MSc Thesis Applied Mathematics

Red-Light-Green-Light algorithm with MaxC heuristic for solving Markov chains

Iris Ottens

Supervised by:
Prof.dr. N.V. Litvak(UT)
Prof.dr K. Avrachenkov(INRIA)
Dr. M. Schlottbom(UT)
Dr.ir. W.R.W. Scheinhardt(UT)

February 16, 2022

Mathematics of Operations Research
Department of Applied Mathematics
Faculty of Electrical Engineering,
Mathematics and Computer Science

UNIVERSITY OF TWENTE.

Abstract

Markov chains are versatile mathematical models which can represent stochastic processes from many different fields. A common task when using Markov chains is the computation of the stationary distribution. A new and fast algorithm that can be used to calculate this stationary distribution is the Red-Light-Green-Light(RLGL) algorithm. To give a brief summary of the algorithm: first each state receives a certain amount of cash, which can be positive or negative, and at each iteration a subset of all states receive 'green light' after which these states distribute all their cash proportional to the transition probabilities of the Markov chain. The stationary distribution can then be found to be proportional to the total cash distributed by each state. Green light can be given in many different ways and choosing the correct way can lead to faster convergence. Finding the optimal green light sequence can, however, be very challenging. It is therefore better to look at heuristics for giving green light. The MaxC heuristic and the MaxCMinC heuristics are such heuristics. The MaxC heuristic gives green light to the states that have the absolute maximum amount of cash and the MaxCMinC heuristic gives green light to both the states with the maximal positive and those with the maximum negative amount of cash. For both of these heuristics we have proven exponential convergence for certain types of Markov chains and estimated their convergence rate. It is shown that, in practice, the heuristics often outperform their theoretical bounds, as well as methods of how these bounds can be made more precise.

Contents

1 Introduction

Markov chains are versatile mathematical models which can represent stochastic processes from many fields like linguistics, logistics, and biology. Another well known application is Google PageRank [1], which can be seen as a random walk on a graph with a restart property. The application of the Markov chain is often found in finding its stationary distribution, which can be a challenge to find when a Markov chain has an extremely large number of states.

A new and fast algorithm for those cases is the Red-Light-Green-Light(RLGL) algorithm, which was first introduced in [2]. This algorithm represents the states of the Markov chain like nodes in a graph that each have a certain amount of cash, which can be positive or negative. At each iteration the algorithm gives green light to a subset of these nodes, which then proceed to distribute their cash proportional to the transition probabilities. The stationary distribution can then be estimated based on the total amount of cash each node has transferred.

One advantage of the RLGL algorithm is that it is very versatile. Choosing the right states to give green light to at each step can lead to even faster convergence. Finding the most optimal green light sequence can however be quite challenging and beat the purpose of the final algorithm being faster. We therefore instead look at heuristics for giving green light. Many of these heuristics have been proposed in [2], but not yet rigorously analysed. This report mainly focuses on doing this for one of these heuristics, the so called MaxC heuristic. This heuristic gives green light to the states that have the absolute maximum amount of cash.

The analysis of the MaxC heuristic gave the idea for a new heuristic, the MaxCMinC heuristic, that gives green light to both the state that has the maximum positive and the one that has the maximum negative amount of cash. This heuristic is analysed as a natural extension of the MaxC heuristic.

1.1 Goal of this research

The goal of this report is to investigate the MaxC heuristic as well as generalize its results to also apply to the MaxCMinC heuristic. This for the purpose of obtaining conditions under which the heuristics converge and deriving their convergence rate. To achieve this both analytical derivations and numerical experiments are used as well as making comparisons to other heuristics and methods.

1.2 Related works

This report is mostly based on [2] and the algorithm described therein. The relevant results from that paper are summarized in chapter 2. As the algorithm is still quite new, not much research is available about it. The nature of the algorithm, however, allows results of other algorithms to be carried over to some extend. This is as, as shown in [2], several existing methods are incorporated into the RLGL algorithm.

Firstly there is the power iteration method, a simple and widely applicable method. This case is the same as giving green light to all states at every iteration, which looses the flexibility that makes the RLGL algorithm capable of performing better. This makes the power iteration method a nice baseline to compare with, as is often done in other research as well [3][4].

Another family of algorithms that can be incorporated into the RLGL algorithm is the family of Gauss-Southwell methods that use gradient decent for PageRank [4][5][6][3]. The coordinate or direction, which can correspond to the different states of a Markov chain, along with the algorithm is updated can be chosen in different ways, which draws a parallel to how different heuristics can be used to give green light with the RLGL algorithm. Usually the greedy coordinate descent is used, which, when translated to the RLGL algorithm, means giving green light to the state with the maximum amount of cash. It is easy to see the similarities with the MaxC heuristic that this report discusses. In fact, as the Gauss-Southwell algorithm only interacts with the states that have positive cash, the MaxC heuristic is a natural extension to greedy coordinate descend that takes advantage of the fact that states can also have negative cash. Therefore [7][8][3][4][5], which show the advantages of greedy coordinate descent are also a validation for picking the MaxC heuristic as a promising heuristic. Other works [9][4][5] explore alternatives or improvements of greedy coordinate descent and could provide ideas for new heuristics to be analysed in the future.

Lastly there is the OPIC algorithm [10] that the RLGL algorithm was based on as well as other "cash based" algorithms [6]. The great similarities these algorithms have with the RLGL algorithm make papers that look at different interpretations of these types of algorithms [11][12] and give results regarding their convergence [12] useful. This is as they provide angles that can be taken to analyse and possibly prove the convergence of the RLGL algorithm when certain heuristics are used.

1.3 Structure of the report

The next chapter, chapter 2, provides definitions and background knowledge necessary for the report. Any explanations not given in this chapter are given when they are needed. This chapter starts, in section 2.1, with highlighting the formal description of the RLGL algorithm as given in [2]. This is followed by, in section 2.2, a description of all heuristics for giving green light discussed in this report. These are the MaxC heuristic, MaxCMinC heuristic, and the Theta heuristic and they are discussed in that order.

Chapter 3 completely analyses the MaxC heuristic for a three state ergodic Markov chain without self loops. Section 3.1 proves exponential convergence for this case and the next section 3.2 analyses the rate of convergence. Lastly, in section 3.3 this chapter makes a theoretical comparison of the MaxC heuristic and the power iteration method.

Chapter 4 proves exponential convergence for and estimates the convergence rate of the three heuristics for when the Markov chains is a random walk on a complete graph. The first section 4.1 does this for all three heuristics, the MaxC heuristic, the MaxCMinC heuristic, and the Theta heuristic, while making no assumptions about the graph or transition probabilities other than them being non-zero. The second section 4.2 provides a better estimate for the convergence rate of the MaxC heuristic and the MaxCMinC heuristic when the so called Mean-Field Stochastic Block Model is used as model for the graph.

Chapter 5 contains the numerical results of several experiments. The models used include several three state ergodic Markov chains without self loops, a complete graph where all transition probabilities are picked according to a Uniform distribution, and the Mean-Field Stochastic Block Model with various number of blocks. The provided graphs demonstrate the convergence of the different heuristics and compare them to each other, to the power iteration method, and to the theoretical bounds found in the previous chapters.

The report in concluded in chapter 6 which summarizes the results in a conclusion and provides ideas and recommendations for future research.

