Reinforcement Learning as Probabilistic Inference: Application to the Vehicle Routing Problem

Team 14 - Amine Kobeissi, Danil Garmaev, Jeremy Kim, Michael Almanza

Université de Montréal

ABSTRACT

In this project, we explore the use of reinforcement learning (RL) reformulated as a probabilistic inference problem for solving the Vehicle Routing Problem. Building on the framework presented in *'Reinforcement Learning and Control as Probabilistic Inference: Tutorial and Review'* by S. Levine, we leverage the connection between RL and graphical models, where policy optimization is treated as a probabilistic inference process. By incorporating the maximum entropy principle, we aim to improve exploration efficiency and policy robustness.

We re-implemented and adapted the Maximum Entropy RL, and Soft Actor-Critic (SAC) algorithm, as proposed in the paper "*Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor*" by Haarnoja et al. This framework was originally designed to solve problems in continuous spaces, in this project, we implement an adapted approach to evaluate the performance of this framework on discrete problems such as VRP instances of increasing complexity, with metrics such as total travel cost, computational efficiency, and exploration quality. Experiments are conducted on synthetic VRP environments and benchmark datasets like the Solomon Dataset to validate the approach.

This research bridges the gap between probabilistic graphical models (PGMs) and RL, and extends this current framework to discrete cases, demonstrating the potential of inference-based RL techniques for complex route optimization problems.

INTRODUCTION

The Vehicle Routing Problem (VRP) is a cornerstone of optimization research, with real-world applications spanning logistics, transportation, and supply chain management. The challenge lies in efficiently routing vehicles to serve customers while minimizing costs such as distance, fuel, or time, often under complex constraints. While traditional optimization techniques, like heuristics or exact solvers, have achieved success, they struggle to scale to larger, more dynamic problem instances.

Reinforcement Learning (RL) offers a promising alternative by enabling intelligent agents to learn adaptive routing strategies through interaction with the environment. However, standard RL approaches often face difficulties with exploration and stability, particularly in high-dimensional or sparse-reward settings.

To address these challenges, we explore inference-based RL techniques, specifically the Soft Actor-Critic (SAC) algorithm and Maximum Entropy (ME) methods. By framing the VRP as a probabilistic inference problem, we integrate the maximum entropy principle to enhance exploration and policy robustness. This allows the agent to discover diverse, high-quality routing solutions while avoiding premature convergence.

Through experiments on synthetic datasets, we demonstrate the potential of these methods to outperform traditional RL techniques like Deep Q-Networks (DQN), providing a foundation for applying inference-based RL to more complex optimization problems.

Reinforcement Learning introduction

In Reinforcement Learning (RL) problems, an intelligent agent learns to take actions in an environment to maximize rewards over time. A basic RL problem is modeled as a Markov Decision Process (MDP):

- A set of environment and agent states / state space ($s \in S$)
- A set of actions ($a \in A$) of the agent
- State transition probability: transition probability at time t from s_t to s_{t+1} under action a_t [$s_{t+1} \sim p(s_{t+1}|s_t, a_t)$]
- Reward: given after transition from s_t under action a_t [$r_t = r(s_t, a_t)$]

METHODOLOGY

- The *Optimality variable O* represents whether a trajectory is optimal. It is important for inverse RL problem (inferring the hidden preferences of another agent from its observed behavior)
- $$\mathrm{p}(\mathcal{O}_\mathrm{t} = 1 \mid \mathrm{s}_\mathrm{t}, \mathrm{a}_\mathrm{t}) = \exp(\mathrm{r}(\mathrm{s}_\mathrm{t}, \mathrm{a}_\mathrm{t})).$$
- The RL problem is reframed as finding the probability of a trajectory being optimal:

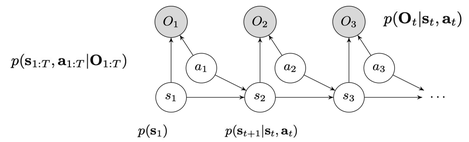

Figure 1. *Graphical model with optimality variables (observed).*

1. **Backward messaging**

   Propagates how "good" each state-action pair (s_t ,a_t) is for achieving the goal - starting from the goal and working backwards.
2. **Optimal policy**

   Uses information from **backward messaging** to select the best action in each state to maximize success.
3. **Forward messaging**

   Uses the **optimal policy** to generate trajectories by propagating forward in time from the initial state. Samples actions from the **optimal action distribution** and generates **a path** through the system.

