Submitted to:

- Sir Muhammad Ata Ur Rehman

Group Members:

- Munim Siddiqui (20K-1730)
- Murtaza Salman(20K-0398)
- Ali Qureshi (20K-0149)

Point-based Value Iteration for Neuro-Symbolic POMDPs

Rui Yana , Gabriel Santosa , Gethin Normanb , David Parkera , Marta Kwiatkowskaa

Table of contents

Table of contents

01 Introduction

- **AI Techniques Integration:** Explores merging symbolic AI with neural perception for decision-making under uncertainty.
- **NS-POMDPs Framework:** Introduces neuro-symbolic partially observable Markov decision processes to handle continuous environments.
- **Optimization Objective**: Aims to optimize rewards in complex, continuous-state spaces, highlighting scalability challenges.
- **Algorithmic Innovations:** Presents two novel value iteration algorithms tailored for NS-POMDPs that exploit neural network insights.
- **Real-world Application:** Demonstrates effectiveness through case studies on dynamic car parking and aircraft collision avoidance.

02 Background

- **POMDPs and Challenges:** Outlines traditional partially observable Markov decision processes (POMDPs) and their limitations in handling continuous-state environments.
- **Neuro-Symbolic Systems:** Discusses the integration of neural networks with symbolic decision-making systems, emphasizing their potential in complex control tasks.
- **Continuous-State Spaces:** Highlights the challenges of solving POMDPs with continuous states, where traditional discrete methods fail.
- **Point-based Methods:** Explains existing point-based value iteration techniques for finite-state models and their extension challenges to continuous domains.
- **Scalability Issues:** Addresses scalability and efficiency issues when applying traditional methods to continuous-state POMDPs .

03

Neuro-Symbolic Partially Observable Markov Decision Processes (NS-POMDPs)

- **Framework Introduction:** NS-POMDPs combine neural network perception and symbolic decision-making for continuous environments.
- **Components:** Composed of symbolic decision processes and neural perception, tailored for dynamic, uncertain settings.
- **Decision Making:** Decisions in NS-POMDPs rely on symbolic representations with neural network-driven observation functions.
- **State and Observation Handling:** Handles continuous states and discrete actions, with neural networks synthesizing environmental observations.

03

Neuro-Symbolic Partially Observable Markov Decision Processes (NS-POMDPs)

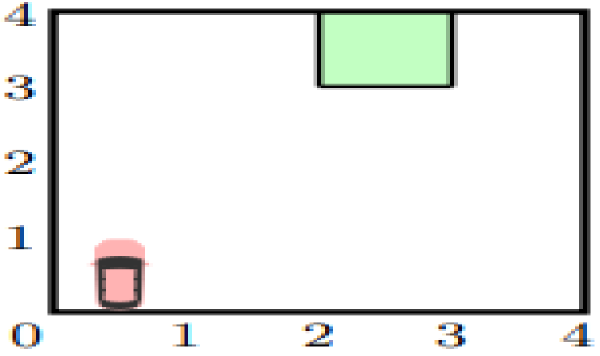


Figure 1: Car parking example

04 Value Iteration

- **Adaptation for NS-POMDPs:** The value iteration algorithm is adapted to handle the complexities of NS-POMDPs, which incorporate both continuous state spaces and neural network-based perceptions .
- **Algorithm Extensions:** Two specific algorithms are introduced — a classical value iteration algorithm and a point-based heuristic search value iteration (NS-HSVI), both tailored to the NS-POMDP framework .
- **Function Representation:** These algorithms utilize a novel piecewise linear and convex representation (P-PWLC) for functions over continuous state beliefs, enhancing computational efficiency .

04 Value Iteration

- **Convergence Properties:** The algorithms are designed to ensure the convergence and continuity of value functions, maintaining the mathematical integrity required for robust decision-making in NS-POMDPs .
- **Finite Representability:** By exploiting the structure of the continuous-state model and neural perception mechanisms, the algorithms ensure finite representability, crucial for practical implementation .
- **Practical Applicability:** Demonstrated through prototype implementations and case studies, showing that these value iteration methods are not only theoretically sound but also practically viable .