2 RLGL algorithm

The RLGL (Red-Light-Green-Light) algorithm is a versatile algorithm for computing the stationary distribution of large Markov chains first introduced in [2]. In this algorithm, each state can be seen as a node of a graph that has a certain amount of 'cash' that can be positive or negative. When a state receives 'green light' from the algorithm it distributes its cash to other states, regardless of whether this cash is positive or negative, proportional to the transition probabilities of the Markov chain. The main difference from OPIC [10], which is based on the redistribution of positive cash, is the addition that cash can also be negative. This addition of negative cash makes it possible for the sum of all cash present in the system to equal to zero at all times, whereas the absolute total cash present in the system, $\|C(t)\|_1$, will decrease. This decrease to the absolute total amount of cash to zero defines the convergence of the RLGL algorithm, as will be further discussed in the next section. Another feature of the RLGL algorithm is its flexibility caused by the many ways in which green light can be given at each step of the algorithm. This flexibility allows many existing methods to be incorporated under its umbrella, OPIC being one of those as well the power iteration method. It is this same flexibility that gives the RLGL algorithm the potential to outperform many of these other methods.

2.1 Formal description of the RLGL algorithm and notation used

The RLGL algorithm is a method for computing the stationary distribution of a Markov chain. To ensure the existence of a unique stationary distribution, among other things, we consider ergodic Markov chains. Ergodic means that the Markov chain is aperiodic and positive recurrent. Take such a Markov chain with a finite state space of n states, $\{1, 2, ..., n\}$, and transition probability matrix P with entries p_{ij}. Here p_{ij} is the probability of going to state j when the Markov chain is currently in state i. We assume the Markov chain is ergodic, so each state has a period of 1 and a probability of 1 that the Markov chain will return to the state after a finite number of steps. This Markov chain then has a stationary distribution π^*, such that $\pi^* = \pi^* P$ and $\sum_{i=1}^{n} \pi^* = 1$, that should be the output of the RLGL algorithm.

The algorithm is given in pseudo code in Algorithm 1 and its various parts are explained below in a similar manner as in [2].

Algorithm 1 RLGL Algorithm[2]

Input: n, P, ε, rule for choosing G_t, $M(0)$
Output: $\hat{\pi}(t)$

1: $t = 0, C(0) = \mathbf{0}, H(0) = \mathbf{0}$;
2: $H(t+1) = H(t) + M(t)$;
3: $C(t+1) = C(t) - M(t)(I - P)$;
4: $t \leftarrow t + 1$;
5: $\hat{\pi}(t) = \frac{1}{H(t)\mathbf{1}_n^T} H(t)$;
6: $M(t) = C(t) I(G_t)$;
7: Iterate 2 - 6 until $\|\hat{\pi}(t) - \hat{\pi}(t-1)\|_1 < \varepsilon$ for some $t \geq 2$

The algorithm goes through multiple iterations that are numbered $t = 0, 1, ...$ At any $t \geq 0$, $C(t)$ denotes the vector with entries $C_i(t) \in \mathbb{R}$, $i \in \{1, ..., n\}$. Here $C_i(t)$ is the cash at state i at the beginning of iteration t that can both be positive and negative. To initialize we set $C(0) = \mathbf{0}_n$, where $\mathbf{0}_n$ is a row vector of zeros of length n.

At any $t \geq 1$ a subset of all states, $G_t \subseteq \{1, ..., n\}$, receives green light to move their cash proportional to the transition probabilities. The cash moved by state i at iteration t is denoted by $M_i(t)$, which are entries of the row vector $M(t)$. Formally this means that $M_i(t) = C_i(t)$ if $i \in G_t$ and $M_i(t) = 0$ otherwise. For the initial step this is slightly different. For this we give a vector $M(0)$ containing the cash that each state 'borrows' to distribute in the first iteration. As all $C_i(0) = 0$, state i will be left with $-M_i(0)$ after giving away its cash, which will be (partially) compensated with the positive cash it gets from its neighbors.

In most cases $M(0)$ is simply chosen to be equal to a uniform distribution over all states, $M(0) = \frac{1}{n}\mathbf{1}_n$ where $\mathbf{1}_n$ is a row vector of ones of length n. Note that, regardless of how $M(0)$ is chosen, the total amount of cash present in the system remains zero for the full duration of the algorithm, $\sum_{i=1}^{n} C_i(t) = 0$ for all $t \geq 0$.

The total amount of cash distributed by state i before iteration t is stored as $H_i(t)$ as an entry of the row vector $H(t)$ and called its history. Formally this means $H_i(0) = 0$ and $H_i(t) = \sum_{k=0}^{t-1} M_i(k)$ for $t \geq 0$ and $i \in \{1, ..., n\}$.

The estimator $\hat{\pi}(t)$ of π^* provided by the algorithm is based on the history:

$$\hat{\pi}(t) = \frac{1}{H(t)\mathbf{1}_n^T} H(t). \tag{1}$$

It should be noted that $H(t)\mathbf{1}_n^T$ can be zero which would make equation (1) ill defined. When this happens we restart the algorithm with a possible change in input.

The algorithm stops upon convergence of $\hat{\pi}(t)$: $\|\hat{\pi}(t) - \hat{\pi}(t-1)\|_1 < \varepsilon$ for some previously defined $\varepsilon > 0$.

Reference [2] shows that $\hat{\pi}(t)$ converges to π^* and that they relate as follows:

$$\hat{\pi}(t) = \pi^* + \frac{1}{H(t)\mathbf{1}_n^T} \left(C(0) - C(t)\right) \sum_{k=0}^{\infty} \left(P^k - \mathbf{1}_n^T \pi^*\right). \tag{2}$$

By assuming that $\inf_{t>0} |H(t)\mathbf{1}_n^T| > 0$ and having chosen $C(0) = \mathbf{0}$, as in [2], we know that $\hat{\pi}(t)$ converges to π^* at the same rate as $\|C(0) - C(t)\|_1 = \|C(t)\|_1$ converges to 0. Therefore, if this report talks about the convergence of the algorithm it will do so in terms of convergence of $\|C(t)\|_1$ to 0. One should also note here that, with the exception of the first iteration, $\|C(t)\|_1$ can not increase, but is not guaranteed to strictly decrease either. Whether $\|C(t)\|_1$ converges to 0 and how quickly it does so mostly depends on the choice of G_t. The way in which green light is given is therefore both what makes the algorithm versatile and what needs to be carefully chosen to give quick convergence.

2.2 Heuristics for giving green light

As mentioned before, one of the factors that makes the RLGL algorithm versatile is the many different ways in which green light can be given. As shown in [2] it is in most cases, however, not feasible to find the optimal strategy for giving green light considering its dependence on the cash distribution and the exponential number of options for each possible cash distribution. We therefore turn to heuristics instead. Reference [2] already introduces some prospective heuristics. This research mainly focuses on one of these heuristics, namely the MaxC heuristic. The definition of this heuristic along with an intuitive explanation and a justification for its use in this research is given in the following subsection. The subsections that follow do the same for another heuristic, called the Theta heuristic, introduced in [2] as well as for the new MaxCMinC heuristic introduced in this report.

