Dynamic Programming via the Randomized Mini-Batch Gauss-Seidel Operator

Matilde Gargiani, Andrea Martinelli
SA/MA Project

December 15, 2020

---

1 Posting

Optimal control theory is a branch of mathematical optimization that focuses on the design of a controller for a dynamical system which minimizes over time a certain objective function, called value function. Optimal control finds many applications across different disciplines, from robotics to bioengineering to finance, to name a few. In the 1950s, R. Bellman used the concept of value function and its inherent recursivity to define an operator, known as the Bellman operator, which constitutes the foundation of dynamic programming schemes, namely value iteration, policy iteration and the linear programming approach.

1.1 Description

This project focuses on the study of the theoretical properties and a practical and theoretically sounded extension of the Bellman operator, which we call randomized mini-batch Gauss-Seidel operator. In particular, the main idea behind the new operator is to take inspiration from the techniques used in randomized mini-batch first-order methods, such as stochastic mini-batch gradient descent, to increase the flexibility and efficiency of the operator. In addition, in order to deploy the new operator within the standard dynamic programming schemes, it is important that it verifies some fundamental properties. Under such circumstances, the new operator can be used within the different standard dynamic programming methods, such as value iteration and policy iteration. In order to theoretically compare the convergence of such extended schemes with respect to their vanilla version, it is important to study the relation between the extended operator and the Bellman operator. Finally, better performance means not only faster convergence but also faster iterations in general, therefore an additional aspect to be considered is parallelization, which is taken into account in the design of the new operator. This aspect is non-negligible since parallelization not only allows to fully exploit the modern multi-cores systems but also can dramatically speed up the execution time of the program, allowing for more scalability especially in presence of a large state space.

1.2 Goals

1. Deep understanding of the Bellman operator, its fundamental properties and its use in the different dynamic programming algorithms, e.g. value iteration, policy iteration [1].
2. Formal description of the randomized mini-batch extension of the Gauss-Seidel operator taking inspiration from the techniques used in randomized mini-batch first-order methods [2]. **(*)**
3. Formal discussion of the newly introduced operator in the main dynamic programming schemes, e.g. value iteration, policy iteration.
4. Sequential implementation of the dynamic programming schemes with the newly introduced operator on some artificial benchmarks. **(**)**

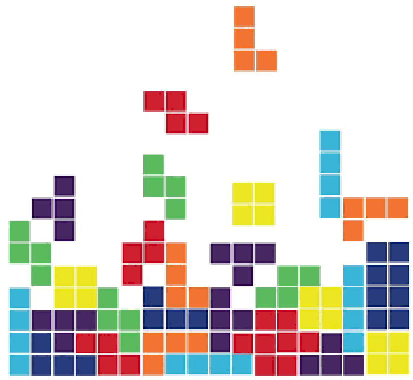

Figure 1: Graphical representation of the Tetris game.

5. Theoretical analysis of the properties of the newly introduced operator and theoretical considerations of the extended dynamic programming schemes.

6. Parallel implementation of the dynamic programming schemes with the newly introduced operator on some artificial benchmarks and comparison against the vanilla schemes and the Gauss-Seidel schemes. **(**)**

7. (optional) Parallel implementation of the extended dynamic programming schemes on a mini Tetris game and comparison against the vanilla schemes and the Gauss-Seidel schemes [3] - [4]. **(**)**

8. (optional) Modify the randomized mini-batch Gauss-Seidel operator by considering different sampling techniques, e.g. importance sampling. Conduct an empirical and theoretical analysis of the impact of such sampling techniques on the properties of the operator.

(*) Remark: Notice that the student is not required to design from scratch the new operator but rather work on an already existing but rough definition of it.
(**) Remark: Skeleton code in Python can be provided for the artificial benchmarks and for a mini Tetris game.

Publications: If the final results are promising they can potentially be turned into a publication.

Corona Disclaimer: This project can be done in person at the Automatic Control Laboratory, hybrid, or completely remotely, depending on the current ETH regulations. Most importantly, we can change between these forms whenever needed.

1.3 Qualifications

We are looking for a motivated master student with solid mathematical foundations and good programming skills (Python language is preferred). Familiarity with the main concepts of dynamic programming is not required but is certainly a plus.

1.4 How to Apply

If you are interested in this project, please send your CV and updated transcript of records to M. Gargiani (gmatilde@ethz.ch) and A. Martinelli (andremar@ethz.ch).

References

[1] D. P. Bertsekas. *Dynamic Programming and optimal control.* Athena Scientific, 1995.

[2] L. Bottou. *Large-Scale Machine Learning with Stochastic Gradient Descent* Proceedings of COMPSTAT'2010. Lechevallier Y., Saporta G. (eds) Physica-Verlag HD.

[3] V. F. Farias and B. Van Roy. *Tetris: A Study of Randomized Constraint Sampling* Probabilistic and Randomized Methods for Design Under Uncertainty. New York: SpringerVerlag, 2006: 189 – 202.

[4] J. N. Tsitsiklis and B. Van Roy. *Feature-Based Methods for Large Scale Dynamic Programming* Machine Learning. Boston: Kluwer Academic Publishers, 1996: 59 - 94.