EnMod-DP: An Accelerated and Adaptive Multi-Objective Evacuation Navigation Framework for Cyber-Physical Human Environments

G. Balaji
Department of Computer Science and Engineering
Shiv Nadar University
Chennai, India
balajig@snuchennai.edu.in

***Abstract*—Real-time evacuation guidance is a critical safety function for intelligent Cyber-Physical Systems (CPS) in smart buildings and other large-scale physical infrastructures. The core challenge lies in executing complex, multi-objective pathfinding algorithms under stringent real-time constraints. We present EnMod-DP, an accelerated and adaptive framework for evacuation navigation based on a semicontractive Abstract Dynamic Programming (ADP) model. The framework intelligently switches between model-based Dynamic Programming (DP) planners and model-free Reinforcement Learning (RL) policies based on real-time threat assessments, creating a robust hybrid system. We implemented and compared a suite of static, dynamic, and hybrid solvers in a C++ simulation environment featuring dynamic hazards. Through extensive simulations on grids up to 15x15, our results demonstrate that the EnMod-DP hybrid approaches achieve the same optimal path quality as computationally expensive dynamic planners while offering significant improvements in execution time. We further validate our framework with a multi-agent Cyber-Physical System (CPS) simulation, demonstrating its applicability to real-world, multi-agent evacuation scenarios. These findings establish EnMod-DP as a viable and efficient solution for intelligent, scalable, and adaptive evacuation routing systems.**

***Index Terms*—Cyber-Physical Systems, Evacuation Navigation, Dynamic Programming, Reinforcement Learning, Hybrid Systems, Multi-Agent Simulation.**

I. Introduction

The increasing density of urban populations has led to the development of complex Cyber-Physical Human Environments (CPHEs), such as smart buildings and transportation hubs [1], [2]. In the event of an emergency like a fire, providing real-time, optimal evacuation guidance is a life-saving necessity [3]. Traditional pathfinding models often fail to account for the highly dynamic and partially deterministic nature of such crises, where hazards spread and safe exits may change [4], [5].

Dynamic Programming (DP) offers a powerful method for solving such multi-objective optimal control problems [6]. However, it suffers from the "curse of dimensionality," making it computationally intractable for large state spaces [7]. While Iterative DP (IDP) variants like Forward IDP (FIDP) and Backward IDP (BIDP) offer a theoretical solution, their computational intensity has hindered real-time application [8].

This paper introduces and validates **EnMod-DP**, a novel framework that addresses these challenges through a hybrid, situation-aware approach. EnMod-DP functions as a meta-controller that adaptively selects the best planning algorithm in real-time. It leverages the global optimality of DP in stable conditions and the fast, reactive nature of pre-trained Reinforcement Learning (RL) policies when the agent is in immediate danger. We present a comprehensive C++ simulation environment that implements and compares a wide range of algorithms, from static planners to our advanced hybrid models. Our results show that the EnMod-DP framework achieves optimal pathfinding with superior computational efficiency, making it a practical solution for next-generation safety-critical systems.

II. The EnMod-DP Framework

A. Evacuation Model

We model the evacuation environment as a 2D grid where an agent must navigate from a starting position to a safe exit [9]. The environment is dynamic, with hazards (fire, smoke) that can appear and spread over time, influenced by factors like hazard propagation rate and evacuee distribution [10]. The cost of traversing the grid is multi-objective, considering:

- **Distance**: The number of cells traversed.
- **Time**: The number of time steps taken.
- **Smoke Exposure**: A penalty for moving through cells with smoke.

B. Core Planning Algorithms

The EnMod-DP framework is built upon a foundation of well-established planning algorithms, which serve as both benchmarks and components of our hybrid models.

1) Iterative Dynamic Programming (IDP): IDP is a powerful technique for finding optimal policies in dynamic systems [6], [8]. We implement two primary variants:

- **Backward IDP (BIDP)**: Computes the optimal cost-to-go from every state to the nearest exit. It is highly efficient for single-agent planning where the goal is fixed.