2.2.1 MaxC heuristic

The heuristic that is the main focus of this report is the MaxC heuristic, a heuristic that was also proposed in [2]. This heuristic gives green light to states that have the maximum absolute amount of cash. Different versions or implementations of this heuristic could include variants that only give green light to one (specific) state belonging to the set $\{i||C_i(t)| = \max_j |C_j(t)|\}$. For example picking one at random, always picking, if possible, a state with positive cash or specifically one with negative cash. Though basic experiments have been performed on all these versions, this report only focuses on the variant that gives green light to all states with the maximum absolute amount of cash,

$$G_t = \text{argmax}_i |C_i(t)| = \{i||C_i(t)| = \max_j |C_j(t)|\}.$$

In [2] the exponential convergence of certain classes of green light strategies has already been given. These three classes of green light strategies are cyclic, completely random and generalized cyclic green light strategies. A cyclic green light strategy is one that gives green light to all states in some order and that same order keeps repeating itself. A completely random green light strategy, on the other hand, gives green light to a single randomly picked state each iteration, independent of anything else. It is clear that the MaxC heuristic does not fall under either of these two classes. The last class contains generalized cyclic green light strategies for Markov chains who's transition probability matrix has a Dobrushin coefficient smaller than one. Generalized cyclic green light strategies are strategies that give green light to all states within a finite number of iterations m: $\cup_{l=1}^m G_{mt+l} = \{1, 2, ..., n\}$, following the definition of [2]. Our first result in Lemma 1 states that the MaxC heuristic does not fall under the class of generalized cyclic strategies either.

Lemma 1. *The MaxC heuristic is, in general, not generalized cyclic.*

Proof. Take a 3 state Markov chain with states $i, j, k \in \{1, 2, 3\}$. Furthermore assume that the transition matrix only has zero entries on the diagonal. Let $C_i(t) = -C_j(t)$ and $C_k(t) = 0$ and therefore $G_t = \{i, j\}$. Let the transition probabilities to state k be the same for both other states, $p_{ik} = p_{jk} < 1$. In this case $C_k(t+1) = p_{ik}C_i(t) + p_{jk}C_j(t) = p_{ik}C_i(t) - p_{ik}C_i(t) = 0$ and $C_i(t+1) = p_{ji}C_j(t) = (1 - p_{jk})C_j(t) = -(1 - p_{ik})C_i(t) = -p_{ij}C_i(t) = -C_j(t+1)$. Hence recreating the previous situation with $G_t = \{i, j\}$, which will be repeated until convergence. As such, once this situation occurs state k will never get green light again making the MaxC heuristic not generalized cyclic. $\square$

With this we know that the MaxC heuristic does not belong to one the classes of green light strategies for which exponential convergence has already been proven. This will therefore be proven in this report.

2.2.2 MaxCMinC heuristic

The MaxCMinC heuristic is a natural extension of the MaxC heuristic that is first introduced in this report. The idea for this heuristic came to be during the analysis of the MaxC heuristic. This analysis showed that the MaxC heuristic can perform better when G contains both states with positive cash and states with negative cash. The MaxCMinC heuristic therefore gives green light to all states with the maximum amount of cash, which is, by definition, positive, and all states with the minimum amount of cash, which is, by definition, negative,

$$G_t = \{i|i \in \text{argmax}_j C_j(t) \vee i \in \text{argmin}_j C_j(t)\} = \{i|C_i(t) = \max_j C_j(t) \vee C_i(t) = \min_j C_j(t)\}.$$

For this heuristic is it also clear that it does not belong to any of the classes of green light strategies for which exponential convergence has already been proven, as it clearly isn't a cyclic or completely random green light strategy. It also does not fall under the class of generalized cyclic as stated in Lemma 2.

Lemma 2. *The MaxCMinC heuristic is, in general, not generalized cyclic.*

Proof. See the example given as proof of Lemma 1. As the manner in which green light is given also corresponds to the MaxCMinC heuristic it also shows that this heuristic is not generalized cyclic. $\square$

As the MaxCMinC heuristic also does not belong to one of the classes of green light strategies for which exponential convergence has already been proven, this will be done in this report.

2.2.3 Theta heuristic

The Theta heuristic is another heuristic that has already been proposed in [2]. This one is specifically interesting in the context of this report as it can be seen as a further extension of the MaxC and MaxCMinC heuristics. This heuristic gives green light to all states that have an absolute amount of cash that exceeds a threshold,

$$G_t = \{i||C_i(t)| \geq \theta_t\},$$

where the threshold, θ_t may vary with t and depend on the total cash, $\|C(t)\|_1$, present in the system.

In this report it will be shown that it is possible to choose θ_t in such a way the Theta heuristic gives green light to a subset of states that includes, at least, the same states as the MaxCMinC heuristic, $G_t(\text{MaxCMinC}) \subseteq G_t(\text{Theta})$. This is also the only context in which the Theta heuristic will be discussed.

3 MaxC heuristic for three state ergodic Markov chains

This chapter discusses the convergence of the MaxC heuristic for three state ergodic Markov chains with no self loops. In other words, the transition matrix only has zero entries on the diagonal, $p_{ii} = 0 \; \forall i \in \{1, 2, 3\}$. The first section proves exponential convergence of the algorithm under these conditions. The second section discusses the convergence rate found within the proof. Lastly, in the final section this value is compared to that of the power iteration method.

3.1 Exponential convergence of the MaxC heuristic

The main result of this section is given in Theorem 3. For its proof, we first proof that green light can only be given in two different ways: one for which $|G_t| = 1$ and another for which $|G_t| = 2$. After this each case is considered separately.

For the first case, where $|G_t| = 1$, we prove that $\|C(t+1)\|_1 < \|C(t)\|_1$ and that the next step will be guaranteed to belong to the second case, where $|G_t| = 2$.

The second case, where $|G_t| = 2$, is, in turn, divided into more cases depending on certain transition probabilities, with the exception of those on the diagonal, being equal to zero or one. We first proof that $\|C(t+1)\|_1 \leq \gamma \|C(t)\|_1$ for a $\gamma < 1$ when no additional transition probabilities are zero or one. After that we check what happens when certain transition probabilities are set to zero or one through the course of five cases. Each of these sub cases is illustrated by its own figure, see figures 2-6. In each of these sub cases we either show complete convergence after a fixed number of steps or show that the resulting Markov chain is no longer ergodic.

Lastly, the results of all these cases are combined to show the final result of Theorem 3.

Theorem 3. *Let P be the transition matrix of an ergodic Markov chain with 3 states and no self loops. The RLGL algorithm gives green light according to the MaxC heuristic. In this case there exists a $\gamma \in (0, 1)$ such that $\|C(t+4)\|_1 < \gamma \|C(t)\|_1$ and $\gamma \leq 1 - \min_{l,m \in \{1,2,3 | p_{lm} \neq 0\}} p_{lm}$.*

Proof. There are two different ways in which green light can be given according to the MaxC heuristic:

- Case 1: Green light is given to one state that has the absolute maximum amount of cash. $G_t = \{i\}$ for $i \in \{1, 2, 3\}$
- Case 2: Green light is given to two states that both have the absolute maximum amount of cash. $G_t = \{i, j\}$ for $i, j \in \{1, 2, 3\}$ $i \neq j$.

