Towards Topologically Diverse Probabilistic Planning Benchmarks: Synthetic Domain Generation for Markov Decision Processes

Jaël Champagne Gareau, Éric Beaudry, and Vladimir Makarenkov

Abstract Markov Decision Processes (MDPs) are often used in Artificial Intelligence to solve probabilistic sequential decision-making problems. In the last decades, many probabilistic planning algorithms have been developed to solve MDPs. However, the lack of standardized benchmarks makes it difficult to compare the performance of these algorithms in different contexts. In this paper, we identify important topological properties of MDPs that can make a significant impact on the relative performance of probabilistic planning algorithms. We also propose a new approach to generate synthetic MDP domains having different topological properties. This approach relies on the connection between MDPs and graphs and allows every graph generation technique to be used to generate synthetic MDP domains.

Key words: Markov decision process, probabilistic planning, synthetic domains generation, topological diversity, benchmarking

Jaël Champagne Gareau✉
Université du Québec à Montréal, QC, Canada, e-mail: champagne_gareau.jael@uqam.ca

Éric Beaudry
Université du Québec à Montréal, QC, Canada, e-mail: beaudry.eric@uqam.ca

Vladimir Makarenkov
Université du Québec à Montréal, QC, Canada, e-mail: makarenkov.vladimir@uqam.ca

1 Introduction

In Artificial Intelligence, problems of sequential decision-making under uncertainty are often modeled using Markov Decision Processes (MDPs). In the last decades, many new probabilistic planning algorithms have been developed to find optimal solutions for MDP instances. Some of these algorithms are especially good in specific contexts, when, for example, the MDP of interest contains a large number of Strongly Connected Components (SCCs) in its transition graph or when there exists a trajectory to a goal state using a small number of actions [5].

Usually, new planning algorithms intended to solve MDPs proposed in the literature are evaluated on a small number of carefully designed domains to demonstrate their efficiency. However, the lack of standardized benchmarks makes it difficult to compare the performance of these algorithms in different contexts. For example, we know that some algorithms (e.g., Topological Value Iteration [5]) are good for solving MDPs with a large number of SCCs, whereas others (e.g., Labeled Real-Time Dynamic Programming [4]) are better for solving MDPs containing a large number of goal states inside its state space. However, we do not know *a priori* which of these algorithms is better for solving MDPs that have both a large number of SCCs and a large number of goal states. Since there are no currently existing benchmarks that contain MDPs with both of these properties, it is difficult to know which algorithm will be the most efficient in this context.

The domains that are the closest to standardized domains for probabilistic planning algorithms are those used in the International Planning Competition, which is organized in the context of the International Conference on Automated Planning and Scheduling (ICAPS) [11]. Even though a few planning domains have been added during the last occurence of the competition, their total number is still relatively small and does not cover the entire range of combination of topological properties one might be interested in. Moreover, the domains used in the competition are mostly designed to evaluate finite horizon MDPs and infinite horizon discounted MDPs, whereas in this research, we are mostly interested in domains related to Stochastic Shortest Path MDPs (SSP-MDPs). The lacking of standardized benchmarks for SSP-MDPs as been highlighted as an important issue in the literature:

> [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 partitionning 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. [13]

SSP-MDPs are known to be more general than other common types of MDPs [3]. They can be viewed as a generalization of the problem of finding a shortest path in a graph with probabilistic transitions. More formally, an SSP-MDP is defined as a tuple (S, A, T, C, G), where S is a finite set of states, A is a finite set of actions, $T\colon S \times A \times S \to [0,1]$ is a transition function, $C\colon S \times A \to \mathbb{R}^+$ is a cost function and $G \subseteq S$ is a set of goal states. The objective is to find a policy $\pi\colon S \to A$ that minimizes the expected cost of reaching a goal when starting from any state in S.

Our main contributions in this paper are as follows:

- We provide a list of topological properties that we deem important to estimate the performance of probabilistic planning algorithms on SSP-MDPs.
- We propose a new approach to generate synthetic SSP-MDPs that can cover different topological properties of interest.

2 Topological Properties

In this section, we present a list of topological properties of MDPs, some of them are similar to graph properties, while the other are unique to MDPs. We believe that most of them can have a significant impact on the relative performance of probabilistic planning algorithms. Some of these properties can also be given as parameters to the synthetic MDP generation process we will describe in the next section. The list of properties, or model parameters, we propose is as follows:

- 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 $k_s < 2$.
- 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.