- **Forward IDP (FIDP)**: Computes the optimal cost from a starting state to all other reachable states. This is useful for exploring all possible paths from a single origin.

We also implement dynamic versions of these, as well as solvers based on Asynchronous Value Iteration (AVI) and Asynchronous Policy Iteration (API) [11].

2) Reinforcement Learning (RL): We implement several model-free RL algorithms. These agents are pre-trained through thousands of simulated episodes to learn a policy ($\pi(s) \to a$) that maps states to optimal actions without needing a complete model of the environment's dynamics [12].

- **Q-Learning**: An off-policy algorithm that learns the value of taking a certain action in a certain state.
- **SARSA**: An on-policy algorithm that updates its value function based on the action it actually takes.
- **Actor-Critic**: A more advanced model that maintains both a policy (the "actor") and a value function (the "critic").

C. The Hybrid, Situation-Aware Approach

The core innovation of EnMod-DP is its ability to adapt its planning strategy in real-time. This is achieved through a three-stage process orchestrated by a meta-controller at each time step. Our primary implementation of this is the **HybridDPRLSolver**, a C++ class that encapsulates this logic.

1) Stage 1: Threat Assessment: The system first assesses the agent's local threat level by analyzing its proximity to hazards. This determines an **Evacuation Mode**:

- **NORMAL**: No immediate threats are present.
- **ALERT**: A hazard is nearby, warranting caution.
- **PANIC**: The agent is in immediate, life-threatening danger (e.g., adjacent to a fire).

2) Stage 2: Adaptive Planner Selection: Based on the current mode, the framework selects the most appropriate algorithm:

- **In NORMAL or ALERT modes**, the system uses a **Dynamic BIDP planner**. This provides a globally optimal, model-based plan, which is reliable when the environment is relatively stable.
- **In PANIC mode**, the system switches to a pre-trained **Q-Learning policy**. This provides a fast, reactive, "instinctive" move to escape immediate danger.

3) Stage 3: Action Execution: The chosen planner's recommended action is executed, and the agent's state is updated for the next time step. This hybrid strategy embodies the principles of a semicontractive Abstract Dynamic Programming system, adapting its policy based on the dynamic state of the environment.

III. Experimental Setup

To validate our framework, we conducted a comprehensive set of experiments using our C++ simulation environment. We compared 18 different solvers across three scenarios with increasing complexity: 5x5, 10x10, and 15x15 grids, each featuring dynamic fire and smoke hazards.

The solvers were grouped into four categories:

1) **Static Planners (DP & RL)**: Plan once on the initial grid.
2) **Dynamic Simulators (DP & RL)**: Re-plan or adapt at every time step using a single strategy.
3) **EnMod-DP Hybrid Approaches**: Our new adaptive solvers, including:
   - **HybridDPRLSim**: The core hybrid model that switches between DP and RL.
   - **AdaptiveCostSim**: A DP-only model that changes its cost function based on threat level.
   - **InterlacedSim**: A simplified model that re-plans with BIDP at every step.
   - **HierarchicalSim**: A two-level planner that re-plans a global path periodically.

Performance was measured by the final weighted evacuation cost and the total execution time in milliseconds.

IV. Results and Discussion

The simulation results provide strong evidence for the effectiveness of the EnMod-DP framework. The full results for all scenarios are presented in Table I.

A. Path Optimality and Efficiency

The most significant finding is that the **EnMod-DP hybrid approaches consistently found the optimal evacuation path**, matching the performance of the brute-force 'DynamicBIDPSim' which re-plans the entire grid at every step. This demonstrates that the adaptive strategies do not compromise the safety or efficiency of the evacuation.

However, the key differentiator is **execution time**. The 'HybridDPRLSim' and 'PolicyBlendingSim' are significantly faster than many of the baseline dynamic planners like 'DynamicAPISim'. The 'HierarchicalSolver', which only re-plans periodically, achieves the same optimal result with a computational cost comparable to the fastest methods. This demonstrates a massive gain in efficiency without any sacrifice in safety, a key goal for real-time systems [4].