Let us now prove that no other case is possible and every step of the algorithm, except for the first step and any step taken after $\|C(t)\|_1 = 0$, belongs to either of these 2 cases. As all states getting green light would mean $|C_i(t)| = |C_j(t)| = |C_k(t)|$ while also $C_i(t) + C_j(t) + C_k(t) = 0$ so $C_i(t) = C_j(t) = C_k(t) = 0$, hence no step would need to be taken. The two possible cases will be regarded separately. For the remainder of the proof it may be assumed that $\{i, j, k\} = \{1, 2, 3\}$ are the names of the three states of the Markov chain in arbitrary order, as is illustrated in figure 1, unless noted otherwise.

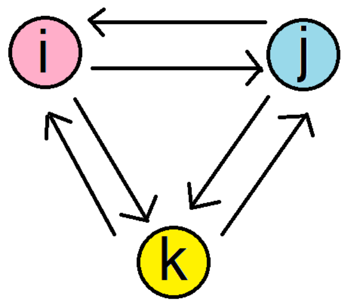


Figure 1: General representation of states of the Markov chain used in the proof of Theorem 3.

Case 1. Let state i have the absolute largest amount of cash, $G_t = \{i\}$, and assume without loss of generality that this amount is positive. Then we can argue that both other states will have a negative amount of cash, $\operatorname{sgn}(C_i(t)) = -\operatorname{sgn}(C_j(t))\ \forall j \in \{1,2,3\}\ j \neq i$. Indeed assume that $\exists j.\ j \in \{1,2,3\}$ such that $\operatorname{sgn}(C_i(t)) = \operatorname{sgn}(C_j(t))$. Then as $\sum_{m\in\{1,2,3\}} C_m(t) = 0$, $C_k(t) = -(C_i(t) + C_j(t))\ k \neq i,j$. But then $|C_k(t)| = |C_i(t) + C_j(t)| = |C_i(t)| + |C_j(t)| > |C_i(t)|$ and $|C_k(t)| > |C_j(t)|$ and state k would have received the green light instead thus leading to a contradiction.

So as both other states have been established to have cash of the opposite sign and all cash is guaranteed to be sent to a different state some cash will be guaranteed to cancel out at each of these steps. As such the total amount of cash is strictly decreasing, $\|C(t+1)\|_1 < \|C(t)\|_1$. The magnitude of this decrease can, however, be very small as it largely depends on the amount of cash present in the other states. For example if one of the states has an amount of cash that is very close to zero, $|C_k(t)| = \epsilon$ for some $\epsilon > 0$, meaning that $|C_j(t)| = |C_i(t)| - \epsilon$. Then if all cash gets send to state k, $p_{ik} = 1$, the total amount of cash will only decrease by an amount of ϵ, which can be very small.

In general after this step the cash distribution will be: $C_i(t+1) = 0$ and thus, as $\sum_{m\in\{1,2,3\}} C_m(t+1) = 0$, $C_j(t+1) = -C_k(t+1)$. If $C_j(t+1) = C_k(t+1) = 0$ the algorithm has converged and there won't be more steps. Otherwise $G_{t+1} = \{j,k\}$ making the next step one corresponding to Case 2.

Case 2. Let state i and j both have the absolute largest amount of cash, $G_t = \{i,j\}$. In this case we can argue that one of the states receiving green light will have positive cash, the other state receiving green light will have negative cash and therefore the third state will have zero cash; $C_i(t) = -C_j(t)$ so $C_k(t) = -(C_i(t) + C_j(t)) = 0$. Indeed instead assume that $C_i(t) \neq -C_j(t)$ so $C_i(t) = C_j(t)$. Then as $\sum_{n\in\{1,2,3\}} C_n(t) = 0$, $C_k(t) = -(C_i(t) + C_j(t)) = -2C_i(t)$. But then $|C_k(t)| > |C_i(t)|$ and $|C_k(t)| > |C_j(t)|$ and state k would have received the green light instead leading to a contradiction.

The cash distribution after this step will be: $C_i(t+1) = p_{ji}C_j(t)$, $C_j(t+1) = p_{ij}C_i(t)$ and $C_k(t+1) = p_{ik}C_i(t) + p_{jk}C_j(t)$. As these can be a wide variety of values the next step can belong to either case. If $0 \neq p_{ij} \neq p_{ji} \neq 0$ none of the values will be zero and the next step will corresponds to Case 1. We can argue that this is the only configuration for which the next step will corresponds to Case 1. Indeed it $p_{ij} = 0$ and/or $p_{ji} = 0$ it will lead to $C_j(t+1) = 0$ and/or $C_i(t+1) = 0$ respectively. And if $p_{ij} = p_{ji}$ then $p_{ik} = 1 - p_{ij} = 1 - p_{ji} = p_{jk}$ so $C_k(t+1) = 0$. Having one state with zero cash will always give a step corresponding to Case 2, moreover having more than one state with zero cash means that all states have zero cash and the algorithm has converged. This makes it impossible to get a step corresponding to Case 1 without the previously mentioned conditions.

If both $p_{ik} \neq 0$ and $p_{jk} \neq 0$ an amount of cash proportional to $\min(p_{ik}, p_{jk})$ will cancel in state k. If, however, either state doesn't send cash to state k no cash will be canceled. In general:

$$\begin{aligned}
\|C(t+1)\|_1 &= |C_i(t+1)| + |C_j(t+1)| + |C_k(t+1)| && \text{(3a)}\\
&= p_{ji}|C_j(t)| + p_{ij}|C_i(t)| + |p_{ik}C_i(t) + p_{jk}C_j(t)| && \text{(3b)}\\
&= (p_{ji} + p_{ij})|C_i(t)| + |(p_{ik} - p_{jk})C_i(t)| && \text{(3c)}\\
&= (p_{ij} + p_{ji} + |p_{ik} - p_{jk}|)|C_i(t)| && \text{(3d)}\\
&= 1/2(p_{ij} + p_{ji} + |p_{ik} - p_{jk}|)\,\|C(t)\|_1 && \text{(3e)}\\
&= 1/2(1 - p_{ik} + 1 - p_{jk} + |p_{ik} - p_{jk}|)\,\|C(t)\|_1 && \text{(3f)}\\
&= 1/2(2 - p_{ik} - p_{jk} + \max(p_{ik}, p_{jk}) - \min(p_{ik}, p_{jk}))\,\|C(t)\|_1 && \text{(3g)}\\
&= 1/2(2 - 2\min(p_{ik}, p_{jk}))\,\|C(t)\|_1 && \text{(3h)}\\
&= (1 - \min(p_{ik}, p_{jk}))\,\|C(t)\|_1 && \text{(3i)}\\
&\leq (1 - \min_{l,m\in\{1,2,3\}} p_{lm})\,\|C(t)\|_1\,. && \text{(3j)}
\end{aligned}$$

It can be seen that the total cash will indeed have decreased if and only if $p_{ik} \neq 0$ and $p_{jk} \neq 0$ or in other words $p_{ij} \neq 1$ and $p_{ji} \neq 1$. If either $p_{ik} = 0$ or $p_{jk} = 0$ ($p_{ij} = 1$ or $p_{ji} = 1$) the total cash does not decrease, $\|C(t+1)\|_1 = \|C(t)\|_1$, and the step would be called non decreasing.

We can argue that an infinite amount of non decreasing steps is possible if and only if the Markov chain is not ergodic. And if the Markov chain is ergodic there will come a decreasing step that corresponds to the Case 1 in the next three steps after the first non decreasing step and this step will result in convergence.

