Towards Topologically Diverse Probabilistic Planning Benchmarks

Synthetic Domain Generation for Markov Decision Processes

Jaël Champagne Gareau
Éric Beaudry
Vladimir Makarenkov

Computer Science Department
Université du Québec à Montréal

15 – 19 July 2024

Outline

Automated planning & scheduling

Machine Learning (ML)	*Natural Language Processing (NLP)*
Robotics	*Automated planning and Scheduling*
Expert Systems	...

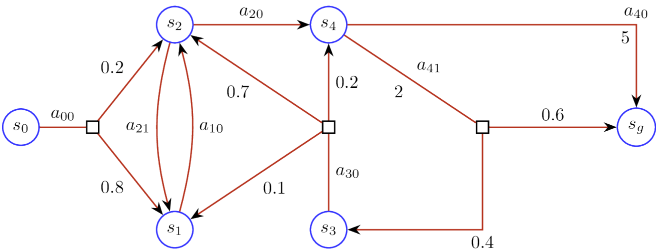
- **Automated planning & scheduling** is a branch of Artificial Intelligence.
- Its objective is to find **plans** allowing **agents** to reach **goals**.
- Some planning problems are **probabilistic** (i.e., there are **uncertainties**) :
  - endogenous uncertainties (i.e., due to the agent) ;
  - exogenous uncertainties (i.e., due to the environment).
- **Markov Decision Processes** (MDPs) are often used to model these problems of decision-making under uncertainty.

Mathematical MDP representation

Stochastic Shortest-Path MDP

An **(SSP) MDP** is a tuple (S, A, T, C, G) where :

- S is the finite set of **states** ;
- A is the finite set of **actions** that the agent can execute ;
- $T : S \times A \times S \to [0, 1]$ is the **transition function**, where $T(s, a, s')$ gives the probability that the agent reaches state s' if it executes action a at state s ;
- $C : S \times A \times S \to \mathbb{R}^+$ is the **cost function** where $C(s, a, s')$ gives the cost an agent must pay if it reaches state s' when executing action a at state s ;
- $G \subseteq S$ is the set of **goal states**.



Existing algorithms

Objective

Find a **policy** $\pi : S \rightarrow A$ that minimizes the expected total cost to reach a goal.

Classical algorithms

- Value Iteration (VI) [1]
- Policy Iteration (PI) [2]

Prioritization methods

- Generalized Prioritized Sweeping (genPS) [3]
- Partitioned, Prioritized, Parallel Value Iteration (P3VI) [4]

1. Bellman, R. (1957). Dynamic Programming. Prentice Hall.
2. Howard, R. A. (1960). Dynamic Programming and Markov Processes. John Wiley.
3. Andre, D. et al. (1998). Generalized prioritized sweeping. Proceedings of the 10th International Conference on Neural Information Processing Systems (p. 1001-1007). MIT Press.
4. Wingate, D. and Seppi, K. D. (2005). Prioritization methods for accelerating MDP solvers. Journal of Machine Learning Research, 6, 851-881.

Existing algorithms

Objective

Find a **policy** $\pi : S \rightarrow A$ that minimizes the expected total cost to reach a goal.

Heuristic approaches

- Labeled Real-Time Dynamic Programming (LRTDP) [5]
- Improved Looped And/Or* (ILAO*) [6]

Topological approaches

- Topological Value Iteration (TVI) [7]
- Parallel-Chained Topological Value Iteration (pcTVI) [8]

5. Bonet, B. and Geffner, H. (2003). Improving the Convergence of Real-Time Dynamic Programming. Proceedings of the 13th International Conference on Automated Planning and Scheduling (ICAPS 2003) (vol. 3, p. 12-21).
6. Hansen, E. A. and Zilberstein, S. (2001). LAO* : A heuristic search algorithm that finds solutions with loops. Artificial Intelligence, 129(1-2), 35-62.
7. Dai, P. et al. (2011). Topological value iteration algorithms. Journal of Artificial Intelligence Research, 42, 181-209.
8. Champagne Gareau, J. et al. (2023). pcTVI : Parallel MDP solver using a decomposition into independent chains. Classification and data science in the digital age (p. 101-109). Springer International Publishing.

Given a certain planning domain, which algorithm is faster ?