We explain these properties more precisely through the following example. The MDP in Figure 1 (top) contains 6 states, 7 actions, 1 goal state (s_g) and 3 SCCs, $\{\{s_0\},\{s_1,s_2,s_3,s_4\},\{s_g\}\}$. The largest SCC contains 4 states. Moreover, the distribution of actions is given by $\mathbf{P^a} = [\frac{1}{6},\frac{3}{6},\frac{2}{6}]$ and the distribution of probabilistic transitions is given by $\mathbf{P^t} = [0,\frac{4}{7},\frac{2}{7},\frac{1}{7}]$. The clustering coefficient is $\mathfrak{C} = \frac{1}{6}(\frac{2}{2\cdot1}+0+\frac{0}{2\cdot1}+\frac{3}{3\cdot2}+0+0) = \frac{1}{4}$ and the goals-eccentricity is $\mathcal{G} = 3$, since it takes at least 3 actions to reach s_g from s_0.

3 Synthetic Domain Generation

Some existing MDP planning domains are synthetic, in the sense that they are not directly mapped into a real-world domain, but are designed to measure how the change in one particular topological aspect of the MDP can affect the relative performance of existing MDP planners. For example, the Layered [5] and the Chained [8] domains were designed specifically to measure, respectively, the impact of the number of SCCs and the impact of their relative placement in "independent chains" of SCCs on the performance of several planning algorithms. However, these domains are limited in the sense that they only cover a small subset of possible combinations of topological properties we would like to compare. Moreover, the process of designing synthetic domains is time-consuming. Therefore, in this section, we propose to leverage the connection between MDPs and graphs to generate synthetic MDPs using existing graph generation techniques.

Our synthetic MDP generation technique is inspired by the concept of *all-outcomes determinization*. It consists in finding a graph from an MDP, where there is an arc for every possible outcome of each action. MDP determinization was originally proposed as a way to solve MDPs using deterministic planning algorithms [14]. Figure 1 shows an example of such a determinization.

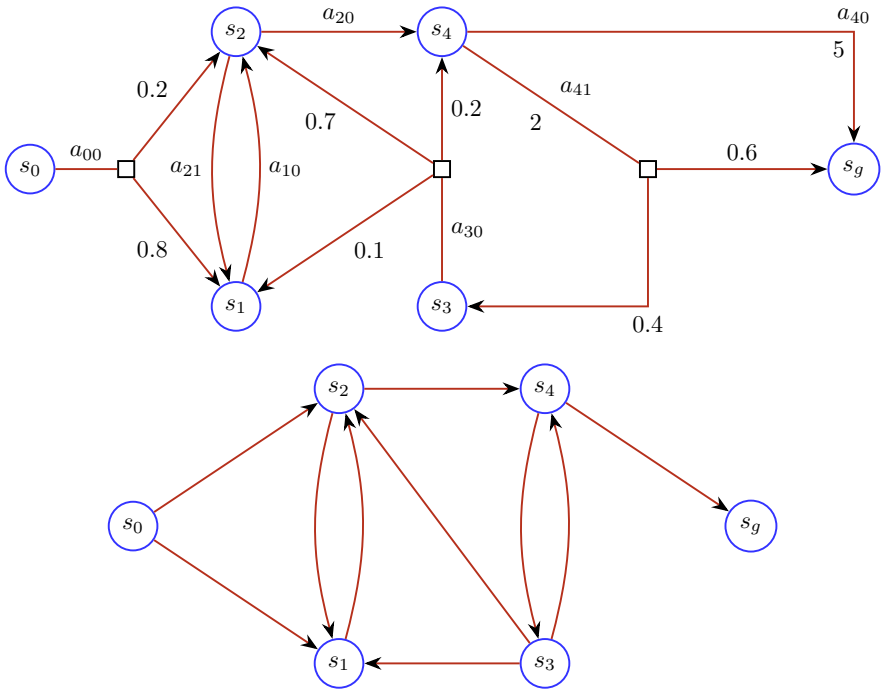

Fig. 1 An MDP (top) and the graph corresponding to its all-outcomes determinization (bottom).

The graph resulting from a determinization can also be used to find and analyze topological properties of the original MDP. For example, the clustering coefficient of an MDP, as defined above, is equivalent to the clustering coefficient of its corresponding graph. Other topological properties, such as the number of SCCs, are also equivalent in the MDP and in its corresponding graph. However, some properties, such as the distribution of probabilistic transitions, have no equivalence in graph theory and must be computed directly from the MDP.

MDP determinization allows us to generate a graph from an MDP. The key idea behind our proposed synthetic MDP generation technique is to reverse this process by generating an MDP from a graph. This allows us to use existing graph generation techniques to create synthetic MDPs. We can then use the graph properties to control some of the topological properties of the generated MDPs. Table 1 shows some examples of graph generation techniques and their respective properties.