Improving exploration with Variational Inference

- Optimality variables often push the agent toward short-term optimal actions. To prevent this, we use variational inference to introduce a new trajectory distribution that is no longer dependent on optimality variables. It leads to a broader exploration, so the graphical model now looks slightly different as the figure below:

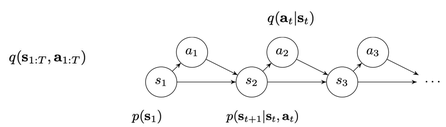

Figure 2. *Graphical model without the optimality variables.*

- The problem reduces to approximating distribution via KL-divergence. By reformulating it, we obtain a reward with an entropy term that controls exploration:

$$\underset{\theta\,=\,(\mathbf{s}_{1:T},\,\mathbf{a}_{1:T})}{\operatorname{argmin}} D_{KL}(q(\theta), p(\theta|\mathbf{O}_{1:T}=1)) = \underset{\theta\,=\,(\mathbf{s}_{1:T},\,\mathbf{a}_{1:T})}{\operatorname{argmax}} \mathbb{E}_{r\sim q}\left[\sum_t r(\mathbf{s}_t, \mathbf{a}_t) + H(q(\mathbf{a}_t|\mathbf{s}_t))\right]$$

Implementation

To address the Vehicle Routing Problem (VRP), we developed a synthetic grid-based environment, where we implemented the **Soft Actor-Critic (SAC)**, and **Maximum Entropy RL (MaxEntRL)** algorithms proposed in the literature from scratch using PyTorch and NumPy. A custom environment was designed to simulate a 2D grid where an agent starts at a central depot and must visit randomly placed customers while minimizing travel distance. The state representation combines the agent's current position, customer locations, and a binary indicator for visited customers.

A Deep-Q Network (DQN) and Proximal Policy Optimization (PPO) algorithm were used as reinforcement learning baselines, and Nearest Neighbor and OR-Tools were used as Heuristic baselines. SAC employs an actor-critic framework that outputs probabilistic continuous actions mapped to discrete indices, integrating entropy maximization to balance exploration and exploitation. MaxEntRL uses entropy-regularized policies, encouraging diverse action selection while optimizing both the policy and state-value networks.

The agents were trained for 500 and 1000 episodes using experience replay and periodic target network updates. Episodic rewards, calculated as negative Euclidean distances with penalties for invalid actions, were tracked to evaluate performance. Results were visualized using raw and smoothed reward plots.

RESULTS

Results of our experiments consist of a comparison of average distance, computation time and reward per method on two environments of increasing complexity.

Initial experiments for these methods were run on a synthetic VRP environment. The first environment focuses on modeling the VRP in a simple and computationally lightweight way. It tracks customer visits and minimizes travel distance. The environment consists of a 50x50 grid with 10 customers initialized at random positions.

Basic Environment Results

Method	Avg. Distance	Avg. Run Time per Episode	Overall Avg. Reward
DQN	157.910713	0.003150	-90.21
MaxEnt	31.360598	0.002669	-70.07
SAC	41.974137	0.001650	-70.22
PPO	32.926397	0.007455	-69.96
Nearest Neighbor	259.521880	0.026442	N/A
OR-Tools	261.943091	0.437152	N/A

Table 1. *Comparison of average distance, average run time and average reward for different methods on Simple Environment*

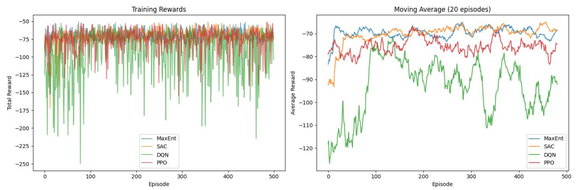

Figure 3. *Training rewards of RL models over 500 episodes on simple environment*

- **SAC and MaxEnt** achieve superior performance, with **MaxEnt** having the shortest average distance. Both methods balance efficient routing and reward optimization effectively.
- **DQN** performs worse than other RL methods in both distance and rewards, showing instability and difficulties in learning efficient policies in this simple environment.
- **PPO** achieves the best rewards and nearly lowest average distance, matching **MaxEnt**'s performance on the basic environment, but taking longer to run.
- Non-RL methods like **Nearest Neighbor** and **OR-Tools** have the worst distances, though they remain computationally efficient in this simpler setting.