- For certain domains, we already know the answer :
  - Dense MDPs (actions can lead to a large set of states) : **VI** and **PI** are often the best ;
  - MDPs having a large number of goal states : **heuristic approaches** are often the best.
  - MDPs having a large number of strongly connected components : **topological approaches** are often the best.
- What if we have a combination of the above features ?

Open research problem

"[M]ore theory is needed to guide the development and selection of such enhancements. The most useful would be problem features and optimality definitions that would indicate which metric, reordering method and partitioning scheme are maximally effective, and which would guide the development of new enhancements. These may include distributional properties of the reward functions, distributional properties of transition matrices, strongly/weakly connected component analyses, etc."[9]

9. Wingate, D. and Seppi, K. D. (2005). Prioritization methods for accelerating MDP solvers. Journal of Machine Learning Research, 6, 851-881.

Lack of standardized benchmarks when evaluating a new algorithm

Open research problem

- The closest thing to standardized probabilistic planning benchmark domains are those used in the International Probabilistic Planning Competition (IPPC).
- Their number is relatively small.
- They are mostly domains for finite horizon and infinite horizon problems, instead of stochastic shortest-path problems.
- They are not designed to cover the complete list of features of MDPs that can influence the performance of the algorithms.

Objective

- We need a way to generate a large set of MDPs with different features.
  - They could serve as training data to train a classifier.
  - They could serve as benchmarks to evaluate new algorithms.
- Which features should we consider ?
- How can we generate a large set of synthetic MDPs with different features ?

Features of interest

- The **number of states** $|S|$ in the MDP.
- The **number of actions** $|A|$ in the MDP.
- The **number of goal states** $|G|$ in the MDP.

Features of interest

- The **number of states** $|S|$ in the MDP.
- The **number of actions** $|A|$ in the MDP.
- The **number of goal states** $|G|$ in the MDP.
- The **number of Strongly Connected Components (SCCs)** $|\mathfrak{S}|$ in the MDP.
- The **number of states in the largest SCC** $\max_{\mathcal{S} \in \mathfrak{S}} |\mathcal{S}|$.

Features of interest

- The **number of states** $|S|$ in the MDP.
- The **number of actions** $|A|$ in the MDP.
- The **number of goal states** $|G|$ in the MDP.
- The **number of Strongly Connected Components (SCCs)** $|\mathfrak{S}|$ in the MDP.
- The **number of states in the largest SCC** $\max_{\mathcal{S} \in \mathfrak{S}} |\mathcal{S}|$.
- The **distribution of actions** :
  $\forall k, P_k^a :=$ proportion of states which have k applicable actions.
- The **distribution of probabilistic transitions** :
  $\forall k, P_k^t :=$ proportion of actions which have k probabilistic transitions.

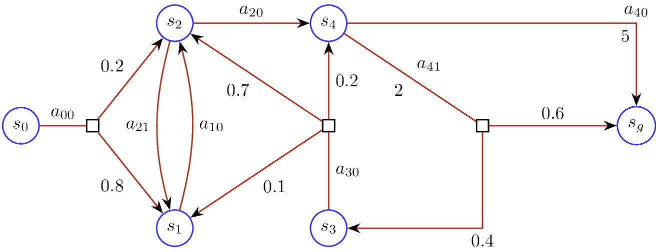
Features of interest

- The **number of states** $|S|$ in the MDP.
- The **number of actions** $|A|$ in the MDP.
- The **number of goal states** $|G|$ in the MDP.
- The **number of Strongly Connected Components (SCCs)** $|\mathfrak{S}|$ in the MDP.
- The **number of states in the largest SCC** $\max_{\mathcal{S}\in\mathfrak{S}} |\mathcal{S}|$.
- The **distribution of actions** :
  $\forall k, P^a_k :=$ proportion of states which have k applicable actions.
- The **distribution of probabilistic transitions** :
  $\forall k, P^t_k :=$ proportion of actions which have k probabilistic transitions.
- The **clustering coefficient** : $\mathfrak{C} := \frac{1}{|S|} \sum_{s\in S} \frac{e_s}{k_s(k_s-1)}$, where e_s is the number of pairs of states directly reachable from s that are also directly reachable from each other, and k_s is the number of states directly reachable from s. Moreover, $\mathfrak{C}$ is set to be 0 when $k_s < 2$.