If either $p_{ij} = 1$ or $p_{ji} = 1$, knowing that $G_t = \{i, j\}$, we get a non decreasing step. Without loss of generality we assume $p_{ji} = 1$ for the first non decreasing step and we will consider case-by-case what can happen after this step. The complete starting situation of a sub case at iteration t are summarized in an accompanying picture.

Case 2.1. The starting situation is illustrated in figure 2. If $p_{ij} \neq 1$ and $p_{ij} \neq 0$ there will only be one non decreasing step. The cash distribution after this step: $C_i(t+1) = p_{ji}C_j(t) = C_j(t) = -C_i(t)$, $C_j(t+1) = p_{ij}C_i(t)$ and $C_k(t+1) = p_{ik}C_i(t) + p_{jk}C_j(t) = p_{ik}C_i(t)$. It can easily be seen that $G_{t+1} = \{i\}$, thus being a decreasing step corresponding to Case 1, and the cash becomes $C_i(t+2) = 0$, $C_j(t+2) = C_j(t+1)+p_{ij}C_i(t+1) = p_{ij}C_i(t)-p_{ij}C_i(t) = 0$ and $C_k(t+2) = C_k(t+1)+p_{ik}C_i(t+1) = p_{ik}C_i(t)-p_{ik}C_i(t) = 0$ which means the algorithm has converged in the decreasing step at t=1.

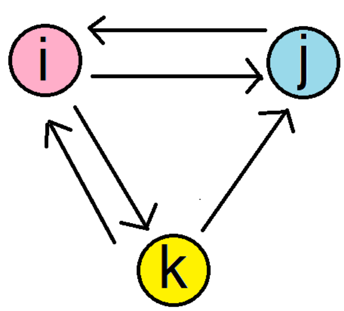


Figure 2: Representation of the Markov chain as used in Case 2.1. Here, as stated, $G_t = \{i, j\}$, $p_{ji} = 1$, $p_{ij} \neq 1$ and $p_{ij} \neq 0$.

Case 2.2. The starting situation is illustrated in figure 3. If $p_{ij} = 1$ there will be more decreasing steps as $C_i(t+1) = p_{ji}C_j(t) = C_j(t) = -C_i(t)$, $C_j(t+1) = p_{ij}C_i(t) = C_i(t) = -C_j(t)$ and $C_k(t+1) = C_k(t) = 0$ means that we are in a similar state as before the first non decreasing step, just with the signs of C_i and C_j swapped. As such the same non decreasing step will keep infinitely repeating while only swapping signs at each step. It is, however, also clear that this case concerns a non-ergodic Markov chain as state k cannot be reached from any of the other states.

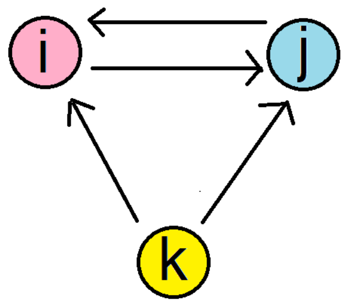


Figure 3: Representation of the Markov chain as used in Case 2.2. Here, as stated, $G_t = \{i, j\}$, $p_{ji} = 1$ and $p_{ij} = 1$.

Case 2.3. The starting situation is illustrated in figure 4. If $p_{ij} = 0$ it is guaranteed that there will be at least two non decreasing steps. After the first non decreasing step the cash distribution is $C_i(t+1) = C_j(t)$, $C_j(t+1) = 0$ and $C_k(t+1) = C_i(t)$ so clearly $G_{t+1} = \{i, k\}$ and as $p_{ik} = 1 - p_{ij} = 1$ it will again be a non decreasing step. If $p_{ki} \neq 1$ and $p_{ki} \neq 0$ it can easily be seen that the next steps are similar to Case (2.1) so after the third step $C_i(t+3) = C_j(t+3) = C_k(t+3) = 0$ and the algorithm has once again converged with the first decreasing step, this time occurring at t+2.

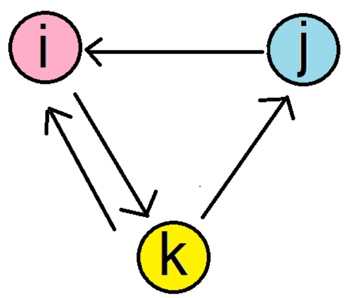


Figure 4: Representation of the Markov chain as used in Case 2.3. Here, as stated, $G_t = \{i, j\}$, $p_{ji} = 1$, $p_{ij=0}$, $p_{ki} \neq 1$ and $p_{ki} \neq 0$.

Case 2.4. The starting situation is illustrated in figure 5. If $p_{ij} = 0$ and $p_{ki} = 0$, so $p_{kj} = 1 - p_{ki} = 1$, we have created a cycle from j to i to k and back again. This means we will have infinitely many non decreasing steps as the positive and negative amounts of cash chase each other through the cycle. It also means that the Markov chain has a periodicity of three and is therefore not ergodic.

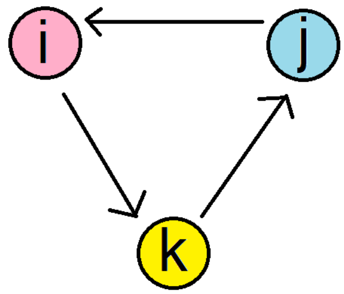


Figure 5: Representation of the Markov chain as used in Case 2.4. Here, as stated, $G_t = \{i, j\}$, $p_{ji} = 1$, $p_{ij=0}$ and $p_{ki} = 0$.

Case 2.5. The starting situation is illustrated in figure 6. If $p_{ij} = 0$ and $p_{ki} = 1$ that after the first non decreasing step we get a situation similar to Case 2.2 where two states, in this case states i and k, keep continuously interchanging their cash. There is also once again a state, state j, that is not reachable from any other state making the Markov chain non-ergodic.

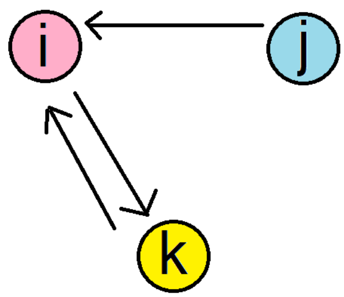


Figure 6: Representation of the Markov chain as used in Case 2.5. Here, as stated, $G_t = \{i, j\}$, $p_{ji} = 1$, $p_{ij=0}$ and $p_{ki} = 1$.

To summarize the above, an ergodic Markov chain can have two non decreasing steps at maximum before a decreasing step corresponding to Case 1 lets the algorithm reach convergence. So

$$\|C(t+3)\|_1 = 0 \leq \gamma \|C(t)\|_1 \tag{4}$$

for any positive γ if the step at iteration t was a non decreasing step, so when $\|C(t+1)\|_1 = \|C(t)\|_1$.

