

# Bayesian Hierarchical Partial Order Planning: A Framework for Multi-Agent Coordination under Uncertainty

Your Name  
Department  
University  
Email

April 21, 2025

## Abstract

This paper introduces Bayesian Hierarchical Partial Order Planning (BHPOP), a novel framework for representing and reasoning about plans under uncertainty. The key innovation is the combination of partial-order plans, which allow for flexibility and concurrency, with Bayesian inference, which handles uncertainty in a principled manner. Our approach represents partial orders using a multi-dimensional latent space model and incorporates Markov Chain Monte Carlo (MCMC) methods for inferring partial orders from observed execution sequences. We extend the framework to multi-agent scenarios through a hierarchical model that enables coordination while preserving agent autonomy. We demonstrate our approach on a disaster recovery planning problem, showing how it can effectively infer planning constraints and success probabilities from observed action sequences. Experimental results indicate that our approach can effectively capture complex planning dependencies, adapt to noisy observations, and scale to realistic planning scenarios with multiple agents.

## 1 Introduction

Planning under uncertainty is a fundamental challenge in artificial intelligence, robotics, and multi-agent systems. Traditional planning approaches of-

ten struggle to handle concurrency, uncertainty, and coordination among multiple agents. Partial-order planning (POP) offers a promising approach by representing plans as a set of actions linked by ordering constraints only where necessary, maximizing concurrency and flexibility [Weld, 1994]. However, classical POP approaches do not naturally account for uncertainty or learn from observations.

In this paper, we introduce Bayesian Hierarchical Partial Order Planning (BHPOP), which combines the flexibility of partial-order plans with the principled handling of uncertainty offered by Bayesian inference. Our key contributions include:

- A Bayesian framework for representing and inferring partial-order plans based on observed action executions
- A hierarchical extension that enables coordination in multi-agent settings while preserving concurrency
- A formulation that captures both action ordering constraints and success/failure outcomes
- An MCMC sampling approach for efficient posterior inference
- Experimental validation on a disaster recovery planning scenario

The rest of the paper is organized as follows: Section 2 discusses related work, Section 3 presents the

mathematical framework for BHPOP, Section 4 extends the approach to hierarchical multi-agent settings, Section 5 describes the MCMC sampling approach, Section 6 presents experimental results, and Section 7 concludes with a discussion of limitations and future work.

## 2 Related Work

### 2.1 Partial-Order Planning

Partial-order planning was introduced as an alternative to total-order planning to maximize flexibility and concurrency [Sacerdoti, 1975]. Instead of committing to a specific sequence of actions, POP represents plans as a partially ordered set of actions, where ordering constraints are imposed only when necessary to resolve conflicts or ensure proper causal relationships [Weld, 1994]. This approach allows for more flexibility in execution and has been particularly successful in domains where concurrent actions are common.

### 2.2 Planning under Uncertainty

Planning under uncertainty has been addressed through various frameworks, including Markov Decision Processes (MDPs) [Puterman, 2014] and Partially Observable MDPs (POMDPs) [Kaelbling et al., 1998]. These approaches model uncertainty in action outcomes and state transitions but typically rely on total-order plans. Probabilistic planning approaches extend classical planning to handle uncertainty in action outcomes [Kushmerick et al., 1995], but often do not fully exploit the potential for concurrency.

### 2.3 Multi-Agent Planning

Multi-agent planning extends planning to settings with multiple autonomous agents. Approaches include centralized planning for decentralized execution [Boutilier, 1999], distributed planning where agents coordinate their individual plans [Durfee, 1999], and hierarchical approaches that decompose planning problems into subproblems [Tambe, 1997]. However,

these approaches often struggle to balance coordination and concurrency effectively.

## 2.4 Bayesian Approaches to Planning

Bayesian approaches to planning leverage probabilistic inference to handle uncertainty and learn from data. Bayesian Reinforcement Learning [Ghavamzadeh et al., 2015] uses Bayesian inference to learn transition dynamics and reward functions in MDPs. Probabilistic plan recognition approaches use Bayesian inference to infer an agent’s plan from observed actions [Charniak, 1993, Geib and Goldman, 2001]. However, few approaches combine Bayesian inference with partial-order planning or extend to multi-agent settings.

