# A Multi-Level Method for the Steady State Solution of Discrete-Time Markov Chains

Claudia Isensee and Graham Horton

Department of Simulation and Graphics,
University of Magdeburg, 39106 Magdeburg, Germany
(claudia,graham)@sim-md.de

**Abstract.** Markov chains are one of the most important kinds of models in Simulation. A fast iterative algorithm for the steady state solution of continuous-time Markov chains (CTMCs) was introduced by Horton and Leutenegger [HL94]. The so-called multi-level algorithm utilizes ideas from algebraic multigrid to provide an efficient alternative to the currently used Gauss-Seidel and successive overrelaxation (SOR) methods.
This paper examines the applicability of the algorithm and its ideas to discrete-time Markov chains (DTMC). The steady state solution is determined by successively coarsening an initial problem and solving it in parallel on various levels of detail. Since CTMC and DTMC are closely related and stiff problems pose the same difficulties for the iterative solution methods Power (DTMC) and SOR (CTMC) the effect of the multi grid approach is equally good.
An adapted algorithm is presented and tested on different chains, that were derived from the ones used in testing the original Multi-Level Algorithm. The experiments show, that the runtime can be greatly reduced when using the Power method and a slightly modified aggregation scheme. When using the Gauss-Seidel method as smoothing algorithm the performance is even better.

## 1 Introduction

The goal of our current research is to develop an algorithm for the steady state solution of discrete-time Markov chains, that uses the same idea of successively coarsening an initial problem and solving it in parallel on various levels of detail. We hope to utilize this algorithm later to improve our new proxel- & phase-based simulation method, whose underlying stochastic process is a DTMC.

The paper will not contain an elaborate description of the original multi-level algorithm, but concentrate on its adaption to DTMCs. The standard Markov chain solution methods Power and Gauss-Seidel (*GS*) are taken as smoothing methods. In the experiments the adapted algorithm will be tested against the Power and SOR methods.

## 2 Motivation

Any discrete stochastic simulation model with a bounded state space can be transformed into a Markov chain. This can be done using continuous-time Markov chains (CTMCs).

An efficient algorithm for the steady-state solution of CTMCs has been developed [HL94], which outperforms the common iterative methods SOR and Gauss-Seidel.

A model can also be discretized and turned into a DTMC, which is done for example by the Proxel-based simulation algorithm developed by Horton [Hor02] and improved and elaborated since. When turning a model containing general distribution functions into a DTMC, these distributions have to be approximated. In the proxel method this is usually done on-the-fly using the instantaneous rate function. When using discrete phase-type distributions [IH05], the approximation has to be precomputed. In both cases the resulting expanded model state space has a fairly regular grid-like structure.

The resulting Markov chain could also be solved analytically. When the original state space of the model is large, or the used time step is really small, the resulting expanded state space can contain millions of Markov chain states. This behavior is called state space explosion. Therefore an efficient algorithm is needed to solve the model. The existing Power method works, but might need a large number of iterations to reach a sufficiently accurate solution.

The multi-level algorithm described in [IH04] works very efficient on some CTMCs with regular structures. Therefore we hope, that an adaption of the algorithm will perform similarly well on DTMCs, specifically on regular structures, as can occur in the expanded state space of discrete stochastic models.

CTMCs and DTMCs are closely related, and therefore we assume that many of the features of the CTMC algorithm have their equivalent for the discrete type. The problem of solving Markov chains in reasonable time with iterative methods is much the same for the discrete as for the continuous case. The standard method for finding the steady state solution of a DTMC is the Power method. Its convergence does not only depend on the size of the Markov chain, but also on its parameters. The number of iterations needed can become unreasonably high for stiff problems, which include the NCD Markov chains. This is in essence the same behavior that the Gauss-Seidel Algorithm exhibits when applied to CTMCs, and which could be successfully counteracted by the CTMC multi-level algorithm.

