Disaster Evacuation Route Planning in Uncertain Environments using Dyna-Q

1st Ionel-Cătălin Butacu
Faculty of Automatic Control and Computer Engineering
Technical University Gheorghe Asachi Iasi
Iași, Romania
ionel-catalin.butacu@student.tuiasi.ro

2nd Radu-Ionuț Bălăiță
Faculty of Automatic Control and Computer Engineering
Technical University Gheorghe Asachi Iasi
Iași, Romania
radu-ionut.balaita@student.tuiasi.ro

***Abstract*—Disaster evacuation planning is a critical challenge in emergency management, requiring robust strategies to guide individuals to safety amidst dynamic and unpredictable hazards. This paper presents a Multi-Agent Reinforcement Learning (MARL) system that utilizes the Dyna-Q algorithm to solve evacuation problems in discrete grid-based environments. The system simulates dynamic hazards such as floods, fires, earthquakes, and tornadoes, creating a stochastic environment where agents must balance immediate survival with efficient route planning. Unlike traditional pathfinding algorithms which assume static terrains, our approach enables agents to adapt to changing hazard boundaries in real-time. Our results demonstrate that model-based reinforcement learning significantly accelerates convergence compared to model-free approaches. In widely simulated scenarios involving 15 agents on a 20×20 grid, the proposed system achieves high survival rates in stochastic flood scenarios, showcasing the effectiveness of Dyna-Q for real-time adaptive evacuation planning.**

***Index Terms*—reinforcement learning, dyna-q, multi-agent systems, evacuation planning, disaster management**

I. Introduction

Disaster evacuation planning is a fundamental aspect of civil safety, yet it remains computationally challenging due to the inherent uncertainty of emergency situations. Traditional pathfinding algorithms, such as A* or Dijkstra, rely on static graph representations of the environment [4]. While optimal in stable conditions, these algorithms fail when the graph connectivity changes dynamically—such as when floodwaters rise, fires spread, or buildings collapse. In such scenarios, a pre-calculated path may lead agents directly into a newly formed hazard. This project addresses this limitation by determining optimal policies using Multi-Agent Reinforcement Learning (MARL). We specifically utilize the **Dyna-Q** algorithm [2], a model-based approach that combines direct experience with simulated planning. This hybrid architecture allows agents to learn optimal policies in uncertain, stochastic environments where hazard dynamics are not known a priori. By maintaining an internal model of the world, agents can ”dream” about future consequences, effectively propagating safety information across the state space without the need for dangerous trial-and-error in the physical environment.

A. Related Work

1) Classical Methods: Simulation of crowd dynamics has traditionally relied on Social Force Models (SFM) [1], which treat individuals as particles subject to physical forces. While effective for modeling panic and congestion, these models often lack high-level decision-making capabilities regarding optimal route selection in complex, changing mazes.

2) Reinforcement Learning: Recent advancements in Deep Reinforcement Learning (DRL), such as Deep Q-Networks (DQN), have shown success in complex, high-dimensional environments [3]. However, these methods often suffer from high sample complexity, requiring millions of interactions to converge, and lack interpretability—a black-box risk in safety-critical applications. In contrast, tabular methods like Dyna-Q provide transparency and significantly faster initial convergence for grid-based problems. Our work distinguishes itself by demonstrating that Dyna-Q's planning phase allows for the rapid propagation of hazard information (e.g., ”this path leads to death”) across the state space, a crucial feature for minimizing casualties during training and deployment.

II. System Architecture

The system is designed as a modular simulation framework composed of three main layers: the Simulation Environment, the Agent Model, and the Interaction Loop.

A. Environment

The environment is modeled as a discrete $N \times N$ grid, representing a simplified urban layout.

- **State Space**: The state is defined by the agent's position $s = (x, y)$. For a grid of size $N = 20$, the observation space size is 400 discrete states.
- **Action Space**: Agents have 4 discrete actions: $A = \{Up, Down, Left, Right\}$. Movement is deterministic unless blocked by an obstacle.
- **Entities**:
  - **Agents**: Independent entities seeking evacuation zones. They possess local observations of the grid.
  - **Hazards**: Dynamic danger zones (Flood, Fire, etc.) that expand according to probabilistic rules.
  - **Obstacles**: Static blocked cells representing walls or debris.