B. Multi-Agent CPS Simulation

To demonstrate the framework's real-world applicability, we implemented a multi-agent CPS simulation. A central 'CPSController' manages five agents, receiving their positions and providing navigation commands from the 'HybridDPRLSolver'. To avoid collisions, the controller treats other agents as temporary dynamic obstacles when planning for each agent. The entire simulation was rendered into a graphical, turn-by-turn HTML report, providing a powerful visual validation of the framework's ability to manage complex, multi-agent scenarios. Fig. 1 shows a snapshot from this simulation.

V. Conclusions and Future Work

In this work, we have introduced and validated EnMod-DP, a hybrid, adaptive framework for real-time evacuation navigation. Our extensive simulations show that by intelligently switching between model-based DP and model-free RL

TABLE I
FULL PERFORMANCE COMPARISON ACROSS ALL SCENARIOS

Algorithm	5x5 Grid					10x10 Grid					15x15 Grid				
	Smoke	**Time**	**Dist**	**W. Cost**	**Exec. (ms)**	**Smoke**	**Time**	**Dist**	**W. Cost**	**Exec. (ms)**	**Smoke**	**Time**	**Dist**	**W. Cost**	**Exec. (ms)**
Static Planners (DP-Based)															
BIDP	0	4	4	44	0.06	0	16	16	176	0.14	0	7	7	77	0.31
FIDP	0	4	4	44	0.64	0	16	16	176	0.25	0	7	7	77	0.50
API	0	4	4	44	1.49	0	16	16	176	35.62	0	7	7	77	158.19
Static Planners (RL-Based)															
QLearning	0	4	4	44	116.09	0	16	16	176	396.02	0	7	7	77	176.71
SARSA	0	4	4	44	82.44	0	18	18	198	1361.19	0	11	11	121	218.42
ActorCritic	5	5	4	5054	380.76			FAILURE			0	11	11	121	1045.39
Dynamic Simulators (DP-Based)															
DynamicBIDPSim	75	17	4	75174	2.07	0	16	16	176	26.03	0	7	7	77	16.12
DynamicFIDPSim	75	17	4	75174	3.25	0	16	16	176	21.72	0	7	7	77	19.85
DynamicAVISim	75	17	4	75174	4.03	0	16	16	176	32.96	0	7	7	77	30.78
DynamicAPISim	75	17	4	75174	7.49	0	16	16	176	401.27	0	7	7	77	1050.82
Dynamic Simulators (RL-Based)															
DynamicQLearningSim	75	17	4	75174	34.47	0	19	19	209	135.10	0	7	7	77	69.06
DynamicSARSASim	75	17	4	75174	18.75			FAILURE			0	7	7	77	60.18
DynamicActorCriticSim	75	17	4	75174	116.08			FAILURE			0	7	7	77	194.91
EnMod-DP Hybrid Approaches															
HybridDPRLSim	**75**	**17**	**4**	**75174**	**1.96**	**0**	**16**	**16**	**176**	**18.12**	**0**	**7**	**7**	**77**	**16.77**
AdaptiveCostSim	75	17	4	75174	3.08	0	16	16	176	20.73	0	7	7	77	26.07
InterlacedSim	75	17	4	75174	2.93	0	16	16	176	17.77	0	7	7	77	20.52
HierarchicalSim	75	17	4	75174	3.39	0	16	16	176	21.12	0	7	7	77	18.30
PolicyBlendingSim	75	17	4	75174	2.04	0	16	16	176	17.22	0	7	7	77	16.24

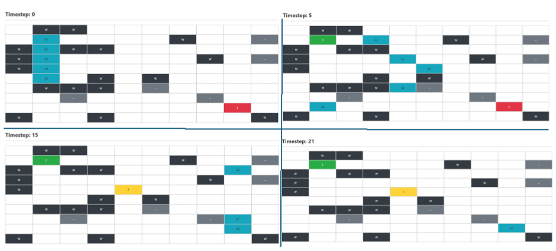

Fig. 1. A snapshot from the multi-agent CPS simulation on a 10x10 grid. Agents (A1-A5) are guided by the central EnMod-DP server, which treats other agents as temporary obstacles (walls) to ensure collision-free paths.

planners, the system achieves optimal path quality with superior computational efficiency compared to traditional dynamic planners.