Complex Environment Results

Method	Avg. Distance	Avg. Run Time per Episode	Overall Avg. Reward
DQN	92.924996	0.003150	-108.07
MaxEnt	88.078785	0.002669	-88.84
SAC	39.446166	0.001650	-103.28
PPO	165.027688	0.007455	-53.00
Nearest Neighbor	466.586744	0.026442	N/A
OR-Tools	404.791015	0.437152	N/A

Table 2. *Comparison of average distance, average run time and average reward for different methods on Complex Environment*

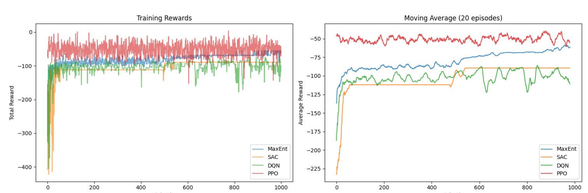

Figure 4. *Training rewards of RL models over 1000 episodes on complex environment*

RESULTS CONT.

The second portion of experiments were run on a more complicated synthetic VRP environment resembling the Solomon Benchmark dataset, consisting of a 50x50 grid, 100 customers, randomly generated demand between 1 to 30 units, and a vehicle capacity of 200. Our results show that the RL models based on PGM's outperform the traditional reinforcement learning algorithms as well as Operations Research based approaches. Table 2 shows the average distance traveled by each method as well as the average reward. PPO achieves best rewards but produces a longer average distance, showing a trade-off between exploration and route optimization. Figure 4 demonstrates the reward training curves of the RL-based approaches, showing that SAC and MaxEnt perform well compared to other traditional RL methods, and are slightly better than discrete RL models like PPO.

In our next steps we will be testing our models on the Solomon Benchmark. The Solomon C101 benchmark represents a real-world VRP scenario with tight time windows and capacity constraints. It consists of 100 customers with specific locations, demands, and strict service time windows. The benchmark is particularly challenging due to its clustered customer distribution and interdependent time-window constraints, making it a standard test for VRP algorithm effectiveness and efficiency.

CONCLUSION

Based on our results, our method of reformulating the maximization of a reward function as an inference problem in a graphical model outperforms traditional RL algorithms. The method is already used to devise more effective and powerful forward RL algorithms, to developing probabilistic algorithms for modeling and reasoning about observed goal-driven behavior.

Recently, an intersection of maximum entropy RL and latent variable models have been explored, where the graphical model for control as inference is augmented with additional variables for modeling **time-correlated stochasticity for exploration** or **higher-level control through learned latent action spaces**.

One area that could be explored more is the connection between maximum entropy reinforcement learning and robust control. Training a policy to achieve high expected reward under the highest possible amount of injected noise (highest entropy) should result in it being **robust to unexpected perturbations** at test time. A detailed theoretical exploration of this phenomenon could be applied more broadly to a range of challenging problems involving domain shift, unexpected perturbations, and model errors.

Also, the exploration of the relationship between probabilistic inference and control can demystify the design of reward functions, which often are assumed to be merely an extrinsic and unchanging signal by traditional RL algorithms. Reinterpretation of **rewards as log probability of some discrete event variable $\mathbf{O}_t$** can lead to more interpretable, more effective, and easier to specify reward functions and could lead to substantially more practical reinforcement learning methods in the future.

The probabilistic inference-based RL can be incorporated in a wide range of real-world challenges beyond VRP such as Bidding Strategy Optimization, and decision-making in Intensive Care Units (ICUs) that balance short-term patient survival with long-term recovery outcomes.

REFERENCES

Levine, S. (2018). Reinforcement Learning and Control as Probabilistic Inference: Tutorial and Review.

Haarnoja et al (2018). Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor

Liu, M., Liu, J., Hu, Z., Ge, Y. and Nie, X. (2022). Bid optimization using maximum entropy reinforcement learning.