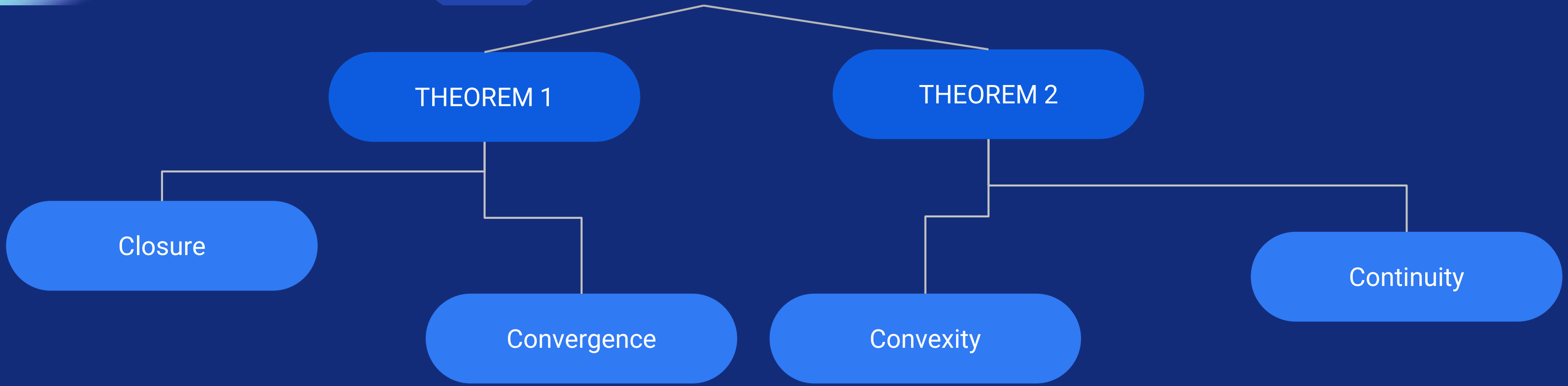
05 PWC Representations

- **Function Definition:** PWC functions are used to represent various elements within NS-POMDPs, such as perception functions and reward structures, to handle the complexity of continuous spaces effectively .
- **State Space Partitioning:** PWC representations enable the partitioning of the state space into observationally equivalent regions, simplifying the belief state updates and calculations involved in NS-POMDPs .
- **Efficient Computation:** By utilizing PWC functions, the algorithms can operate directly on a finite set of distinct regions instead of the entire continuous state space, significantly reducing computational complexity .

05 PWC Representations

- **Support for Neural Networks:** These representations are compatible with the outputs of neural networks used for perception, allowing seamless integration of data-driven perception into the symbolic decision-making framework of NS-POMDPs .
- **Mathematical Properties:** The PWC approach maintains crucial mathematical properties such as linearity and convexity, which are necessary for the effective application of Bellman updates and value iteration methods within NS-POMDPs .
- **Practical Implementation:** PWC representations facilitate the practical implementation of NS-POMDPs by enabling a structured and manageable approach to handling continuous variables and complex decision processes .

06 Value Function



Theorem 1 (P-PWLC closure and convergence). *If* $V \in \mathbb{F}(S_B)$ *and P-PWLC, then so is* $[TV]$. *If* $V^0 \in \mathbb{F}(S_B)$ *and P-PWLC, then the sequence* $(V^t)_{t=0}^{\infty}$, *such that* $V^{t+1} = [TV^t]$ *are P-PWLC and converges to* $V^\star$.

Theorem 2 (Convexity and continuity). *For any* $s_A \in S_A$, *the value function* $V^\star(s_A, \cdot) : \mathbb{P}(S_E) \to \mathbb{R}$ *is convex and for any* $b_E, b'_E \in \mathbb{P}(S_E)$:

$$|V^\star(s_A, b_E) - V^\star(s_A, b'_E)| \leq K(b_E, b'_E) \qquad (8)$$

where $K(b_E, b'_E) = (U - L)\int_{s_E \in S_E^{b_E > b'_E}} (b_E(s_E) - b'_E(s_E))\mathrm{d}s_E$ *and* $S_E^{b_E > b'_E} = \{s_E \in S_E^{s_A} \mid b_E(s_E) - b'_E(s_E) > 0\}$.