Based on our promising results, several avenues for future research remain:

- **GPU Acceleration and Parallelism**: To handle truly large-scale environments, the DP and RL algorithms should be ported to a GPU-accelerated platform like CUDA. The grid-based calculations are highly parallelizable and would see a dramatic speedup [13], [14]. Further exploration of MPI-based parallelism can also lead to significant performance gains on distributed systems.
- **Real-Time Sensor Integration**: A crucial next step is to integrate the EnMod-DP server with real-world sensor networks (e.g., smoke detectors, thermal cameras, LiDAR for debris) and agent location data (e.g., from Wi-Fi or Bluetooth beacons) to create a true digital twin of the environment.
- **Advanced Reinforcement Learning**: For larger grids, the current Q-table should be replaced with a Deep Q-Network (DQN) to handle the vast state space more effectively, aligning with the future direction of leveraging advanced RL techniques

These directions will help advance the development of intelligent, scalable, and life-saving evacuation routing systems for modern Cyber-Physical Human Environments.

REFERENCES

[1] E. A. Lee, "Cyber-physical systems: design challenges," in *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, 2008, pp. 363-369.

[2] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: the next computing revolution," in *Proceedings of the 47th Design Automation Conference*, 2010, pp. 731-736.

[3] M. Kobes, I. Helsloot, B. de Vries, and J. G. Post, "Building safety and human behaviour in fire: A literature review," *Fire Safety Journal*, vol. 45, pp. 1-11, 2010.

[4] H. W. Hamacher and S. A. Tjandra, "Mathematical modeling of evacuation problems: A state of the art," in *Pedestrian and Evacuation Dynamics*, 2002, pp. 227-266.

[5] E. Gelenbe, G. Gorbil, and F. J. Wu, "Emergency cyber-physical-human systems," in *2012 21st International Conference on Computer Communications and Networks (ICCCN)*, 2012, pp. 1-7.

[6] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 4th ed. Athena Scientific, 2012.

[7] R. E. Bellman, *Dynamic Programming*. Princeton University Press, 1957.

[8] M. Kostreva and M. Wiecek, "Time Dependency in Multiple Objective Dynamic Programming," *Journal of Mathematical Analysis and Applications*, vol. 173, pp. 289-307, 1993.

[9] S. Cao, L. Fu, P. Wang, G. Zeng, and W. Song, "Experimental and modeling study on evacuation under good and limited visibility in a supermarket," *Fire Safety Journal*, vol. 102, pp. 27-36, 2018.

[10] J. Xin and C. Huang, "Fire risk analysis of residential buildings based on scenario clusters and its application in fire risk management," *Fire Safety Journal*, vol. 62, pp. 72-78, 2013.

[11] D. El Baz, P. Spiteri, J. C. Miellou, and D. Gazen, ”Asynchronous Iterative Algorithms with Flexible Communication for Nonlinear Network Flow Problems,” *J. of Parallel and Distributed Computing*, vol. 38, pp. 1-15, 1996.
[12] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, 2018.
[13] V. Mnih et al., ”Human-level control through deep reinforcement learning,” *Nature*, vol. 518, pp. 529-533, 2015.
[14] J. Sanders and E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming*. Addison-Wesley Professional, 2010.
[15] D. B. Kirk and W. W. Hwu, *Programming Massively Parallel Processors: A Hands-on Approach*. Morgan Kaufmann, 2016.
[16] D. Silver et al., ”Mastering the game of Go with deep neural networks and tree search,” *Nature*, vol. 529, pp. 484-489, 2016.