DTMC Benchmark Generation*

Mark van Wijk
University of Twente
Enschede, The Netherlands
mark.vanwijk@utwente.nl

Probabilistic model checking (PMC) is used to verify safety-critical systems in uncertain environments. When neither continuous time nor nondeterminism is relevant in these systems, they can be modelled using discrete-time Markov chains (DTMCs). In the space domain, DTMCs can be applied for dependability analysis [5, 14, 18] to model the system and using PMC to analyse the effects of uncertainty.

Given the fundamental importance of DTMCs, a variety of algorithms exists to analyse them, such as interval iteration [3], optimistic value iteration [12], and sound value iteration [19]. Moreover, these algorithms have been implemented in a variety of tools, including PRISM [17], STORM [15], MCSTA [10], and UPPAAL [4].

With such a wide variety of analysis options, one might wonder about the differences in performance between these tools and algorithms. Several studies have been conducted to determine the differences between these algorithms and tools, with most of them using the Quantitative Verification Benchmark Set [13] (QVBS). QVBS contain only contains 10 DTMC models, which is quite limited for such a fundamental type of model. Moreover, this raises questions regarding the variety available when benchmarking and testing DTMC algorithms and tools. For Markov decision processes, for example, an earlier investigation has shown that QVBS did not include any models with non-trivial end components [11].

An obvious solution to this problem is to generate DTMCs randomly. However, PMC might provide incorrect results [1, 21], which would go unnoticed if the actual result is unknown. Moreover, it has been shown that for totally random models, the results of properties may converge in the non-probabilistic context [6], which allows for almost certain model checking without inspecting the model at all.

To overcome this problem, we introduce a method to generate DTMCs where the result of each problem is known by construction, as is also used for LTL benchmarking [20]. This allows us to benchmark PMC tools and algorithms and test for correctness while keeping the approach sufficiently general to generate any DTMC. Furthermore, we make the generation configurable by guiding the generation towards properties such as the branch factor and strongly connected components. This allows for more thorough investigations of algorithms and tools, since it could be that one algorithm outperforms another only on models with a high branching factor.

Many PMC tools use double precision to represent probabilities, which means that these tools cannot represent each real number exactly [16]. This leads to rounding errors propagating throughout the PMC tools, potentially compromising the final results [9]. To ensure that the model and actual results can be represented without rounding errors, if required, we introduce a method to generate models which only contain rational probabilities with a fixed denominator.

The generated DTMCs are stored in the recently proposed UMB format [2] using the UMBI [8] Python library. This format allows explicit state spaces to be stored and is supported by several tools, including PRISM, STORM, and MCSTA, which allows our generated benchmarks to be used by all these tools.

*This work was supported by the European Union's Horizon 2020 research and innovation programme under Marie Skłodowska-Curie grant agreement 101008233 (MISSION), and by NWO VIDI grant VI.Vidi.223.110 (TruSTy).

Submitted to:
MISSION Workshop 2026

References

[1] R. Andriushchenko, A. Bork, C.E. Budde, M. Češka, K. Grover, E.M. Hahn, A. Hartmanns, B. Israelsen, N. Jansen, J. Jeppson, S. Junges, M.A. Köhl, B. Könighofer, J. Křetínský, T. Meggendorfer, D. Parker, S. Pranger, T. Quatmann, E. Ruijters, L. Taylor, M. Volk, M. Weininger & Z. Zhang (2024): *Tools at the Frontiers of Quantitative Verification (QComp 2023 Competition Report)*. In D. Beyer, A. Hartmanns & F. Kordon, editors: *TOOLympics Challenge 2023*, *Lecture Notes in Computer Science* 14550, Springer, pp. 90–146, doi:10.1007/978-3-031-67695-6_4.

[2] R. Andriushchenko, A. Hartmanns, J. Jeppson, S. Junges, T. Meggendorfer, D. Parker, T. Quatmann & M. Weininger (2026): *UMB: A Unified Markov Binary Format for Probabilistic Model Checking*. Under review, draft shared with the authors in private communication.

[3] C. Baier, J. Klein, L. Leuschner, D. Parker & S. Wunderlich (2017): *Ensuring the Reliability of Your Model Checker: Interval Iteration for Markov Decision Processes*. In Rupak Majumdar & Viktor Kuncak, editors: *29th International Conference on Computer Aided Verification (CAV 2017)*, *Lecture Notes in Computer Science* 10426, Springer, pp. 160–180, doi:10.1007/978-3-319-63387-9_8.

[4] J. Bengtsson, K.G. Larsen, F. Larsson, P. Pettersson & W. Yi (1995): *UPPAAL - a Tool Suite for Automatic Verification of Real-Time Systems*. In R. Alur, T.A. Henzinger & E.D. Sontag, editors: *Hybrid Systems III: Verification and Control, Proceedings of the DIMACS/SYCON Workshop on Verification and Control of Hybrid Systems, October 22-25, 1995, Ruttgers University, New Brunswick, NJ, USA*, *Lecture Notes in Computer Science* 1066, Springer, pp. 232–243, doi:10.1007/BFB0020949.

[5] M.A. Boyd (1998): *An Introduction to Markov Modeling: Concepts and Uses*. Available at `https://ntrs.nasa.gov/api/citations/20020050518/downloads/20020050518.pdf`.