## 3 The Multi-Level Algorithm

This section will describe the multi-level approach in general and some specifics when applying it to DTMCs and the Power method. The original steady state solution problems of DTMC can be specified as finding the vector $\pi$ that satisfies Equation 1. Where $\pi$ is a probability vector and $P = [p_{ij}]_{n \times n}$ is the transition probability matrix of the DTMC. Since the usual solution method for DTMCs is Power, this was also the first choice for the smoothing algorithm in the DTMC multi-level approach. Without changing the solution vector, the problem definition can be transformed as seen in Equations 2 and 3 ($I$ identity matrix). The newly formulated problem 3 is similar to the steady state problem of a CTMC: $Q\pi = \mathbf{0}$, and therefore Gauss-Seidel can also be used as smoothing method. Since Gauss-Seidel most of the time converges faster than the Power method, we hope to achieve better performance when using it as smoothing method.

$$P\pi = \pi \tag{1}$$

$$P\pi - I\pi = \pi - I\pi \tag{2}$$
$$(P - I)\pi = \mathbf{0} \tag{3}$$

The general strategy of the multi-level algorithm is to create a series of ever smaller Markov chains from the original one. By solving all of these simultaneously within each iteration, the algorithm is much faster than the conventional iterative ones. This works, because the multi-level algorithm combines nodes that influence each other strongly and whose values can be smoothed by a few low-level iterations (Gauss-Seidel or Power). They are mapped to a common node on the so-called coarser level, thereby removing the disparity between strong and weak connections. Probability can flow freely between the nodes on the coarse level, therefore an equilibrium can be reached in fewer iterations. By applying this process recursively, the chain is coarsened, and any disparities between transition rates are compensated.

However, it is of vital importance how the nodes are aggregated, which ones are combined, or even split between their neighbors. The process of calculating that mapping is called aggregation. It is an important part of the algorithm, and will be described in Section 3.1. A detailed description of the multi-level Algorithm for CTMCs can be found in [IH04]. The following section will describe the equivalent for a DTMC.

The *restriction* maps a Markov chain from the fine to the coarse level. A child node can have multiple parents. Since the Power method is susceptible to the absolute size of the transition probabilities, a simple restriction does not decrease the number of iterations needed to obtain an accurate solution. However, when edges with large transition probabilities are eliminated in the coarse chain, the probabilities of the other links can be scaled up. When the result is a stochastic matrix, the ratio of two neighboring nodes and the original solution of the chain are preserved in this step. This step increases the runtime of the method significantly, when applying it to NCD Markov chains.

The *prolongation* maps a Markov chain from the coarse to the fine level. A parent node can have multiple children, and again the probability has to be transferred from the coarse to the fine level completely, which results in column sums of one. Both prolongation and restriction factors are calculated in exactly the same way for DTMCs as for CTMCs, the detailed description can be found in [IH04].

To iterate through the Markov chain on each level the Power method or Gauss-Seidel are employed, which compute the new probability of a node as described in [Ste94] (pages 121-128). Some smoothing steps are performed on every level before restriction and after prolongation to smooth the error. On the lowest level, with the smallest chain, the smoothing method is used to solve the problem exactly. The normalization step that was needed when using Gauss-Seidel is not needed wen using Power, because the Power method preserves the probability vector property of $\pi$.

### 3.1 Aggregation Strategies

The aggregation method is the heuristic part of the multi-level algorithm, and has a major influence on its performance. The decisions which nodes are going to be aggregated and which might be split to their neighbors are made according to the aggregation strategy. The main goal of the aggregation strategy is to aggregate the nodes in a way, that nodes whose relative errors can be smoothed out in a few iterations are combined and

mapped as one to the next level, and that nodes who are equally strongly connected to more than one neighbor might be split to them.

The aggregation strategies that are implemented in the original multi-level Algorithm will not be described here in detail, they can be found in [LH95], [Lab97], [IH04]. This section will briefly describe the ideas behind the different strategies and examine their applicability to DTMCs and the Power method.