If, on the other hand, $p_{ij} \neq 1$ and $p_{ji} \neq 1$ equation (3) gives the rate of convergence. So for decreasing steps corresponding to Case 2 we have

$$\|C(t+1)\|_1 \leq \gamma \|C(t)\|_1 \tag{5a}$$

$$\text{with } \gamma = 1 - \min_{l,m \in \{1,2,3\}} p_{lm} < 1. \tag{5b}$$

In summary this gives that if we start with a step conforming to Case 2 at time t equation (4) or equation (5) will hold. And if we start with a step conforming to Case 1 at time t the amount of cash will be strictly decreasing and the next step is guaranteed to be conform to Case 2, as such it would give:

$$\|C(t+4)\|_1 = 0 \leq \gamma \|C(t+1)\|_1 < \gamma \|C(t)\|_1 \tag{6a}$$

$$\text{for any positive } \gamma < 1 \tag{6b}$$

if the step at $t+1$ is a non decreasing one. Or

$$\|C(t+2)\|_1 \leq \gamma \|C(t+1)\|_1 < \gamma \|C(t)\|_1 \tag{7a}$$
$$\text{with } \gamma = 1 - \min_{l,m \in \{1,2,3\}} p_{lm} < 1 \tag{7b}$$

if the step at $t+1$ is decreasing step. If we combine equations (4), (5), (6) and (7) we get that at any point we have at least

$$\|C(t+4)\|_1 < \gamma \|C(t)\|_1 \tag{8a}$$
$$\text{with } \gamma = 1 - \min_{l,m \in \{1,2,3\}} p_{lm} < 1. \tag{8b}$$

As such it can be concluded that the algorithm converges exponentially. □

Remark. The algorithm converges if $\|C(t)\|_1$ goes to zero. If, however, the history, $H(t)$, also goes to zero the algorithm will converge to the trivial solution and not give the desired answer. The occurrence of this situation seems to depend on the initial distribution of the cash, but this conjecture remains an open problem.

3.2 Convergence rate of the MaxC heuristic

Equation (8) proofs exponential convergence for the RLGL algorithm with the MaxC heuristic. For this it only matters that γ is smaller than 1. The estimation of the convergence rate can, however, be made more precise than that.

This can be done by firstly differentiating between two different scenarios. If no non-decreasing step is taken the total cash will decrease with $\gamma = 1 - \min_{i,j \in \{1,2,3\}} p_{ij}$ at least every 2 steps, as in equation (7). Whereas if a non-decreasing step is taken, convergence will be reached after a maximum of 3 steps, as in equation (6). Therefore, the first case will dominate the general rate of convergence and the second is a unique case where the algorithm quickly reaches exact convergence.

We first discuss the second case. This case is an interesting one that can only occur if one (or more) of the transition probabilities is zero, $p_{ij} = 0$ for some $i, j \in \{1,2,3\}$, $i \neq j$. It triggers when a non decreasing step is taken after which the error, which is proportional to $\|C(t)\|_1$, instantly drops to zero after staying the same for one or two steps. This can speed up the convergence, especially if it is triggered in one of the first few steps.

For the first case it should be noted that this convergence rate does not necessarily give a tight bound. The first bit of accuracy is lost from step i to step j in equation (3) by taking the global minimum. In each step where green light is given to two state there are only two relevant transition probabilities, as can be seen in step i from equation (3). The global minimum is only taken as it is impractical to have the convergence rate depend on the states receiving green light. Because of this it is possible for the transition probability causing the global minimum to be irrelevant for some or even most of the steps. For example, one of the transition probabilities being close to zero will cause γ to be close to 1, but if this transition probability is never one of the relevant ones the effective γ could be a lot smaller. Taking the global minimum can therefore somewhat reduce the accuracy of the convergence rate.

Another point that indicates that the bound may be made more tight is that we only consider a step with one state receiving green light as strictly decreasing. It is possible for very little cash to cancel during these steps. It is, however, also possible for almost all cash to cancel during these steps. This makes it possible that far more cash disappears over the course of two steps than indicated by the convergence rate.

Lastly it should be noted that while (3) gives the convergence rate over two steps, this is only the case if in one of those steps green light is given to only one state. If green light is given to two states for multiple steps in a row γ will be the convergence rate for each of these steps, as indicated in equation (5).

To summarize, convergence faster than indicated in (3) is possible if one or more of the following applies: the transition probability indicating the global minimum isn't relevant in most of the steps, a large portion of cash cancels when only one state receives green light, two states receive green light for multiple steps in a row, or a non decreasing step triggers an early end for the algorithm. Some of this will be shown in practice in chapter 5.

3.3 Theoretical comparison to power iteration

This section will compare the convergence rate of the MaxC heuristic to that of the power iteration method. This is to get a better idea of how well the MaxC heuristic will perform in comparison to existing methods.

It is well known that with power iteration the convergence is dominated by powers of the second largest eigenvalue, $|\lambda_2|$, of the transition matrix P. For a 3 state Markov chain with zeros on the diagonal it can easily be calculated that

$$|\lambda_2| = \left| \frac{1}{2} \left(\sqrt{1 + 4(p_{21}(p_{12} + p_{31} - 1) - p_{12}p_{31})} + 1 \right) \right|, \tag{9}$$

by using the fact that with a zero diagonal $p_{13} = 1 - p_{12}$ and similarly for the other rows which leaves us with 3 variables. This can be compared to the convergence rate found in the proof from the previous section while taking into account that this rate generally ranges over 2 steps:

$$\gamma = 1 - \min_{i,j \in \{1,2,3 | p_{ij} \neq 0\}} p_{ij}. \tag{10}$$

The two convergence rates have been compared to each other by calculating the value for different values of p_{12}, p_{21} and p_{31} and indicating which is smaller in a plot. The plot for $p_{31} = 0.51$ can be found in figure 7.

The full comparison shows that $|\lambda_2|$ is mostly smaller or equal to γ, which could mean that the power iteration method would almost always converge faster or just as fast as the RLGL algorithm that uses the MaxC heuristic. As discussed in the previous section, however, it is possible that γ doesn't give a tight bound.

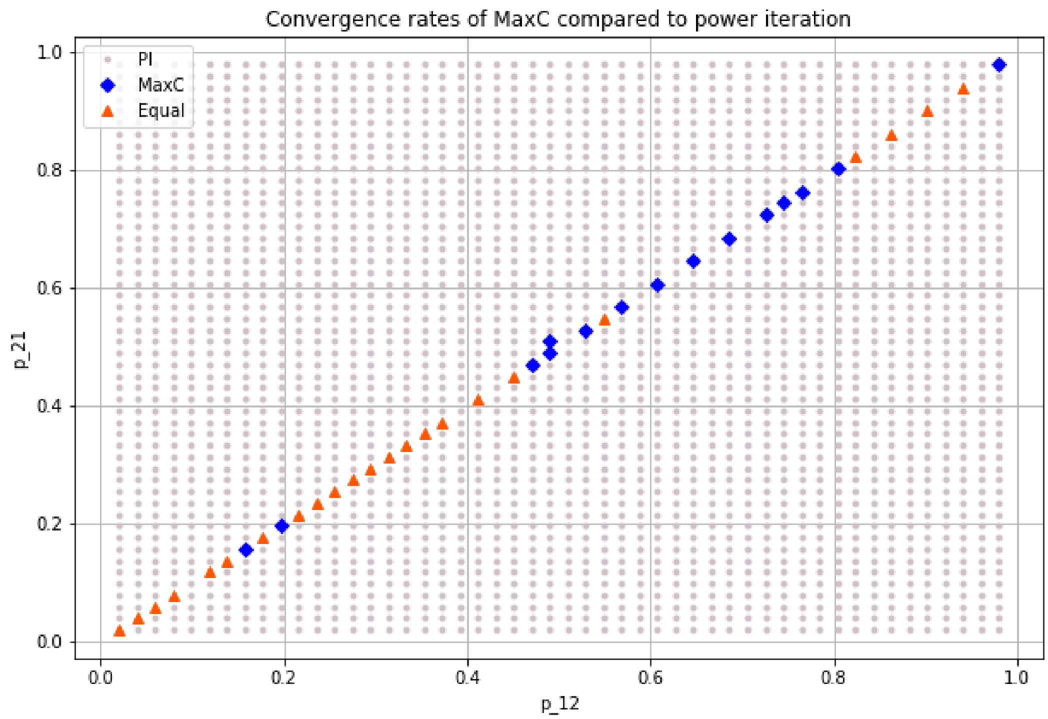


Figure 7: Comparison between γ and $|\lambda_2|$ for $p_{31} = 0.51$. Which value is the smallest, the MaxC heuristic for the blue diamonds and the power iteration method for the grey dots, or whether they are equal, the red triangles, is indicated at each point.