07

Heuristic Search Value Iteration

It is a technique commonly used in decision-making processes, where you have uncertainty about the environment. Traditional value iteration works well for problems with a finite number of states but struggles with continuous-state problems like NS-POMDPs because it relies on discretization or approximation

08 Lower and Upper Bound Representations

- The lower bound function V_Γ^{LB} is represented as a piecewise linear and convex (P-PWLC) function using a finite set of piecewise constant (PWC) alpha (α) functions.
- Each alpha function corresponds to a specific partition of the state space, ensuring that the lower bound is finitely representable and captures the structure of the NS-POMDP accurately.

2. **Upper Bound Function:**
   - The upper bound V_Υ^{UB} is represented differently. Instead of using alpha functions, it's represented by a finite set of belief-value points.
   - Each belief-value point consists of a belief state (describing the agent's uncertainty) and an associated upper bound value.
   - Since the value function V^* is convex (as per Theorem 2), a convex combination of these belief-value points provides an upper bound on V^* in finite-state POMDPs. However, this doesn't directly apply to NS-POMDPs due to the continuous belief space.
   - Therefore, the upper bound function in NS-POMDPs is defined as the lower envelope of the lower convex hull of the belief-value points. This is a way to ensure that the upper bound covers the belief space effectively.

09 Point-Based Updates

Algorithm 1 Point-based $Update(s_A, b_E)$ of $(V_{LB}^{\Gamma}, V_{UB}^{\Upsilon})$

$\bar{a} \leftarrow$ the maximum action in computing $[TV_{LB}^{\Gamma}](s_A, b_E)$
$\bar{S}_A \leftarrow \{s'_A \in S_A \mid P(s'_A \mid (s_A, b_E), \bar{a}) > 0\}$
$\alpha^{s'_A} \leftarrow \text{argmax}_{\alpha \in \Gamma} \int_{s_E \in S_E} bval((s_A, s_E), \bar{a}, s'_A, \alpha) b_E(s_E) \mathrm{d}s_E$ for all $s'_A \in \bar{S}_A$
for $\phi \in \Phi_P$ **do**
    if $s_A^{\phi} = s_A$ and $\int_{(s_A, s_E) \in \phi} b_E(s_E) \mathrm{d}s_E > 0$ **then**
        Compute $\alpha^{\star}(\hat{s}_A, \hat{s}_E)$ by (12) for $(\hat{s}_A, \hat{s}_E) \in \phi$ ▷ ISPP backup
    else $\alpha^{\star}(\hat{s}_A, \hat{s}_E) \leftarrow L$ for $(\hat{s}_A, \hat{s}_E) \in \phi$
$\Gamma \leftarrow \Gamma \cup \{\alpha^{\star}\}$
$p^{\star} \leftarrow [TV_{UB}^{\Upsilon}](s_A, b_E)$
$\Upsilon \leftarrow \Upsilon \cup \{((s_A, b_E), p^{\star})\}$

NS-HSVI For NS-POMDPs

Algorithm 3 NS-HSVI for NS-POMDPs

Initialize V_{LB}^{Γ} and V_{UB}^{Υ}
while $V_{UB}^{\Upsilon}((s_A^{init}, b_E^{init}) - V_{LB}^{\Gamma}(s_A^{init}, b_E^{init}) > \varepsilon$ **do** $\mathit{Explore}((s_A^{init}, b_E^{init}), \varepsilon, 0)$
function $\mathit{Explore}((s_A, b_E), \varepsilon, t)$
    if $V_{UB}^{\Upsilon}(s_A, b_E) - V_{LB}^{\Gamma}(s_A, b_E) \leq \varepsilon\beta^{-t}$ **then return**
    for $a \in \Delta_A(s_A)$, $s'_A \in S_A$ **do**
        $p^{a,s'_A} \leftarrow P(s'_A \mid (s_A, b_E), a) V_{UB}^{\Upsilon}(s'_A, b_E^{s_A,a,s'_A})$
    $\hat{a} \leftarrow \mathrm{argmax}_{a \in \Delta_A(s_A)} \langle R_a, (s_A, b_E) \rangle + \beta \sum_{s'_A \in S_A} p^{a,s'_A}$
    $\mathit{Update}(s_A, b_E)$
    $\hat{s}_A \leftarrow \mathrm{argmax}_{s'_A \in S_A}\, \mathit{excess}_{t+1}(s'_A, b_E^{s_A,\hat{a},s'_A})$
    $\mathit{Explore}((\hat{s}_A, b_E^{s_A,\hat{a},\hat{s}_A}), \varepsilon, t+1)$
    $\mathit{Update}(s_A, b_E)$