Table 1 Examples of graph generation techniques and their respective properties, where $\bar{k}$ is the average degree of the nodes in the graph, and n is the number of nodes.

Technique	**Ref.**	**Degrees Distr.**	**Clust. Coeff.**	**Diameter**
Erd o s-Rényi	[6]	Binomial	small ($\bar{k}/n$)	small: $O(\log(n))$
Watts-Strogatz	[12]	Almost-constant	large	small
Barabási–Albert	[1]	Scale–free ($\bar{k}^{-3}$)	large ($\bar{k}^{-1}$)	small: $O(\frac{\log(n)}{\log(\log(n))})$
Kronecker	[10]	Multinomial	flexible	flexible

Our approach starts by generating a graph using one of the techniques presented in Table 1. The choice of the technique depends on the desired topological properties of the MDP. For example, if we want to generate an MDP with a small clustering coefficient, we can use the Erdös-Rényi model. The second step is to use this graph as a base for generating the MDP. For every state s in the MDP (which corresponds to a node in the graph), we generate a_s actions, where a_s is a random number ranging between 1 and the degree k_s of the node s in the graph. We then generate an array which consists of a_s random numbers such that their sum is equal to k_s. For example, if a given node has a degree of 8, and the random number of actions is 3, a possible array could be $[4, 1, 3]$. This array represents the number of states that can be reached by applying each of the actions. The next step consists in generating a cost for each action (any distribution can be used here), a probability for each possible transition (normalized to 1) and a state corresponding to each possible probabilistic transition of each action (among all neighbors of the node in the graph). Finally, the goal states are chosen among the set of states. Algorithm 1 shows the main steps of the proposed approach.

Algorithm 1 and the four graph generation techniques presented in Table 1 have been implemented in C++. The resulting graph library as well as an accompanying program (which can analyze and generate synthetic graphs and corresponding syn-

Algorithm 1 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 one of the techniques in Table 1
$S \leftarrow \Gamma$.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)

thetic MDPs) is available publicly on GitLab[1]. Figures 2 and 3 show an example of a synthetic graph generated using the Erdös-Rényi model ($n = 10$ and $m = 15$), and the corresponding synthetic MDP generated using Algorithm 1.

Our algorithm has the advantage of being simple to implement, fast to execute and flexible. It can be used to generate a wide variety of synthetic MDPs. One weakness of our approach is that the choice of the underlying graph generation technique must currently be done manually by the user. We would like to eventually develop a method to automatically select the most appropriate graph generation technique based on the desired topological properties of the MDP.

4 Conclusion

In this paper, we have identified important topological properties of MDPs that can make a significant impact in the performance of probabilistic planning algorithms. We have also proposed a new approach to generate synthetic MDPs having different topological properties. This approach relies on the connection between MDPs and graphs and allows any graph generation technique to be used as a basis to generate synthetic MDPs. We believe that this approach will allow one to generate a wide variety of synthetic MDPs, which will be useful to compare the performance of probabilistic planning algorithms in different practical contexts. As future work, we plan to generate a wide range of synthetic MDPs using this approach and evaluate the performance of existing probabilistic planning algorithms applied to these MDPs.

[1] https://gitlab.info.uqam.ca/champagne_gareau.jael/graph-toolkit

Using these results, we plan on training a classification model, where the input will be the topological properties of the MDP and the output will be the most efficient algorithm to solve it. Using this classifier, we will be able to predict the most efficient algorithm to solve a given MDP based on its topological properties.

Acknowledgements This research has been supported by the *Natural Sciences and Engineering Research Council of Canada* (NSERC) and the *Fonds de Recherche du Québec — Nature et Technologies* (FRQNT).

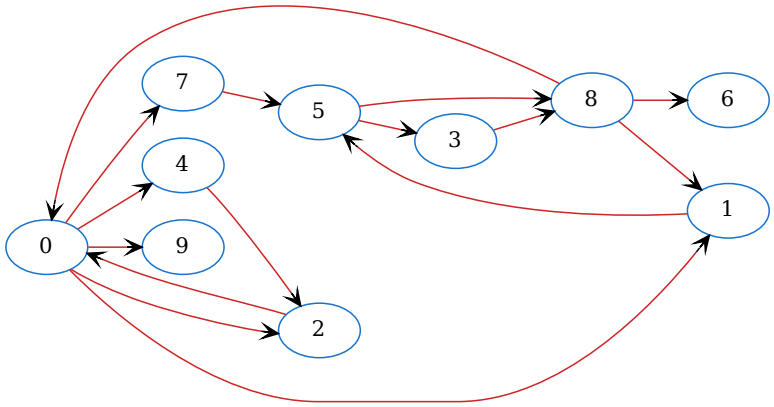

Fig. 2 Example of a synthetic graph generated using the ErdÃ¶s-Rényi model with $n = 10$ and $M = 15$. The graph was generated using our `graph-toolkit` C++ library.