Features of interest

- The **number of states** $|S|$ in the MDP.
- The **number of actions** $|A|$ in the MDP.
- The **number of goal states** $|G|$ in the MDP.
- The **number of Strongly Connected Components (SCCs)** $|\mathfrak{S}|$ in the MDP.
- The **number of states in the largest SCC** $\max_{\mathcal{S} \in \mathfrak{S}} |\mathcal{S}|$.
- The **distribution of actions** :
  $\forall k, P_k^a :=$ proportion of states which have k applicable actions.
- The **distribution of probabilistic transitions** :
  $\forall k, P_k^t :=$ proportion of actions which have k probabilistic transitions.
- The **clustering coefficient** : $\mathfrak{C} := \frac{1}{|S|} \sum_{s \in S} \frac{e_s}{k_s(k_s-1)}$, where e_s is the number of pairs of states directly reachable from s that are also directly reachable from each other, and k_s is the number of states directly reachable from s. Moreover, $\mathfrak{C}$ is set to be 0 when $k_s < 2$.
- The **goals-eccentricity** of the MDP : $\mathcal{G} := \min_{g \in G} \max_{s \in S} \bar{d}(s, g)$, where $\bar{d}(s, g)$ is the minimum number of actions (the cost of each action is not considered) that must be executed to reach g from s.

Example



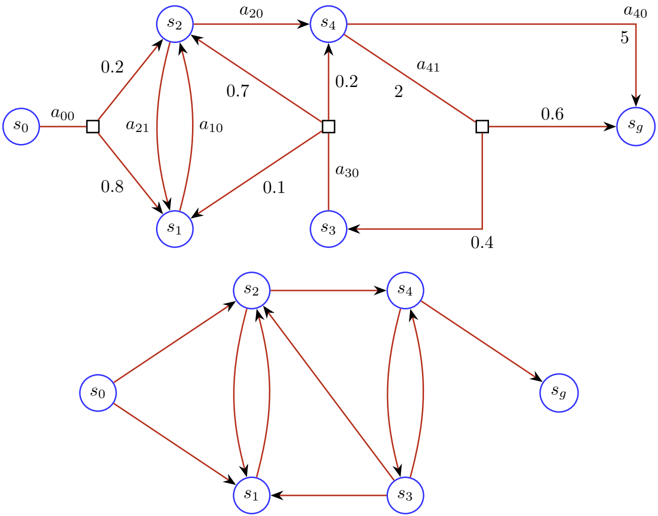
- $|S| = 6$, $|A| = 7$, $|G| = 1$;
- $\mathfrak{S} = \{\{s_0\}, \{s_1, s_2, s_3, s_4\}, \{s_g\}\}$;
- $\mathbf{P^a} = [\frac{1}{6}, \frac{3}{6}, \frac{2}{6}]$;
- $\mathbf{P^t} = [0, \frac{4}{7}, \frac{2}{7}, \frac{1}{7}]$;
- $\mathfrak{C} = \frac{1}{6}(\frac{2}{2\cdot 1} + 0 + \frac{0}{2\cdot 1} + \frac{3}{3\cdot 2} + 0 + 0) = \frac{1}{4}$;
- $\mathcal{G} = 3$.

Synthetic Graphs Generation

- A small number of synthetic MDP planning domains exist, e.g. :
  - Layered MDPs (used to control the number of SCCs) ;
  - Chained MDPs (used to control the number of independent chains of states).
- In comparison, there are a lot more synthetic graph generation methods.

Technique	Degrees Distr.	Clust. Coeff.	Diameter
Erdös-Rényi	Binomial	small $(\bar{k}/n)$	small : $\mathcal{O}(\log(n))$
Watts-Strogatz	Almost-constant	large	small
Barabási–Albert	Scale-free $(\bar{k}^{-3})$	large $(\bar{k}^{-1})$	small : $\mathcal{O}(\frac{\log(n)}{\log(\log(n))})$
Kronecker	Multinomial	flexible	flexible