B. Hazard Dynamics

We model four distinct hazard scenarios, each with unique uncertainty patterns, as detailed in Table I. These dynamics challenge the agents to generalize their navigation policies rather than memorizing fixed paths.

TABLE I
HAZARD SCENARIOS AND UNCERTAINTY PATTERNS

Scenario	**Uncertainty Pattern**
Flood	Deterministic spread from corner (unknown timing)
Fire	Stochastic circular spread (random rate)
Earthquake	Random building collapse (highly stochastic)
Tornado	Moving hazard zone (unpredictable path)

III. IMPLEMENTATION DETAILS

We model the evacuation problem as a Markov Decision Process (MDP) defined by the tuple (S, A, P, R, γ).

A. Dyna-Q Algorithm

The core algorithm integrates model learning with direct RL. The complete procedure is outlined in Algorithm 1.

Algorithm 1 Dyna-Q Algorithm for Evacuation

```
Initialize $Q(s,a)$ and $Model(s,a)$ for all $s \in S, a \in A$
Loop for each episode:
  $S \leftarrow$ current (start) state
  Loop until $S$ is terminal:
    $A \leftarrow \epsilon$-greedy$(S, Q)$
    Execute action $A$; observe resultant reward $R$ and state $S'$
    $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma \max_a Q(S',a) - Q(S,A)]$
    $Model(S,A) \leftarrow (R, S')$
    Loop $K$ times (Planning):
      $S_{plan} \leftarrow$ random previously observed state
      $A_{plan} \leftarrow$ random action taken in $S_{plan}$
      $R_{plan}, S'_{plan} \leftarrow Model(S_{plan}, A_{plan})$
      $Q(S_{plan}, A_{plan}) \leftarrow Q(S_{plan}, A_{plan}) + \alpha[R_{plan} + \gamma \max_a Q(S'_{plan}, a) - Q(S_{plan}, A_{plan})]$
    $S \leftarrow S'$
```

B. Configuration and Hyperparameters

The system configuration is critical for performance. Table II outlines the key parameters used in our experiments. We chose a discount factor $\gamma = 0.95$ to prioritize long-term survival over immediate rewards. The planning parameter $K = 50$ was selected empirically to balance computational cost with convergence speed. A learning rate of $\alpha = 0.15$ ensures stable updates without oscilliating.

TABLE II
SYSTEM CONFIGURATION PARAMETERS

Parameter	**Value**	**Description**
GRID_SIZE	20	Map dimensions (20×20)
NUM_AGENTS	15	Total agents to evacuate
NUM_EXITS	1	Safe evacuation zones
PLANNING_STEPS	50	Simulated steps per real step (K)
GAMMA (γ)	0.95	Discount factor
ALPHA (α)	0.15	Learning rate
EPSILON (ϵ)	0.3	Exploration rate (decaying)

C. Reward Structure

The reward function is shaped to guide agents towards optimal behavior:

- **Evacuation (Goal)**: $+100$. This large positive reward is the primary objective function.
- **Hazard/Death**: -50. A severe penalty to discourage hazardous states.
- **Step Cost**: -1. This encourages the agent to find the shortest path to the exit.
- **Obstacle/Wall**: -2. A penalty for colliding with static geometry.
- **Heuristic Shaping**: $+3$ for moving closer to a known exit, -1 for moving away. This dense reward signal helps guide the initial exploration phase.