It should also be mentioned that when it comes to computational cost, one step of the MaxC heuristic is roughly half as expensive as one step of power iteration. This can be argued as follows. Assuming there are no additional transition probabilities that are zero and none are equal. In this case the MaxC heuristic will alternate between one state and two states receiving green light, whereas power iteration will always give 'green light' to all three states. This comes to roughly double the cost when taking multiple steps into account. This would make the performance of MaxC better than that of power iteration when the difference between γ and $|\lambda_2|$ is small.

How the MaxC heuristic compares to power iteration in practice is discussed in chapter 5.

4 Estimating the convergence rate for a random walk on a complete graph

The previous chapter proved the exponential convergence of the MaxC heuristic for the specific case of a three state ergodic Markov chain without self loops. This chapter generalizes this statement to hold for larger Markov chains. To make this generalization to n states, we need to impose additional conditions. This chapter will therefore only consider Markov chains that have a transition matrix with only positive entries, or, in other words, are a random walk on a complete graph. The first section gives an rough estimate of the convergence rate of three heuristics for general complete graphs. In the second section the estimate is made more precise by considering complete graphs with a specific structure, specifically the Mean-Field Stochastic Block model.

4.1 General complete graphs

4.1.1 MaxC heuristic

As also seen in chapter 3, the MaxC heuristic can give green light in multiple different ways. Each of these ways influences the convergence rate and is thus considered, like green light being given to one or more states with only positive cash, or a mixture of states with positive and negative cash. By considering each of these option, or configurations, in which green light can be given we proof exponential convergence for the MaxC heuristic, as stated in Theorem 4. The convergence rate of this heuristic will, however, remain dependent on the configuration in which green light is given. We know, for example that the heuristic, in general, converges faster if green light has been given to both a state with positive and a state with negative cash than when green light is given to a few states that all have positive cash. Additionally, more states receiving green light results in faster convergence. This is also why further guarantee of what kind of states will receive green light can result in a better estimate of the convergence rate as seen in the next section 4.2.

Theorem 4. *Take a Markov chain with n states and no zero transition probabilities, such that $p_{ij} > 0$ for all $i, j \in \{1, ..., n\}$. The RLGL algorithm gives green light according to the MaxC heuristic. In the case that n is finite there exists a $\gamma \in (0, 1)$ for every configuration in which green light can be given such that $\|C(t+1)\|_1 \leq \gamma \|C(t)\|_1$ and $\gamma \leq 1 - \frac{2}{n} \min_{i,j\in\{1,...,n\}, i\neq j} p_{ij}$.*

Proof. There are multiple ways in which green light can be given according to the MaxC heuristic, these will be discussed case by case. Mind that some of these ways cannot occur unless n is sufficiently large. At any time $i, j, k \in \{1, 2, ..., n\}$ are arbitrary states of the Markov chain unless specified otherwise. For every case it is clear, by the pigeon hole principle, that for every state i that receives green light from the MaxC heuristic at iteration t it holds that $|C_i(t)| \geq \frac{1}{n} \|C(t)\|_1$.

Case 1. Assume green light is given to only one state, say state i, $G_t = \{i\}$, and without loss of generality we assume that $C_i(t)$ is positive. In this case some cash will cancel out in every state j that has negative cash. The amount that disappears when transferring cash from state i to state j is $2\min(|C_j(t)|, p_{ij}|C_i(t)|)$. Moreover $\exists j$, $j \in \{1, ..., n\}$, $j \neq i$ and $\operatorname{sgn} C_j(t) = -\operatorname{sgn} C_i(t)$ such that $|C_j(t)| \geq p_{ij}|C_i(t)|$.

Indeed if $|C_j(t)| < p_{ij}|C_i(t)|$ $\forall j \in \{1, ..., n| \operatorname{sgn} C_j(t) = -\operatorname{sgn} C_i(t)\}$ then

$$\begin{aligned}\sum_{\{j\in\{1,...,n\}|\operatorname{sgn} C_j(t) = -\operatorname{sgn} C_i(t)\}} |C_j(t)| &< \sum_{\{j\in\{1,...,n\}|\operatorname{sgn} C_j(t) = -\operatorname{sgn} C_i(t)\}} p_{ij}|C_i(t)| \\ &\leq \sum_{j\in\{1,...,n\}} p_{ij}|C_i(t)| = |C_i(t)|.\end{aligned}$$

This is, however, not possible as the sum over all cash should be zero. So as $C_i(t)$ is positive, the sum of over all negative cash may not be smaller than $C_i(t)$.

In total this means that if only one state, say state i, receives green light, then:

$$\begin{aligned}\|C(t+1)\|_1 = \|C(t)\|_1 - & \sum_{\{j\in\{1,\dots,n\} \mid \operatorname{sgn} C_j(t) = -\operatorname{sgn} C_i(t)\}} 2\min(|C_j(t)|, p_{ij}|C_i(t)|) && \text{(11a)}\\ &\le \|C(t)\|_1 - 2 \min_{j\in\{1,\dots,n\}} p_{ij}|C_i(t)| && \text{(11b)}\\ &\le (1 - \frac{2}{n} \min_{j\in\{1,\dots,n\}} p_{ij}) \|C(t)\|_1 && \text{(11c)}\\ &\le (1 - \frac{2}{n} \min_{i,j\in\{1,\dots,n\}, i\neq j} p_{ij}) \|C(t)\|_1 . && \text{(11d)}\end{aligned}$$

The step made from (11)a to (11)b takes a single element from the sum, specifically the unspecified one for which $|C_j(t)| \ge p_{ij}|C_i(t)|$ is guaranteed, and takes the minimum to remove the dependence on this unspecified state. So in this case

$$\|C(t+1)\|_1 \le \gamma \|C(t)\|_1 \tag{12a}$$

$$\text{with } \gamma = 1 - \frac{2}{n} \min_{i,j\in\{1,\dots,n\}, i\neq j} p_{ij} < 1. \tag{12b}$$

Case 2. If green light is given to two states, say states i and j, $G_t = \{i,j\}$, and $C_i(t) = -C_j(t)$ some cash is guaranteed to cancel out at each state, regardless of the cash that is present in those states. The cash that is guaranteed to cancel out at any state $k \in \{1,...,n\}$ is $2\min(p_{ik}|C_i(t)|, p_{jk}|C_j(t)|) = 2|C_i(t)|\min(p_{ik}, p_{jk})$. The amount of cash send to state k that has not canceled out yet is $|p_{ik} - p_{jk}||C_i(t)|$. If $k \neq i$ and $k \neq j$ it is possible for more cash to cancel out in that state depending on the sign of the cash already present at state k and the sign of the state that send the higher fraction of cash. This amount is, however, difficult to quantify without more information.