11

Two Belief Representation

Particle-Based Beliefs

Definition: A particle-based belief is represented by a set of weighted particles, each representing a possible state of the environment with a certain probability (weight).

Region-Based Beliefs

Definition: A region-based belief divides the environment into regions and represents each region with a uniform distribution over its area.

12

Implementation and Experimental Evaluation

- Car Parking Case Study

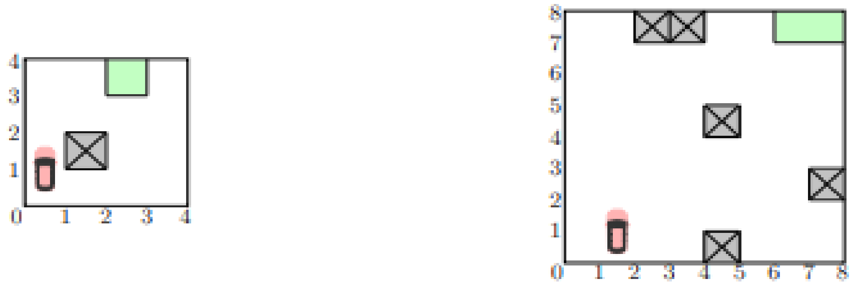


Figure 3: Car parking with obstacles.

Implementation and Experimental Evaluation

- VCAS Case Study

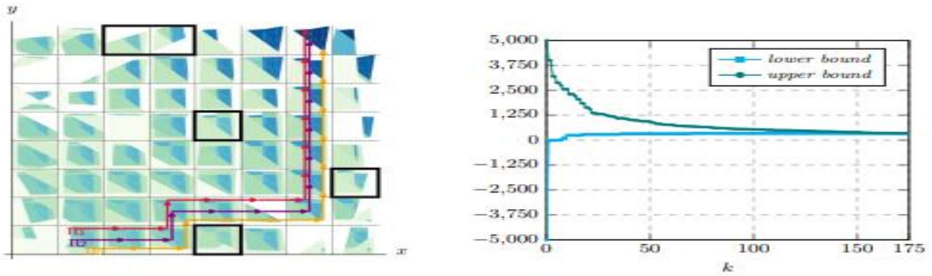


Figure 7: Paths and values for car parking (8×8, $\beta = 0.8$, partially reconstructed).

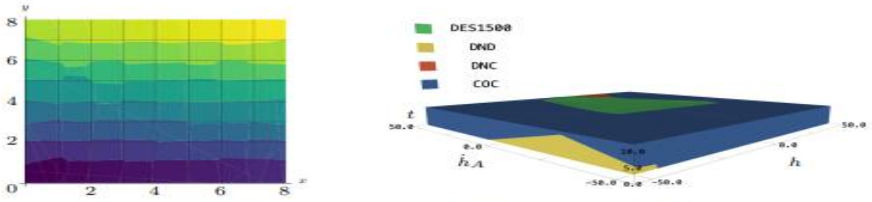


Figure 8: Perception FCP for car parking (8×8), and a slice of the perception FCP for the COC advisory of the VCAS (h scaled 10:1).