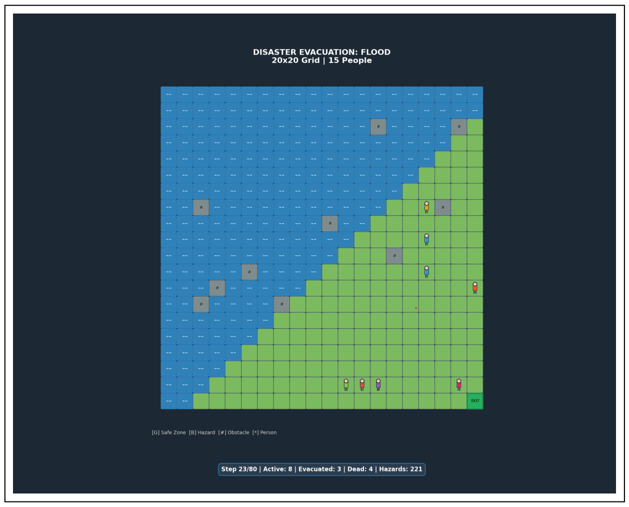
DISASTER EVACUATION: FLOOD
20x20 Grid | 15 People
EXIT
[G] Safe Zone [B] Hazard [#] Obstacle [*] Person
Step 23/80 | Active: 8 | Evacuated: 3 | Dead: 4 | Hazards: 221

Fig. 1. Snapshot of the simulation environment showing agents (colored figures) navigating towards the green exit zone while avoiding the blue spreading flood hazard. Grey blocks represent static obstacles.

IV. EVALUATION AND RESULTS

A. Experimental Setup

We evaluated the system on a 20×20 grid with 15 agents over 5000 training episodes. The primary metric is the **Survival Rate**, defined as the percentage of agents that successfully reach the exit. Agents spawn at random locations in the safe zone at the start of each episode.

B. Comparative Analysis

We conducted a comparative benchmark between Dyna-Q (Model-Based) and Standard Q-Learning (Model-Free) over 500 episodes in the Flood scenario. As illustrated in Figure 2, Dyna-Q demonstrates superior usage of sparse data.

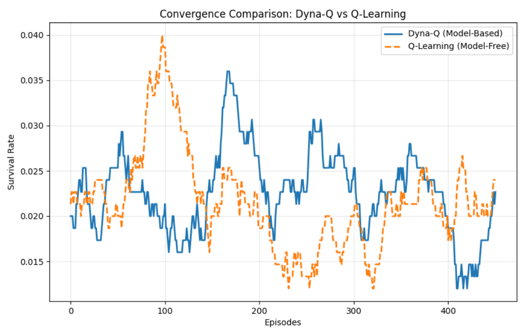

Fig. 2. Survival Rate comparison between Dyna-Q and Q-Learning over 500 training episodes. Dyna-Q (blue) converges faster due to simulated planning.

- **Convergence Speed**: Dyna-Q agents achieve survival considerably faster than Q-Learning agents. This is directly attributable to the Planning steps ($K = 50$).
- **Safety Awareness**: In standard Q-learning, reward propagation is slow; an agent deep in the grid only learns about a distant hazard after many episodes of traversing the entire path. In Dyna-Q, once a single agent experiences a hazard (receiving $R = -50$), the model records this. During the planning phase, this negative value is propagated back to adjacent states almost immediately. Effectively, adjacent states ”learn” they are dangerous before the agent ever visits them again physically.

V. Conclusions

This project validates the efficacy of Dyna-Q for disaster evacuation planning. By integrating model-based planning with model-free learning, agents can learn optimal policies with fewer real-world interactions. The system successfully handles diverse hazard dynamics, demonstrating robust adaptability. The use of a shaped reward function combined with the Dyna-Q planning architecture ensures that agents not only find the shortest path but also adapt to the stochastic nature of the hazards.

References

[1] D. Helbing, I. Farkas, and T. Vicsek, ”Simulating dynamical features of escape panic,” *Nature*, vol. 407, pp. 487–490, 2000.

[2] R. S. Sutton, ”Dyna, an integrated architecture for learning, planning, and reacting,” *ACM SIGART Bulletin*, vol. 2, no. 4, pp. 160–163, 1991.

[3] V. Mnih et al., ”Human-level control through deep reinforcement learning,” *Nature*, vol. 518, pp. 529–533, 2015.

[4] P. E. Hart, N. J. Nilsson, and B. Raphael, ”A Formal Basis for the Heuristic Determination of Minimum Cost Paths,” *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.