References

1. Barabási, A.-L., Albert, R.: Emergence of Scaling in Random Networks. Science **286**(5439), 509–512 (1999)
2. Bellman, R.: Dynamic Programming. Prentice Hall, New York (1957)
3. Bertsekas, D.: Dynamic Programming and Optimal Control. Athena scientific Belmont, MA (1995)
4. Bonet, B., Geffner, H.: Labeled RTDP: Improving the convergence of real-time dynamic programming. In: Proc. of the 13th International Conference on Automated Planning and Scheduling (ICAPS 2003), pp. 12–21. AAAI Press, Trento (2003)
5. Dai, P., Mausam, Weld, D.S., Goldsmith, J.: Topological value iteration algorithms. Journal of Artificial Intelligence Research **42**, 181–209 (2011)
6. Erdos, P., Rényi, A.: On Random Graphs I. Publicationes Mathematicae **6**, 290–297 (1959)

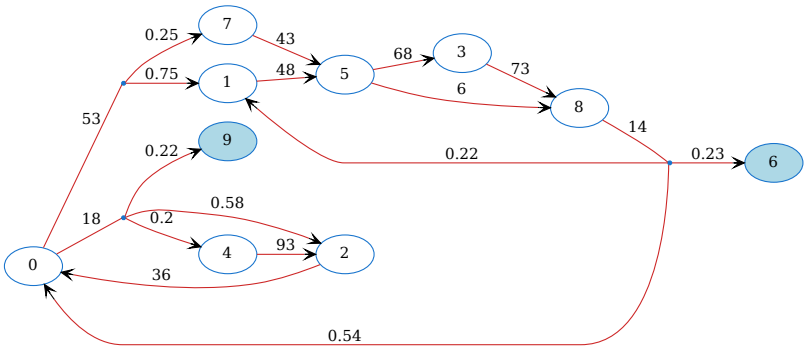

Fig. 3 Example of a synthetic MDP generated using Algorithm 1 on the graph of Figure 2. The distributions used to generate the costs of the actions and the probabilities of the transitions are both uniform, respectively $U(0, 100)$ and $U(0, 1)$. Two goal states have been generated: 6 and 9.

7. Champagne Gareau, J., Gosset, G., Beaudry, é., Makarenkov, V.: Cache-Efficient Dynamic Programming MDP Solver. In: Proc. of the 26th European Conference on Artificial Intelligence (ECAI 2023), pp. 373–380. IOS Press, Krakow (2023)
8. Champagne Gareau, J., Beaudry, é., Makarenkov, V.: pcTVI: Parallel MDP Solver Using a Decomposition Into Independent Chains. In: Brito, P., Dias, J.G., Lausen, B., Montanari, A., Nugent, R. (eds) Classification and Data Science in the Digital Age. IFCS 2022. Studies in Classification, Data Analysis, and Knowledge Organization. Springer, Cham (2022)
9. Hansen, E.A., Zilberstein, S.: LAO*: A heuristic search algorithm that finds solutions with loops. Artificial Intelligence **129**, 35–62 (2001)
10. Leskovec, J., Chakrabarti, D., Kleinberg, J., Faloutsos, C., Ghahramani, Z.: Kronecker Graphs: An Approach to Modeling Networks. Journal of Machine Learning Research **11**(2), 985–1042 (2010)
11. Vallati, M., Chrpa, L., Grzes, M., McCluskey, T.L., Roberts, M., Sanner, S.: The 2014 International Planning Competition: Progress and Trends. AI Magazine **36**(3), 90–98 (2015)
12. Watts, D.J., Strogatz, S.H.: Collective Dynamics of 'Small-World' Networks. Nature **393**(6684), 440–442 (1998)
13. Wingate, D., Seppi, K.D.: Prioritization methods for accelerating MDP solvers. Journal of Machine Learning Research **6**, 851–881 (2005)
14. Yoon, S., Fern, A., Givan, R.: FF-Replan: A Baseline for Probabilistic Planning. In: Proc. of the 17th International Conference on Automated Planning and Scheduling (ICAPS 2007), pp. 352–359. AAAI Press, Providence (2007)