13

Performance Analysis

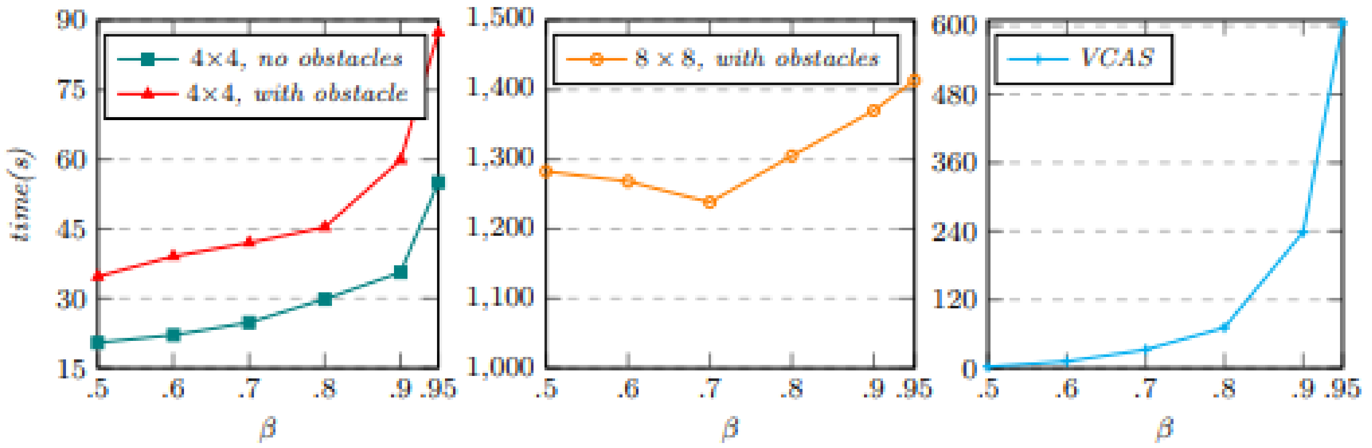


Figure 10: Solution times for different discount factors (for particle-based beliefs).

13

Performance Analysis

Model	Belief type #initial	Total regions (α-functions)	Lower bound	Upper bound	Strat. time (s)	Following ratio	Avg. trust
Car parking (no obstacles, 4×4)	PB, 3	80,494	2389.3309	2389.3333	19.3	88%	3.6
	PB, 5	42,224	2047.9989	2048.0000	14.0	100%	3.9
	RB, 1	36,467	2047.9992	2048.0000	50.0	100%	3.9
Car parking (w/ obstacle, 4×4)	PB, 3	99,513	2218.6653	2218.6666	24.5	78%	3.3
	PB, 5	47,719	2047.9990	2048.0000	14.2	100%	3.9
	RB, 1	35,751	2047.9988	2048.0000	39.4	100%	3.9
Car parking (w/ obstacles, 8×8)	PB, 3	1,410,799	343.5969	343.5974	338.9	85%	4.3
	PB, 5	547,753	343.5970	343.5974	158.4	97%	4.4
	RB, 1	550,685	343.5964	343.5974	473.8	80%	4.3
VCAS (3 actions)	PB, 4	154,009	-1.2281	0.0	75.3	-	-
	PB, 5	278,447	-1.2398	0.0	127.5	-	-
	PB, 6	868,257	-0.2498	0.0	400.8	-	-
	RB, 1	22,919	-0.0715	0.0	65.5	-	-
VCAS (15 actions)	PB, 4	32,387	-0.6718	0.0	18.7	33%	1.3
	PB, 5	30,003	-0.9874	0.0	21.7	0%	1.0
	PB, 6	19,218	-1.0789	0.0	13.0	33%	1.3
	RB, 1	21,102	-0.6133	0.0	49.9	0%	1.0

Table 4: Further statistics for a set of NS-POMDP solution instances.