[6] Y. Dong, M. Lopuhaä-Zwakenberg & M. Stoelinga (2025): *0-1 Laws for LTL and CTL over Random Transition Systems*. In Gidon Ernst & Kristin Yvonne Rozier, editors: *Model Checking Software - 31st International Symposium, SPIN 2025, Hamilton, ON, Canada, May 7-8, 2025, Proceedings*, *Lecture Notes in Computer Science* 15945, Springer, pp. 66–87, doi:10.1007/978-3-032-06847-7_4.

[7] C. Fu, E. Moritz Hahn, Y. Li, S. Schewe, M. Sun, A. Turrini & L. Zhang (2022): *EPMC Gets Knowledge in Multi-agent Systems*. In B. Finkbeiner & T. Wies, editors: *23rd International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI 2022)*, *Lecture Notes in Computer Science* 13182, Springer, pp. 93–107, doi:10.1007/978-3-030-94583-1_5.

[8] Github: *umbi*. Available at `https://github.com/pmc-tools/umbi`.

[9] A. Hartmanns (2022): *Correct Probabilistic Model Checking with Floating-Point Arithmetic*. In D. Fisman & G. Rosu, editors: *Tools and Algorithms for the Construction and Analysis of Systems - 28th International Conference, TACAS 2022, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2022, Munich, Germany, April 2-7, 2022, Proceedings, Part II*, *Lecture Notes in Computer Science* 13244, Springer, pp. 41–59, doi:10.1007/978-3-030-99527-0_3.

[10] A. Hartmanns & H. Hermanns (2014): *The Modest Toolset: An Integrated Environment for Quantitative Modelling and Verification*. In E. Ábrahám & K. Havelund, editors: *20th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2014)*, *Lecture Notes in Computer Science* 8413, Springer, pp. 593–598, doi:10.1007/978-3-642-54862-8_51.

[11] A. Hartmanns, S. Junges, T. Quatmann & M. Weininger (2024): *The Revised Practitioner's Guide to MDP Model Checking Algorithms*. In: *International Conference on Reachability Problems*. Available at `https://sjunges.github.io/files/revised_practitioners_guide.pdf`.

[12] A. Hartmanns & B.L. Kaminski (2020): *Optimistic Value Iteration*. In S. K. Lahiri & C. Wang, editors: *32nd International Conference on Computer Aided Verification (CAV 2020)*, *Lecture Notes in Computer Science* 12225, Springer, pp. 488–511, doi:10.1007/978-3-030-53291-8_26.

[13] A. Hartmanns, M. Klauck, D. Parker, T. Quatmann & E. Ruijters (2019): *The Quantitative Verification Benchmark Set*. In T. Vojnar & L. Zhang, editors: *Tools and Algorithms for the Construction and Analysis of Systems - 25th International Conference, TACAS 2019, Held as Part of the European Joint Conferences on

Theory and Practice of Software, ETAPS 2019, Prague, Czech Republic, April 6-11, 2019, Proceedings, Part I, *Lecture Notes in Computer Science* 11427, Springer, pp. 344–350, doi:10.1007/978-3-030-17462-0_20.

[14] B.R. Haverkort (2000): *Markovian Models for Performance and Dependability Evaluation*. In Ed Brinksma, Holger Hermanns & Joost-Pieter Katoen, editors: *Lectures on Formal Methods and Performance Analysis, First EEF/Euro Summer School on Trends in Computer Science, Berg en Dal, The Netherlands, July 3-7, 2000, Revised Lectures*, *Lecture Notes in Computer Science* 2090, Springer, pp. 38–83, doi:10.1007/3-540-44667-2_2.

[15] C. Hensel, S. Junges, J-P. Katoen, T. Quatmann & M. Volk (2022): *The probabilistic model checker Storm*. *Int. J. Softw. Tools Technol. Transf.* 24(4), pp. 589–610, doi:10.1007/S10009-021-00633-Z.

[16] IEEE (2019): *IEEE Standard for Floating-Point Arithmetic*. *IEEE Std 754-2019 (Revision of IEEE 754-2008)*, pp. 1–84, doi:10.1109/IEEESTD.2019.8766229.

[17] M. Z. Kwiatkowska, G. Norman & D. Parker (2011): *PRISM 4.0: Verification of Probabilistic Real-Time Systems*. In G. Gopalakrishnan & S. Qadeer, editors: *23rd International Conference on Computer Aided Verification (CAV 2011)*, *Lecture Notes in Computer Science* 6806, Springer, pp. 585–591, doi:10.1007/978-3-642-22110-1_47.

[18] C.J. Lowe & M. Macdonald (2020): *Space mission resilience with inter-satellite networking*. *Reliab. Eng. Syst. Saf.* 193, p. 106608, doi:10.1016/J.RESS.2019.106608.

[19] T. Quatmann & J-P. Katoen (2018): *Sound Value Iteration*. In H. Chockler & G. Weissenbacher, editors: *30th International Conference on Computer Aided Verification (CAV 2018)*, *Lecture Notes in Computer Science* 10981, Springer, pp. 643–661, doi:10.1007/978-3-319-96145-3_37.

[20] B. Steffen, M. Isberner, S. Naujokat, T. Margaria & M. Geske (2014): *Property-driven benchmark generation: synthesizing programs of realistic structure*. *Int. J. Softw. Tools Technol. Transf.* 16(5), pp. 465–479, doi:10.1007/S10009-014-0336-Z.

[21] M. van Wijk (2024): *The State of Multi-Objective Model Checking*. Master's thesis, University of Twente. Available at `https://purl.utwente.nl/essays/104622`.