## 3 Bayesian Partial Order Planning

### 3.1 Partial Order Representation

A partial order on a set of actions  $\mathcal{A} = \{A_1, A_2, \dots, A_n\}$  is a binary relation  $\prec$  that is:

- Irreflexive:  $\neg(A_i \prec A_i)$  for all  $A_i \in \mathcal{A}$
- Transitive:  $(A_i \prec A_j) \wedge (A_j \prec A_k) \Rightarrow (A_i \prec A_k)$  for all  $A_i, A_j, A_k \in \mathcal{A}$
- Antisymmetric:  $(A_i \prec A_j) \Rightarrow \neg(A_j \prec A_i)$  for all  $A_i, A_j \in \mathcal{A}$

We represent partial orders using a latent space model based on the theorem that any partial order can be represented as the intersection of a set of total orders [Baker et al., 1994]. Specifically, we associate each action  $A_j$  with a latent vector  $U_j \in \mathbb{R}^K$ , where  $K$  is the dimension of the latent space. The partial order is then induced by the component-wise dominance relation:

$$A_i \prec A_j \iff U_{i,k} > U_{j,k} \text{ for all } k \in \{1, \dots, K\} \quad (1)$$

This representation allows us to infer partial orders from data by learning the latent vectors  $U_j$ .

### 3.2 Noise Model for Observed Sequences

In practice, observed action sequences may not perfectly follow the underlying partial order due to various sources of noise. We introduce a "queue-jump" noise model with parameter  $p \in [0, 1]$ :

- With probability  $1 - p$ , the next action is chosen uniformly from the set of maximal feasible actions (those with no remaining predecessors in the partial order)
- With probability  $p$ , the next action is chosen uniformly from all remaining actions, potentially violating the partial order constraints

This model allows us to account for deviations from the partial order while still inferring the underlying constraints.

### 3.3 Success/Failure Outcomes

We extend our model to handle success/failure outcomes for actions. Each action  $A_j$  has an associated success probability  $\pi_j \in (0, 1)$ . When action  $A_j$  is executed, it succeeds with probability  $\pi_j$  and fails with probability  $1 - \pi_j$ .

The likelihood of observing a sequence of actions with their success/failure outcomes is:

$$p(\text{sequence, outcomes} \mid \text{partial order}, \{\pi_j\}, p) = \prod_{\text{step } s} [p(\text{action}_s \mid \text{partial order}, p) \times p(\text{outcome}_s \mid \pi_{\text{action}_s})] \quad (2)$$

where  $p(\text{action}_s \mid \text{partial order}, p)$  is determined by the queue-jump noise model and  $p(\text{outcome}_s \mid \pi_{\text{action}_s})$  is  $\pi_{\text{action}_s}$  for successful outcomes and  $1 - \pi_{\text{action}_s}$  for failures.

## 4 Hierarchical Extension for Multi-Agent Planning

We extend our approach to multi-agent settings through a hierarchical model. Consider a set of agents  $\{1, \dots, A\}$ , each with a local set of actions  $\mathcal{A}_a$ .

### 4.1 Hierarchical Latent Space Model

We define a global latent space for actions across all agents, with global latent vectors  $U_j^{(0)}$  for each action  $j$  in the union of all agents' action sets. For each agent  $a$ , we define local latent vectors  $U_j^{(a)}$  for each action  $j \in \mathcal{A}_a$ .

The local latent vectors are related to the global latent vectors through a hierarchical model:

$$U_j^{(a)} \sim \mathcal{N}(\tau U_j^{(0)}, (1 - \tau^2) \Sigma_\rho) \quad (3)$$

where  $\tau \in (0, 1)$  is a hierarchical parameter controlling the strength of the connection between global and local latent spaces, and  $\Sigma_\rho$  is a covariance matrix with parameter  $\rho$  controlling the correlation between dimensions.

### 4.2 Cross-Agent Constraints

The hierarchical model allows for both agent-specific partial orders and cross-agent constraints:

- Within each agent, local partial orders are induced by the component-wise dominance of local latent vectors  $U_j^{(a)}$
- Cross-agent constraints are induced by the component-wise dominance of global latent vectors  $U_j^{(0)}$

This approach naturally balances agent autonomy with coordination requirements, allowing for concurrency both within and across agents when there are no explicit constraints.

## 5 MCMC Inference

We use Markov Chain Monte Carlo (MCMC) methods to infer posterior distributions over the model parameters given observed action sequences. The parameters to be inferred include:

- Latent vectors  $U_j^{(0)}$  and  $U_j^{(a)}$
- Correlation parameter  $\rho$
- Hierarchical parameter  $\tau$

- Queue-jump noise parameter  $p$
- Success probabilities  $\{\pi_j\}$

## 5.1 Priors

We place the following priors on the model parameters:

$$U_j^{(0)} \sim \mathcal{N}(0, \Sigma_\rho) \quad (4)$$

$$\rho \sim \text{Uniform}(0, 1) \quad (5)$$

$$\tau \sim \text{Uniform}(0, 1) \quad (6)$$

$$p \sim \text{Beta}(1, 9) \quad (7)$$

$$\pi_j \sim \text{Beta}(1, 1) \quad (8)$$

## 5.2 MCMC Updates

We use a combination of Metropolis-Hastings updates for different parameters:

