New ways of solving large Markov chains

Queueing Systems (2022) 100:217–219
https://doi.org/10.1007/s11134-022-09821-3

New ways of solving large Markov chains

Nelly Litvak[1,2]

Received: 6 February 2022 / Accepted: 28 February 2022 / Published online: 28 April 2022

1 Introduction This article is about solving large Markov chains in a new way. This traditional topic, that is of direct relevance to queueing systems, has been receiving a lot of attention since Google PageRank was introduced in 1998 to rank web pages [4]. PageRank is the stationary distribution of a random walk with restart on the graph of web pages connected by hyperlinks. Most common approach for computing a stationary distribution of a Markov chain, is *power iterations* (PI), when the initial probability vector is iteratively multiplied by the transition matrix till convergence. Here I will explain a new *Red-Light-Green-Light* (RLGL) algorithm that we developed with Konstantin Avrachenkov and Patrick Brown [2]. RLGL is fast, and generalizes many methods, including PI, and the state-of-the-art Gauss-Southwell method for PageRank.

2 Problem statement Consider an ergodic Markov chain with a finite state space $\{1, 2, \dots, N\}$ and transition probability matrix $P = (p_{ij})$, where p_{ij} is the transition probability from state i to state j. Let $\pi^* = (\pi_1^*, \pi_2^*, \dots, \pi_N^*)$ be its stationary distribution. As a small example, in Figure 1(a), Anna, Boris and Cecile together form a Markov chain. The stationary distribution is $\pi^* = (5/12, 4/12, 3/12)$. Figure 1(b) shows convergence of PI, which I will interpret as cash transactions: the probability of each state is its (positive) cash; at each iteration all states transfer all their cash proportionally to the transition probabilities; in this example, there are five transactions per iteration. The number of transactions till convergence is a relevant performance measure because one transaction roughly corresponds to one computer operation.

An example of RLGL is worked out in Fig. 2a.[1] Initially, all states have no cash. At step $t = 0$, all states borrow one unit of cash and distribute it to the other states, proportionally to the transition probabilities. Now, each state has debt -1, and income

[1] If the table is not clear, in this video I fill out a similar table: https://player.vimeo.com/external/519732547.hd.mp4?s=45b4736109005e144b0f792396f2fa7dba4a25bf&profile_id=174

✉ Nelly Litvak
n.litvak@utwente.nl

1 Faculty of Electrical Engineering, Mathematics, and Computer Science, University of Twente, Enschede, The Netherlands

2 Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands

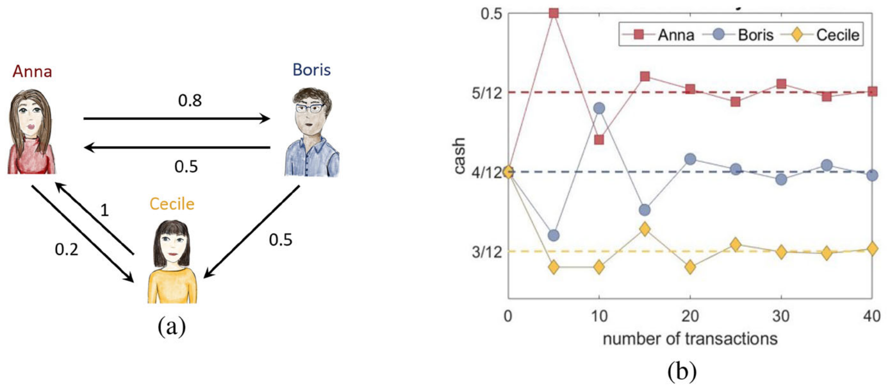


Fig. 1 **a** Three-state Markov chain. **b** Convergence of power iterations (PI). Horizontal axis: the number of transactions (operations). Vertical axis: the probabilities (cash) of each state. Drawings by Natalia Litvak

received from the other states. Hence, some states have positive cash, some have negative cash, and the total cash in the system is zero. Next, at each step $t > 0$, a subset of states receive 'green light' and transfer all their cash proportionally to the transition probabilities. The total cash in the system remains zero at all times. In Fig. 2a, Boris receives green light at $t = 1$ and Cecile at $t = 2$. This is indicated by the green circles. The algorithm returns the estimation $\hat{\pi}_{t,i}$ of π_i^*, that is the cash transferred by i, divided by the total cash transferred by all states before time t. Note that $\hat{\pi}_{t,i}$ is not a probability, it can be negative or greater than one.