In [LH95] the Multi-Level (*ML*) algorithm is applied to NCD Markov chains, and the chosen strategy was to aggregate by strong connections. It was also defined that a coarse node should have at most three children, so that the number of nodes does not decrease too fast, otherwise the advantage of having intermediate levels would be lost. The strategy of aggregation by strong connections works very well for Markov chains with heterogeneous transition rates, such as NCD Markov chains. The performance of the algorithm proved to be much better than that of conventional methods, even a number of iterations that is independent of problem parameters could be achieved, which is an optimal result. Markov chains with almost identical transition rates on the other hand only showed a moderate improvement in computation effort when solved with the multi-level algorithm. This strategy can also be used for DTMCs, since it does not account for the order in which the nodes are processed, and strong connections can also be defined using transition probabilities.

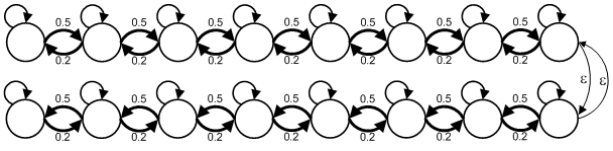
In [Lab97], an extension of the algorithm was proposed (*Ext*). Nodes which are connected equally strongly with more than one neighbor may also be split among these. This necessarily results in a change of the restriction and prolongation formulas. The strategy to determine whether a node is connected equally strong to some neighbors is solely based on thresholds and local transition rates. Specific patterns of the Markov chain are not utilized. For one-dimensional Markov chains with identical transition rates (Figure 2 right), the performance could be improved drastically. A strategy that represents the optimal solution is found by the algorithm. Every second node is split equally to its neighbors, and the problem of choosing one aggregation partner is resolved. This strategy can also be used for DTMCs, since thresholds and strong connections can also be defined.

A new aggregation strategy was described in [IH04] (*New*). The general idea of the method is to first decompose the chain into groups of strongly connected nodes, and then try to find the optimum strategy for the coarsening of the individual groups. Thereby the advantages of the *ML* an *Ext* strategies are combined, disparate transition rates, as well as nearly identical transition rates are treated appropriately. The first step of this strategy is to separate the edges of the Markov chain into sets which are deemed to be relevant or irrelevant for the further aggregation. Throughout the whole process of aggregation only these edges are considered, but they are not weighted against each other. The definition of relevance however includes the order the nodes are processed in the Gauss-Seidel method. Since the processing order is of no concern in the Power method, there this definition does not make sense. It can only be applied to the Power method when taking out all references to the order of the nodes.

## 4 Experiments

In this section the modified multi-level algorithm using Power and Gauss-Seidel as smoothing methods will be tested against the Power and the SOR method.

Experiments with the simple one-dimensional NCD chains showed that the number of needed iterations for a solution could be made independent of the size of the weak link, which is the intention of the original multi-level algorithm. (Figure 4)




| $\epsilon$ | *ML* | Power |
|---|---|---|
| 1 | 10 | 186 |
| 0.1 | 9 | 346 |
| 0.01 | 9 | 1,827 |
| 0.001 | 9 | 13,549 |
| 0.0001 | 9 | 96,478 |
| 0.00001 | 9 | 581,326 |
| 0.000001 | 9 | 1,983,983 |

**Fig. 1.** Left: Simple one-dimensional NCD Markov chain, Right: Number of needed iterations for different sized $\epsilon$ and different methods

Further experiments were performed for one-dimensional Markov chains with identical transition probabilities (see Figure 2 left) and for regular grids with large transition probabilities ($p_{ij} = 0.4$) running in horizontal direction and relatively small transition probabilities ($p_{ij} = 1e{-}6$) running in vertical direction (see Figure 2 right). These grid-like Markov chains can also be classified as *nearly-completely decomposable* (NCD) Markov chains. Experiments with a real expanded reachability graph were not done, due to the lack of an adequate generator, and will have to be done in the future. Nevertheless do the tested chains represent some structures, that can occur, and that do not work well with the original iterative methods.