---

### Algorithm 1 MCMC Inference for BHPOP

---

- 1: Initialize parameters  $\{U_j^{(0)}\}, \{U_j^{(a)}\}, \rho, \tau, p, \{\pi_j\}$
  - 2: **for** iteration = 1 to  $N$  **do**
  - 3:   Choose update type uniformly from  $\{U^{(0)}, U^{(a)}, \rho, \tau, p, \pi\}$
  - 4:   Propose new parameter value
  - 5:   Compute acceptance ratio  $\alpha$  based on likelihood and prior
  - 6:   Accept proposal with probability  $\min(1, \alpha)$
- 

The acceptance ratio for a parameter update depends on the likelihood of the observed sequences under the proposed and current parameter values, as well as the prior ratio and any necessary Jacobian terms for transformed parameters.

## 6 Experiments

We evaluate our approach on a disaster recovery planning scenario with two agents (robots) operating in two disaster locations and a central hospital.

## 6.1 Experimental Setup

### 6.1.1 Actions

Each robot can perform three types of actions:

- Clear debris at a location
- Search for survivors at a location
- Transport survivors from a location to the hospital

### 6.1.2 Constraints

The scenario includes the following constraints:

- Debris must be cleared before searching for survivors
- Survivors must be found before transport
- If robots share a transport vehicle, their transport actions cannot occur simultaneously

### 6.1.3 Data Generation

We generate synthetic data by sampling execution sequences from the true partial order with varying levels of queue-jump noise ( $p \in \{0.1, 0.2, 0.3\}$ ). We also sample success/failure outcomes based on true success probabilities.

## 6.2 Results

### 6.2.1 Partial Order Recovery

We measure the accuracy of the inferred partial order by comparing the inferred and true ordering constraints. Figure 1 shows the F1 score for constraint recovery as a function of the number of observed sequences and the noise level.

### 6.2.2 Success Probability Estimation

We evaluate the accuracy of the inferred success probabilities by measuring the mean absolute error between the inferred and true probabilities. Figure 2 shows the results as a function of the number of observed sequences.

### 6.2.3 Cross-Agent Constraint Inference

We assess the ability of our approach to infer cross-agent constraints by measuring the precision and recall of the inferred constraints. Figure 3 shows the results.

## 6.3 Discussion

Our experiments demonstrate that BHPOP can effectively infer partial orders and success probabilities from observed action sequences, even in the presence of noise. The hierarchical extension successfully captures cross-agent constraints while allowing for autonomy and concurrency when possible. The MCMC inference approach efficiently explores the parameter space and converges to accurate posterior estimates.

## 7 Conclusion

We have introduced Bayesian Hierarchical Partial Order Planning, a novel framework that combines the flexibility of partial-order planning with the principled handling of uncertainty offered by Bayesian inference. Our approach allows for concurrency, handles uncertainty, and scales to multi-agent settings through a hierarchical model.

Future work includes extending the approach to handle dynamic environments where the partial order may change over time, integrating with reinforcement learning to optimize policies over partial orders, and applying the approach to real-world planning problems in robotics and multi-agent systems.

## References

- Stephen Baker, Yonatan Gutman, and Ron Schneidman. Partial order, not partial chaos. *Journal of Combinatorial Theory, Series A*, 67(2):174–183, 1994.
- Craig Boutilier. Sequential optimality and coordination in multiagent systems. *International Joint Conference on Artificial Intelligence*, 16:478–485, 1999.
- Eugene Charniak. Bayesian networks without tears. *AI Magazine*, 12(4):50–63, 1993.
- Edmund H Durfee. Distributed problem solving and planning. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 121–164, 1999.
- Christopher W Geib and Robert P Goldman. Plan recognition in intrusion detection systems. In *DARPA Information Survivability Conference & Exposition II*, volume 1, pages 46–55. IEEE, 2001.
- Mohammad Ghavamzadeh, Shie Mannor, Joelle Pineau, and Aviv Tamar. Bayesian reinforcement learning: A survey. *Foundations and Trends in Machine Learning*, 8(5-6):359–483, 2015.
- Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998.
- Nicholas Kushmerick, Steve Hanks, and Daniel Weld. An algorithm for probabilistic planning. *Artificial Intelligence*, 76(1-2):239–286, 1995.
- Martin L Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.
- Earl D Sacerdoti. The nonlinear nature of plans. *International Joint Conference on Artificial Intelligence*, 4:206–214, 1975.
- Milind Tambe. Towards flexible teamwork. In *Journal of Artificial Intelligence Research*, volume 7, pages 83–124, 1997.
- Daniel S Weld. An introduction to least commitment planning. *AI Magazine*, 15(4):27–27, 1994.