Determinization of an MDP



Synthetic MDP generation

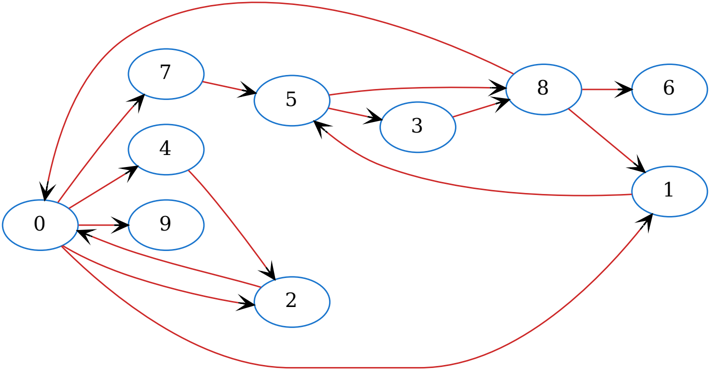
Algorithm Synthetic MDP Generation

Require: A list of desired topo. prop. (e.g., n : number of states ; k : number of goals, etc.)
Ensure: An MDP (S, A, T, C, G)

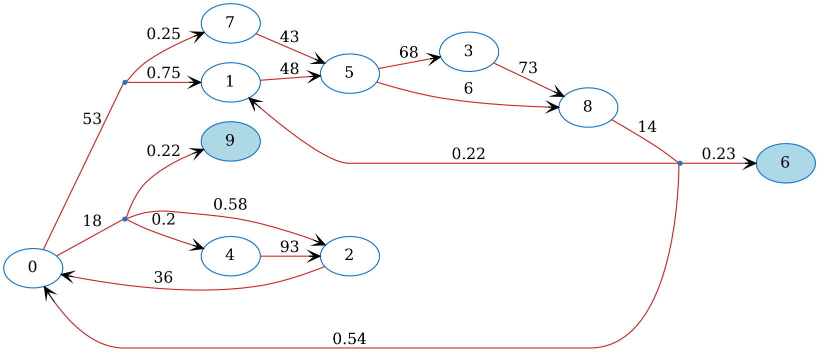
▷ Use the most appropriate graph gen. technique relative to the desired topological properties
$\Gamma \leftarrow$ GenerateSyntheticGraph(n) ▷ e.g., using any graph generation techniques
$S \leftarrow \Gamma$.getStates() ▷ $|S| = n$

for all $s \in S$ **do**
    $a_s \leftarrow$ RandomInt($1, k_s$) ▷ Generate the number of actions ; k_s is the degree of s
    $A_s \leftarrow$ DecompIntoSum(k_s, a_s) ▷ A_s is an array of a_s elements s.t. $\sum_{n_a \in A_s} n_a = k_s$
    for all $n_a \in A_s$ **do** ▷ n_a is the number of possible transitions of the current action
        $a \leftarrow$ new action identifier
        $A \leftarrow A \cup \{a\}$
        $C(s, a) \leftarrow$ RandomCost() ▷ Can be sampled uniformly or with another distribution
        $P_a \leftarrow$ GenProbabilities(n_a) ▷ P_a is an array s.t. $\sum_{p \in P_a} p = 1.0$ and $|P_a| = n_a$
        for all $i \in [1..n_a]$ **do**
            $s' \leftarrow$ RandomNeighbor(Γ, s) ▷ Random neighbor of s in the graph Γ
            $T(s, a, s') \leftarrow P_a[i]$
$G \leftarrow$ RandomSubset(S, k) ▷ k is a parameter to control the number of goal states
return (S, A, T, C, G)

Erdös-Rényi Synthetic Graph



Corresponding Synthetic MDP



Implementation

- We implemented a project in C++, called `graph-toolkit`, which contains :
  - the four aforementioned synthetic graph models ;
  - the synthetic MDP generation algorithm ;
  - different functions to find the topological properties of the generated graphs/MDPs.
- The project is available on GitLab [10].

10. `https://gitlab.info.uqam.ca/champagne_gareau.jael/graph-toolkit`

Conclusion

- We proposed a list of features that could be used to classify MDPs.
- We proposed a method to generate synthetic MDPs which can cover the different combination of features of interest.
- As future work, we plan on using these synthetic MDPs to train a fast classifier that can predict which MDP planner will be the fastest for a given MDP.

Acknowledgments

We acknowledge the support of the *Natural Sciences and Engineering Council of Canada* (NSERC) and the *Fonds de recherche du Québec — Nature et technologies* (FRQNT).