 Monte Carlo episodes for each Tom policy. Figures 6–9 show the resulting 5×5 spatial probability maps for Jerry and Tom.

Under both Tom models, Jerry tends to move through the center of the grid rather than the corners or trap-adjacent regions. However, the differences between the two adversary behaviors are significant:

- **Random Tom** produces an almost uniform distribution over the grid, consistent with an unbiased random walk.
- **Heuristic Pursuit Tom** concentrates strongly in the central region, repeatedly moving to reduce Manhattan distance to Jerry. This reactive behavior creates a pronounced spatial "hot zone" where Tom is most frequently present.

Jerry adapts to this difference in threat geometry. Under Heuristic Tom, Jerry's occupancy distribution shifts *away* from the central region and toward peripheral safe states, reflecting the need to avoid Tom's highly predictable pursuit trajectory. These patterns illustrate that Jerry's optimal controller is shaped more by *avoiding hazards* than by directly pursuing the cheese: progress toward the goal is achieved only when it does not increase collision risk.

C. Joint Distribution of (Jerry, Tom) States

While marginal heatmaps reveal each agent's behavior individually, the interaction between them is most clearly characterized by the *joint* state distribution

$$P(J = j,\ T = t).$$

Figures 4 and 5 show the resulting 25×25 joint occupancy matrices estimated from Monte Carlo simulation.

Under Random Tom, the joint distribution is diffuse and largely unstructured, reflecting the weak coupling between the agents—Tom's motion is independent of Jerry's trajectory, and their co-location patterns arise almost entirely from stochastic overlap.

Under Heuristic Pursuit Tom, the interaction becomes highly structured. Strong probability mass appears along diagonal bands where Tom trails Jerry at short Manhattan distance, revealing Tom's tendency to close distance whenever possible. This concentrated pursuit region sharply reduces the number of safe paths available to Jerry.

These structured interactions explain the empirical result that heuristic pursuit yields *lower* success probability for Jerry: Tom's directed behavior reduces escape options and increases collision likelihood, making the adversary significantly more effective than Random Tom.

VI. Discussion and Conclusions

A. Discussion of Findings

Several conclusions emerge from the analyses presented above:

1) **Heuristic pursuit decreases Jerry's success probability.** Tom's directed motion sharply reduces the number of safe routes available to Jerry, resulting in lower reachability values compared to the Random Surveillance model.
2) **Jerry exploits safe regions while ensuring progress by construction.** Because unsafe states are absorbing with value zero, the optimal policy necessarily avoids traps and collisions. As a result, Jerry's controller navigates through states that preserve safety guarantees while still enabling eventual progress toward a cheese cell.
3) **Tom's pursuit creates strong spatial and joint-state structure.** Under heuristic pursuit, Tom repeatedly moves toward Jerry, producing concentrated occupancy in central cells and structured diagonal bands in the joint distribution. This coupling between the agents' states is absent under random motion.
4) **Joint distributions provide the clearest window into agent interaction.** While marginal heatmaps reveal each agent's tendencies in isolation, the joint $P(J, T)$ heatmaps expose the geometry of the pursuit itself and explain why heuristic pursuit is a more effective adversarial strategy.

Taken together, these findings show that adversarial intelligence can strongly influence reachability outcomes: increasing Tom's strategic behavior indeed makes the task more difficult for Jerry. The value functions, occupancy distributions, and joint-state analyses all converge on this interpretation.

B. Lessons Learned and Future Work

This project highlighted several key insights. First, encoding the safety requirement $\mathbf{G}\neg$unsafe directly through absorbing zero-valued hazard states made safety a *hard constraint*, not something negotiated through rewards or penalties. Although we did not explicitly construct a product MDP, the reachability formulation mirrors formal-methods reasoning: unsafe trajectories are provably eliminated, and the value function corresponds exactly to the satisfaction probability of the LTL formula

$$\mathbf{G}\neg\text{unsafe} \ \wedge \ \mathbf{F}\,\text{goal}.$$

Second, the joint-state analyses revealed that Tom's heuristic pursuit strategy creates predictable but highly dangerous spatial structure. Rather than helping Jerry, this structure constrains escape routes and lowers success probability. The joint heatmaps were essential in diagnosing this effect and understanding how pursuit behavior changes the geometry of safe navigation.

Third, simulated occupancy distributions provided an empirical counterpart to the reachability solution, illustrating which states the optimal controller actually visits under realistic rollouts. These complementary perspectives, analytical reachability and empirical simulation, produced a coherent understanding of the system's behavior.

Future work includes extending Tom's behavior to incorporate probabilistic prediction, game-theoretic strategies, or partial observability; scaling to larger grids where exact value iteration becomes computationally difficult; and exploring robust or learning-based policies that maintain formal safety guarantees even when Tom's behavior deviates from its assumed model.