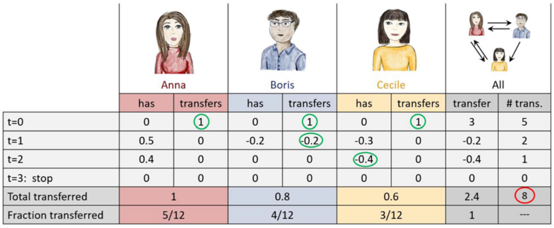
Why does the RLGL algorithm converge to the correct π^*? Denote by $C_{i,t}$ the cash of state i at the beginning of step t, define $C_t = (C_{1,t}, C_{2,t}, \ldots, C_{N,t})$, and let $\mathbf{1}$ be the column vector of ones. It turns out that

$$\hat{\pi}_t = \pi^* - \frac{1}{\text{total cash transferred by all states before step } t} C_t \sum_{k=0}^{\infty} (P^k - \mathbf{1}\pi^*). \quad (1)$$

The error term has factor C_t, so reducing cash means getting closer to the solution. In Fig. 2a, after only 8 transactions, no one had cash left, and the algorithm returned the exact answer! Compare this to the very rough approximation after 10 transactions of PI in Fig. 1b. This also explains why the negative cash is so important. In fact, the OPIC algorithm suggested in [1] works exactly as RLGL but uses only positive cash, with fixed total amount. Then, the error term of OPIC is as in (1), but C_t will not reduce to zero, and thus the convergence occurs only due to the total transferred cash going to infinity. This results in $||\hat{\pi}_t - \pi^*||_1$ reducing as $O(1/t)$ rather than exponentially.

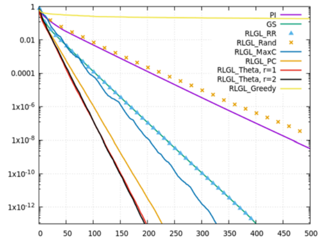
Figure 2b (from [2]) shows an example of the excellent performance of RLGL in a larger real-life Markov chain. The results are promising, and there are many open problems, which we will discuss in the next section.

3 Discussion The first obvious open problem is how to assign green lights. Notice that in Fig. 2a, if Anna transferred cash first, convergence would take longer. In [2] we computed the optimal policy for a three-state Markov chain, using dynamic pro-



	Anna		Boris		Cecile		All	
	has	transfers	has	transfers	has	transfers	transfer	# trans.
t=0	0	1	0	1	0	1	3	5
t=1	0.5	0	-0.2	-0.2	-0.3	0	-0.2	2
t=2	0.4	0	0	0	-0.4	0	-0.4	1
t=3: stop	0	0	0	0	0	0	0	0
Total transferred	1		0.8		0.6		2.4	8
Fraction transferred	5/12		4/12		3/12		1	---

(a)



(b)

Fig. 2 **a** Example of the RLGL algorithm. **b** Results for a random walk on the strongly connected component of the web crawl from www.harvard.edu in 2003, of 500 web pages (Suite Sparse Matrix Collection (http://sparse.tamu.edu/)). On the horizontal axis is the number of transactions/operations. On the vertical axis is the L_1 distance (in the log-linear scale) to the correct stationary distribution. GS stands for the Gauss-Seidel algorithm. Different versions of the RLGL algorithm correspond to the different choices of green light (see [2]). RLGL-Theta gives green light to a fraction of states with maximal absolute cash

gramming. The performance of the optimal policy is impressive, but its form, even for three states, depends on the cash values in a very intricate way, which we could not describe analytically.

Another problem is, how to derive convergence rates. In [2] we could express $||C_t||_1$ through the total variation distance between two Markov chains and used coupling. However, these two Markov chains are inhomogeneous and dependent through the sequence of green lights, so we could prove exponential convergence only in special cases, e.g., when green lights are cyclic. The problem might be easier if a Markov chain has more structure, as we often see in queueing theory. Philippe Robert and I analyzed OPIC in this setting [5]. Also, for some sparse Markov chains, maybe RLGL can be related to some kind of branching process, as it was done for random walks, e.g., in [3].

Finally, ideally, RLGL could work in a distributed way, when each state decides when to transfer its cash. Then, maybe it will be useful to treat each state as a queue with a positive or a negative workload. RLGL, hopefully, will be used to solve large queueing systems, but maybe queueing systems can be used to solve RLGL.

References

1. Abiteboul, S., Preda, M., Cobena, G.: Adaptive on-line page importance computation. In: Proceedings of the 12th International Conference on World Wide Web, pp. 280–290. ACM (2003)
2. Avrachenkov, K., Brown, P., Litvak, N.: Red light green light method for solving large Markov chains. arXiv preprint arXiv:2008.02710, (2020)
3. Bordenave, C., Caputo, P., Salez, J.: Random walk on sparse random digraphs. Probab. Theory Related Fields **170**(3), 933–960 (2018)
4. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Comput. Netw. ISDN Syst. **30**(1–7), 107–117 (1998)
5. Litvak, N., Robert, P.: A scaling analysis of a cat and mouse Markov chain. Ann. Appl. Probab. **22**(2), 792–826 (2012)