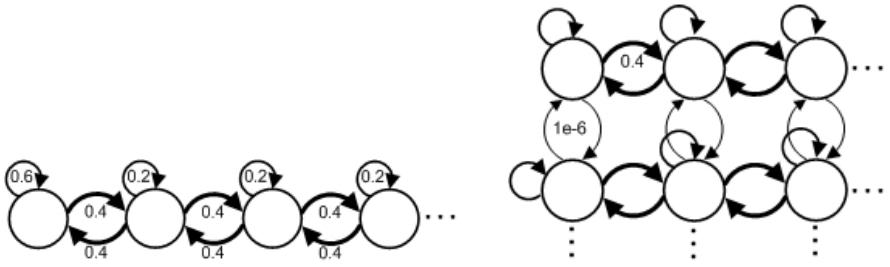



**Fig. 2.** Left: One-dimensional Markov chain with identical transition probabilities, Right: Grid-like NCD Markov chain

The different conventional algorithms used were the Power and SOR methods. The Gauss-Seidel (*GS*) and Power as smoothing methods were combined with all three described aggregation strategies (*ML*, *Ext*, *New*). This lead to six combinations of multi-level algorithms, which are identified by their aggregation strategy and their smoothing method in the diagrams. *New-Power* for example means the *New* aggregation strategy was combined with the Power smoothing method.

The comparison criteria were the number of floating point operations performed, which are equivalent to the runtime of the algorithm, and the number of iterations needed, were a multi-level iteration is defined by traversing the chain levels once and includes several *GS* or Power iterations. The algorithms were terminated, when the initial error, defined by the maximum norm of the defect, was decreased by a factor of $1e-6$. This ensured a comparable performance for all algorithms.

Figure 3 left shows the development of the number of floating point operations with growing chain length for one-dimensional Markov chains with identical transition probabilities. The combinations *Ext-GS* and *New-GS* are by far the cheapest and SOR and Power the most expensive. The performance of the multi-level algorithm using Power and *ML-GS* lie in the middle.

Figure 3 right shows the development of the number of floating point operations with growing grid size in the described grid-like Markov chains. The combination *New-GS* is the cheapest and the Power method the most expensive. When comparing the two smoothing methods using the same aggregation strategy, the combination using *GS* always outperforms the Power combination.

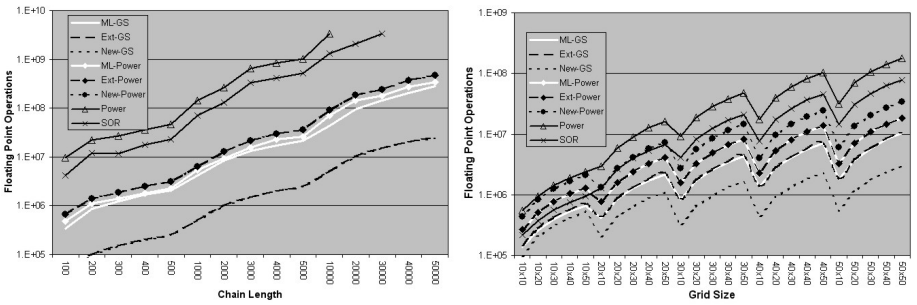



**Fig. 3.** Number of floating point operations for different sized Markov chains, logarithmic scaling, Right: One-dimensional Markov chains, Left: Grid-like Markov chains

One of the goals of the multi-level algorithm was to make the number of needed solution iterations independent of some problem parameters. This can be achieved with the original multi-level algorithm. When using the Power smoothing method, the improvement is not as drastic, but when using *GS* the same positive results can be achieved for DTMCs. As Figure 4 left shows, the number of needed iterations is constant for the *Ext-GS* and *New-GS* combinations but not for the combinations using the Power method for one-dimensional chains.

A simular behavior can be seen for the grid-like Markov chains (see Figure 4 right). Only the *New-GS* combination needs a constant number of iterations to reach an accurate solution. All other multi-level algorithms are dependent on the grid size.