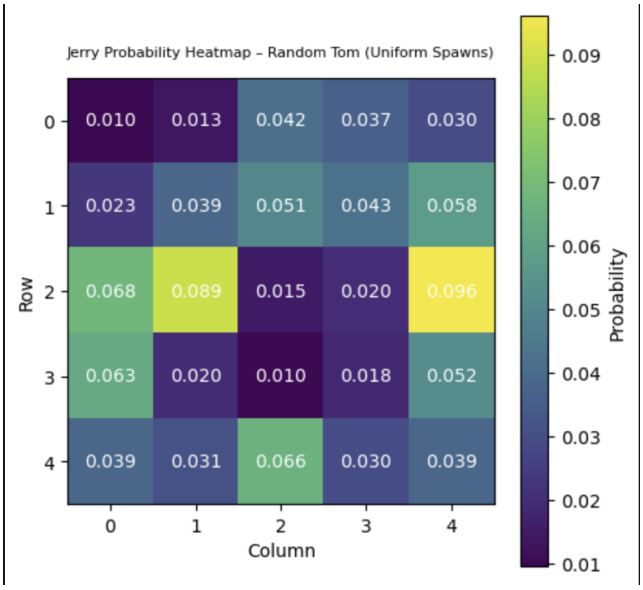

Fig. 6. Spatial probability distribution for Jerry under Random Tom.

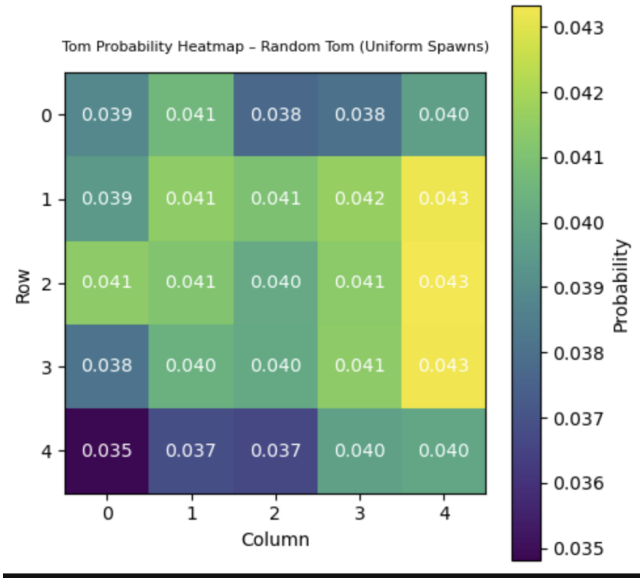

Fig. 7. Spatial probability distribution for Random Tom.

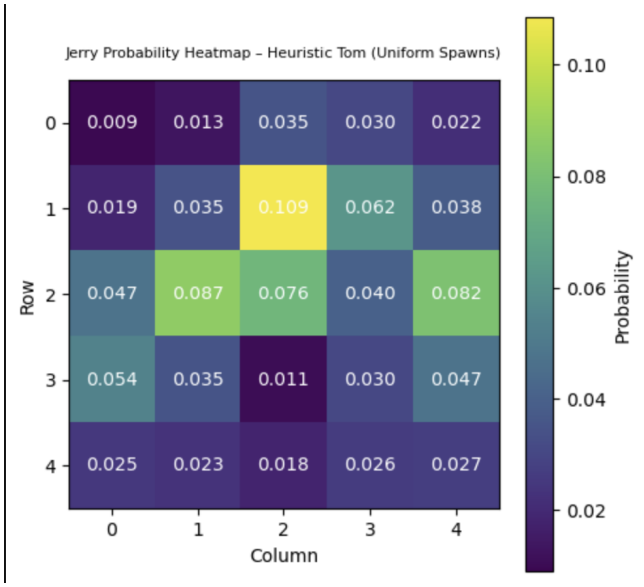

Fig. 8. Spatial probability distribution for Jerry under Heuristic Pursuit Tom.

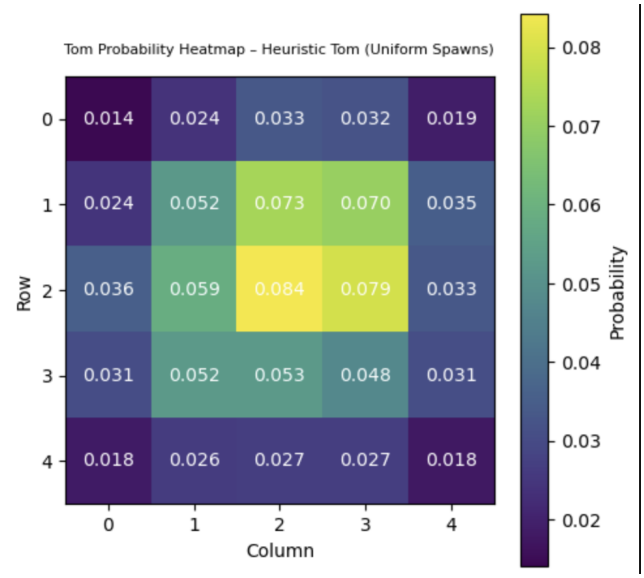

Fig. 9. Spatial probability distribution for Heuristic Pursuit Tom.

REFERENCES

[1] S. Goswami and S. Mondal, "Stochastic models for autonomous systems and robotics," *Spectrum of Operational Research*, vol. 3, pp. 215–237, 05 2025.

[2] J. Fu, "Final project ideas," Canvas Course Materials, 2024, [Online]. Available: https://uflorida-my.sharepoint.com/:p:/g/personal/fujie_ufl_edu/IQD6DgjXMyqiRIXF35p8VuRuAcOpq_sgVwom8gD6fVCU_10?e=Sn419t.