MDPSolver: An Efficient Solver for Markov Decision Processes

Anders Reenberg Andersen [1*] **and Jesper Fink Andersen** [1*]

1 Department of Applied Mathematics and Computer Science, Technical University of Denmark, Denmark ***** These authors contributed equally.

DOI: 10.21105/joss.07544

Software

- Review
- Repository
- Archive

Editor: Kelly Rowland

Reviewers:

- @gartavanis
- @braniii

Submitted: 05 August 2024
Published: 15 May 2025

Summary

MDPSolver is a Python package for easy and fast optimization of Markov Decision Process (MDP) problems. Our package is relevant to any decision-making problem where sequences of actions need to be taken and the outcome of each action is uncertain and revealed to the decision-maker one at a time. This specific type of decision-making problem is often denoted as an MDP or a discrete stochastic dynamic programming problem. For example, consider the sale and replenishment of goods from a wholesale inventory. Once a day, the wholesaler observes the amount of goods in the inventory and decides whether to create a replenishment order. Creating the order too early leads to an excess amount of goods, and unnecessary holding costs, whereas creating the order too late leads to shortage and lost sales. With MDPSolver, the wholesaler will be able to maximize profit by deriving optimized actions for each level of the remaining goods in the inventory.

Statement of need

The industry currently experiences an increasing availability of data and computational resources. This development has created an opportunity for optimized decision-making based on mathematical modeling. Conversely, computational implementations can be costly, opening the door to software packages containing efficient and easily accessible tools. MDPSolver addresses this need by providing the accessibility of algorithms, optimality criteria, and other configurations. MDPSolver is highly applicable as a research tool since users can quickly employ and test various models and solution approaches. Due to the straightforward applicability of MDP models in decision-making sciences, such as operations research, MDPSolver has an important role in achieving any goals related to optimized decision-making under uncertainty.

Several similar software packages are available for Python (*Pymdptoolbox*, 2015), R (Chades et al., 2017), Matlab (Cros, 2015), and Julia (Egorov et al., 2017) (see the section on related software below). MDPSolver targets Python and improves on the *pymdptoolbox* package (*Pymdptoolbox*, 2015) by providing a parallelized efficient implementation of algorithms from MDP theory (Puterman, 1994). Specifically, MDPSolver uses OpenMP (*The OpenMP API Specification for Parallel Programming*, n.d.), ensuring that large vector validations are distributed across multiple threads and executed on multiple cores in parallel. Our numerical experiments show that our implementation results in a substantially shorter runtime (*MDPSolver Wiki*, n.d.). Moreover, MDPSolver provides the user with the option to experiment with various value update methods that are not available in the *pymdptoolbox* package. In an upcoming version, MDPSolver will also contain a selection of built-in MDP problems often considered in the scientific literature, such as multi-component replacement, job scheduling, and inventory management. The built-in problems will enable users to evaluate even larger models by computing the model parameters (e.g. transition probabilities) on demand.

In our past research, Andersen (2022) and Andersen et al. (2022) used earlier versions of MDPSolver for conducting numerical experiments on multi-component replacement problems. In the future, we are planning on utilizing MDPSolver for conducting experiments in our research on the optimization of hospital patient flow.

Features

MDPSolver enables users to derive epsilon-optimal policies for infinite horizon MDP models in Python. Users can choose to derive the policy using the value iteration, policy iteration, or modified policy iteration algorithm with the discounted or average reward optimality criterion. Furthermore, MDPSolver enables users to choose between standard, Gauss-Seidel, and Successive Over-Relaxation value updates. Choosing the standard value updates will activate parallel computing by default. In addition, to conserve computational memory, we have implemented three input options for sparse transition probability matrices, including the option to use the full matrix for teaching purposes. MDPSolver can read and write parameters and results directly from and to text files on the computer.

Related software

The following list contains related software packages that are available for the Python, R, Matlab, and Julia languages.

- Python: **pymdptoolbox** (*Pymdptoolbox*, 2015). Available on the *Python Package Index* (PyPI).
- R: **MDPtoolbox** (Chades et al., 2017). Available on the *Comprehensive R Archive Network* (CRAN).
- Matlab: **Markov Decision Processes (MDP) Toolbox** (Cros, 2015). Available on the *Matlab File Exchange*. See also the built-in **createMDP** function (*createMDP (Matlab)*, n.d.).
- Julia: **POMDPs.jl** (Egorov et al., 2017). Available on *Julia Packages*.

Acknowledgements

The authors appreciate the funding and resources provided by the Technical University of Denmark which led to the completion of MDPSolver's initial version.

References

Andersen, J. F. (2022). *Maintenance optimization for multi-component systems using Markov decision processes (Ph.D. Thesis)*. Technical University of Denmark.

Andersen, J. F., Andersen, A. R., Kulahci, M., & Nielsen, B. F. (2022). A numerical study of Markov decision process algorithms for multi-component replacement problems. *European Journal of Operational Research*, *299*(3), 94–102. https://doi.org/10.1016/j.ejor.2021.07.007

Chades, I., Chapron, G., Cros, M., Garcia, F., & Sabbadin, R. (2017). *MDPtoolbox: Markov decision processes toolbox*. The R Foundation. https://doi.org/10.32614/cran.package.mdptoolbox

createMDP (matlab). (n.d.). [Computer software]. MathWorks. Retrieved July 12, 2024, from https://se.mathworks.com/help/reinforcement-learning/ref/createmdp.html

Cros, M. (2015). *Markov decision processes (MDP) toolbox*. MathWorks. https://se.mathworks.com/matlabcentral/fileexchange/25786-markov-decision-processes-mdp-toolbox

Egorov, M., Sunberg, Z. N., Balaban, E., A., W. T., Gupta, J. K., & Kochenderfer, M. J. (2017). POMDPs.jl: A framework for sequential decision making under uncertainty. *Journal of Machine Learning Research*, *18*(26), 1–5. http://jmlr.org/papers/v18/16-300.html

MDPSolver wiki: Performance experiments. (n.d.). [Computer software]. MDPSolver Github. Retrieved February 15, 2025, from https://github.com/areenberg/MDPSolver/wiki/Performance-experiments

Puterman, M. L. (1994). (pp. 1–649). Wiley. https://doi.org/10.1002/9780470316887

Pymdptoolbox. (2015). [Computer software]. Python Software Foundation. https://pypi.org/project/pymdptoolbox/

The OpenMP API specification for parallel programming. (n.d.). [Computer software]. The OpenMP Architecture Review Board. Retrieved February 15, 2025, from https://www.openmp.org/