13

Performance Analysis

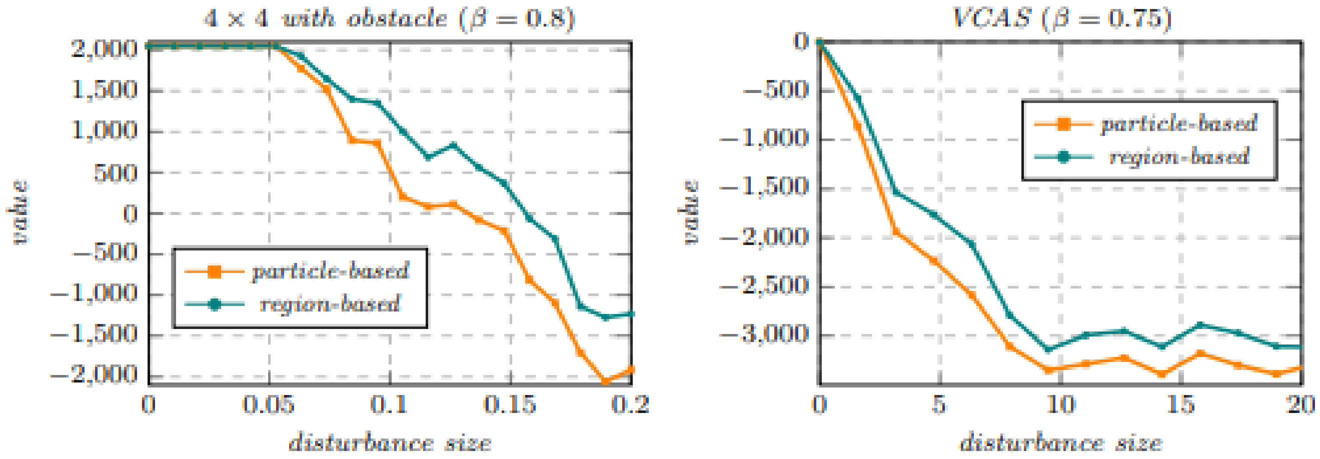


Figure 11: Comparison between particle-based and region-based values.

13 Performance Analysis

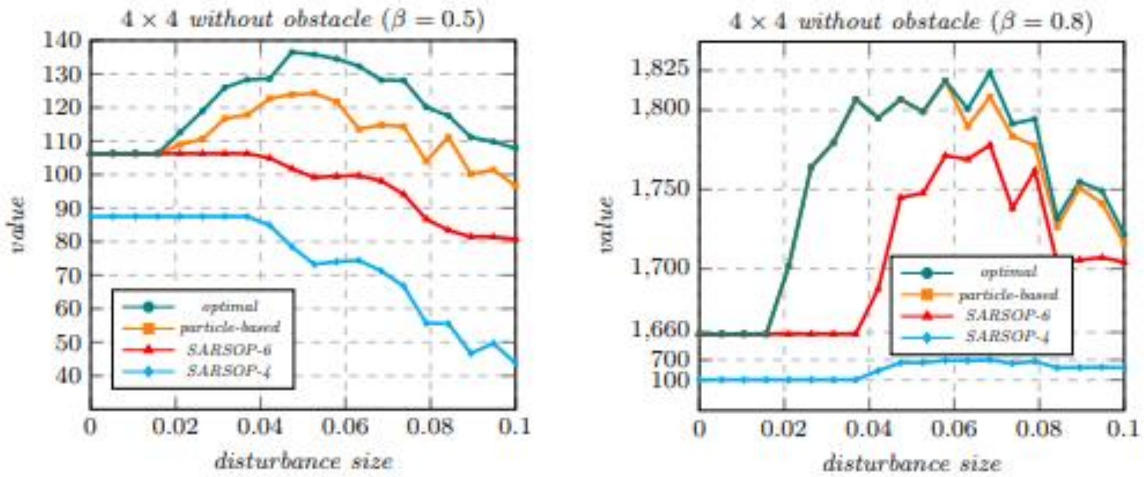


Figure 12: Comparison between particle-based and SARSOP values.

13

Performance Comparison

- The NS-HSVI algorithm was compared against the finite-state POMDP solver SARSOP.
- SARSOP was faster due to its discrete model, while NS-HSVI handled continuous spaces and required expensive operations on polyhedra.
- NS-HSVI showed superior or equal lower bounds for particle-based beliefs compared to SARSOP within a small disturbance range.
- The comparison highlights NS-HSVI's efficiency in updating regions and adapting to disturbances in beliefs.

02 Conclusion

1. **NS-POMDPs Introduction:** NS-POMDPs are a new way to use AI models that blend neural networks with traditional techniques for decision-making in environments where data is incomplete or uncertain.
2. **Optimal Policy Synthesis:** The NS-HSVI algorithm was used to find the best strategies for parking a car and avoiding collisions in aircraft using these AI models.
3. **Future Work:** The team plans to make the method faster, handle more complex situations, and enable multiple agents to work together.

Thankyou!
Any Questions?