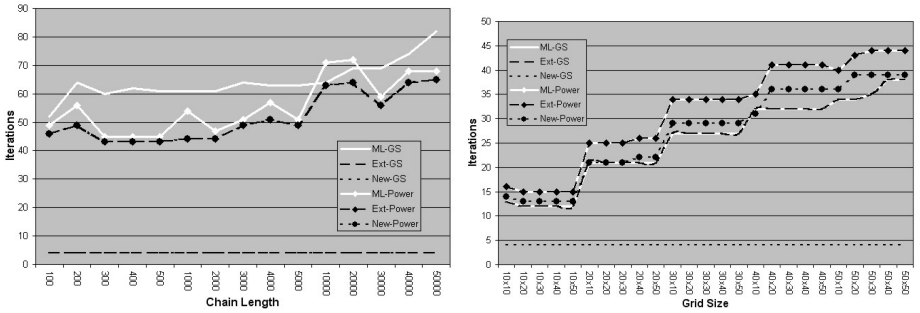



**Fig. 4.** Number of iterations for different sized Markov chains, Right: One-dimensional Markov chains, Left: Grid-like Markov chains

By adapting the multi-level algorithm for DTMCs the runtime needed for finding the steady-state solution could be decreased significantly compared to the Power method. A constant number of solution iterations could only be achieved on the same classes as in the original algorithm, when using the Gauss-Seidel method for smoothing. The aggregation strategies combined with the *GS* smoothing method outperformed the ones using the Power method in most tested cases. When using Power, the number of iterations needed to achieve an accurate solution for the adapted algorithm is independent of the model parametrization, but not of the chain length or the grid size, as the original algorithm is.

One reason for the higher runtime complexity of the adapted algorithm using Power is that the Power method in many cases converges slower than Gauss-Seidel, because one Power iteration does not reduce the error as well as one Gauss-Seidel step. The original multi-level algorithm, especially the *New* aggregation strategy, was developed by exploiting the Gauss-Seidel property of being susceptible to the ordering of the nodes. This included sorting the Markov chains at certain points during the computations. The Power method is not sensitive to the node order, and consequently this property could not be exploited.

## 5 Conclusion and Outlook

In this paper a multi-level algorithm for the steady-state solution of CTMCs was adapted for DTMCs. Some minor changes had to be made to the original algorithm and to the aggregation strategies. The adapted algorithm using Power and *GS* as smoothing methods was then tested against the original algorithm and the SOR and Power iterative methods.

The presented multi-level algorithm needs significantly less computation time than the Power method. The best results were achieved, when using Gauss-Seidel as smoothing method. This is mostly due to the better convergence behavior of the Gauss-Seidel method and its sensitivity to node ordering, which was exploited in the original multi-level algorithm. Therefore when the steady state solution of a DTMC is to be computed, one should choose to use the presented multi-level algorithm with Gauss-Seidel as smoothing method.

The experiments presented in this paper include only artificially generated chains with specific structures, as they might occur in extended reachability graphs of discrete stochastic models. Testing the algorithms on real models has yet to be done.

## References


[HL94] Graham Horton and Scott T. Leutenegger. *A Multi-Level Solution Algorithm for Steady-State Markov Chains*. In: ACM SIGMETRICS, 1994.

[Hor02] Graham Horton. *A New Paradigm for the Numerical Simulation of Stochastic Petri Nets with General Firing Times*. In: European Simulation Symposium, Dresden, Germany, 2002.

[IH04] Claudia Isensee and Graham Horton. *A Multi-Level Algorithm for the Steady State Solution of Markov Chains*. In: Simulation and Visualization 2004, Magdeburg, Germany, 2004.

[IH05] Claudia Isensee and Graham Horton. *Combining Proxels and Discrete Phases*. In: International Conference on Modeling, Simulation and Applied Optimization, Sharjah, U.A.E., 2005.

[Lab97] Ulf Labsik. *Algorithmische Erweiterung des Multi-Level-Verfahrens zum Lösen von Markov-Ketten*. Thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg, 1997.

[LH95] Scott T. Leutenegger and Graham Horton. *On the Utility of the Multi-Level Algorithm for the Solution of Nearly Completely Decomposable Markov Chains*. In: Second Int Workshop on the Numerical Solution of Markov Chains, 1995.

[Ste94] W. J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, Princeton, NJ, 1994.