In total this means that if 2 states, say states i and j, receive green light and $C_i(t) = -C_j(t)$ then at least:

$$\begin{aligned}\|C(t+1)\|_1 &\le \|C(t)\|_1 - \sum_{k\in\{1,\dots,n\}} 2|C_i(t)|\min(p_{ik}, p_{jk}) && \text{(13a)}\\ &\le \|C(t)\|_1 - \frac{2}{n}\|C(t)\|_1 \sum_{k\in\{1,\dots,n\}} \min(p_{ik}, p_{jk}) && \text{(13b)}\\ &\le \left(1 - \frac{2}{n} \sum_{k\in\{1,\dots,n\}} \min_{l\in\{1,\dots,n\}} p_{lk}\right) \|C(t)\|_1 && \text{(13c)}\\ &\le (1 - \frac{2}{n} n \min_{k,l\in\{1,\dots,n\}} p_{lk}) \|C(t)\|_1 && \text{(13d)}\\ &= (1 - 2 \min_{k,l\in\{1,\dots,n\}} p_{lk}) \|C(t)\|_1 . && \text{(13e)}\end{aligned}$$

So in this case

$$\|C(t+1)\|_1 \le \gamma \|C(t)\|_1 \tag{14a}$$

$$\text{with } \gamma = 1 - 2 \min_{k,l\in\{1,\dots,n\}} p_{lk} < 1. \tag{14b}$$

Case 3. If green light is given to 2 states, say states i and j, $G_t = \{i,j\}$, and $C_i(t) = C_j(t)$ the steps are similar to Case 1. Again $\exists k$, $k \in \{1,...,n\}$, $k \neq i$ and $k \neq j$ and $\operatorname{sgn} C_k(t) = -\operatorname{sgn} C_i(t) = -\operatorname{sgn} C_j(t)$ such that $|C_k(t)| \geq p_{ik}|C_i(t)| + p_{jk}|C_j(t)|$ by similar arguments as given in Case 1. Following the same steps gives:

$$\|C(t+1)\|_1 = \|C(t)\|_1 - \sum_{\{k \in \{1,\ldots,n| \operatorname{sgn} C_k(t) = -\operatorname{sgn} C_i(t)\}} 2\min(|C_k(t)|, p_{ik}|C_i(t)| + p_{jk}|C_j(t)|) \tag{15a}$$

$$\leq (1 - \frac{4}{n} \min_{i,j \in \{1,\ldots,n\}, i \neq j} p_{ij}) \|C(t)\|_1 . \tag{15b}$$

So in this case

$$\|C(t+1)\|_1 \leq \gamma \|C(t)\|_1 \tag{16a}$$

$$\text{with } \gamma = 1 - \frac{4}{n} \min_{i,j \in \{1,\ldots,n\}, i \neq j} p_{ij} < 1. \tag{16b}$$

Any further case, in which green light is given to 3 or more states, follows a similar logic to the cases discussed above. If there are at least two states, among all the states receiving green light, that have positive and negative cash respectively the logic from Case 2 that leads to equation (13) applies to each pair of states with positive and negative cash respectively. So if the number of these pairs equals m, than following the same logic for each pair and adding it all up gives:

$$\|C(t+1)\|_1 \leq (1 - 2m \min_{i,j \in \{1,\ldots,n\}} p_{ij}) \|C(t)\|_1 . \tag{17}$$

If, on the other hand, all states receiving green light have the same sign the logic from Case 3, and consequently Case 1, applies:

$$\|C(t+1)\|_1 \leq (1 - \frac{2m}{n} \min_{i,j \in \{1,\ldots,n\}, i \neq j} p_{ij}) \|C(t)\|_1 \tag{18}$$

with m being the number of states receiving green light at the same time.

In total for each configuration in which green light can be given there is a $\gamma < 1$ such that $\|C(t+1)\|_1 \leq \gamma \|C(t)\|_1$, so the MaxC heuristic converges exponentially for general complete graphs. $\square$

Remark. In general complete graphs, without any clear structure, m most likely will not be larger than 1. This holds for both equation (17) and (18).

Remark. Note that equations (17) and (18) give very rough estimations of the convergence rate. Equation (18) uses that there is at least one state in which all cash send to that state is canceled out and then only considers the cash that would, at minimum, cancel out in that state. It does not consider the cash that cancels out in other states. Similarly, equation (17) only considers the cash that is canceled out among the cash that is moved. Not all cash send to other states is canceled out and some of the remainder can still cancel out with the cash already present in the state it is send to.

4.1.2 MaxCMinC heuristic

The MaxCMinC heuristic is a generalization of the MaxC heuristic. To be more specific, it expands on one of the ways in which the MaxC heuristic can give green light. As such proving exponential convergence of the MaxCMinC heuristic follows the same reasoning as the steps leading to equation (17).

Theorem 5. *Consider a Markov chain with n states and no zero transition probabilities. Assume that the RLGL algorithm gives green light according to the MaxCMinC heuristic. Then there exists a $\gamma \in (0,1)$ such that $\|C(t+1)\|_1 \leq \gamma \|C(t)\|_1$ and and $\gamma \leq 1 - \min_{k,l \in \{1,\ldots,n\}} p_{lk}$.*

Proof. For every state that receives green light from the MaxCMinC heuristic at iteration t it holds that it either has the largest positive or the largest negative amount of cash. The sum of all positive cash is the same as the absolute value of the sum of all negative cash and is equal to half of all cash present in the system,

$$\sum_{i\in\{1,\ldots,n|\,\operatorname{sgn} C_i(t)=+1\}} C_i(t) = \frac{1}{2}\left\|C(t)\right\|_1 = -\sum_{i\in\{1,\ldots,n|\,\operatorname{sgn} C_i(t)=-1\}} C_i(t).$$

By the pigeon hole principle as above, for a state with maximal positive cash,

$$\max_i C_i(t) \geq \frac{1}{|\{i|\operatorname{sgn} C_i(t)=+1\}|}\frac{1}{2}\left\|C(t)\right\|_1 \geq \frac{1}{2n}\left\|C(t)\right\|_1,$$

with $\{i|\operatorname{sgn} C_i(t)=+1\}$ the subset of $\{1,...,n\}$ containing all states with positive cash. And similarly for a state with maximal negative cash;

$$|\min_i C_i(t)| = -\min_i C_i(t) \geq \frac{1}{|\{i|\operatorname{sgn} C_i(t)=-1\}|}\frac{1}{2}\left\|C(t)\right\|_1 \geq \frac{1}{2n}\left\|C(t)\right\|_1.$$

In this heuristic at least 2 states get green light, one with positive cash and an other with negative, so it will follow a reasoning similar to Case 2 in the proof of Theorem 4. There are at least 2 states i and j, $i,j\in\{1,...,n\}$, such that $i,j\in G_t$ and $\operatorname{sgn} C_i(t) = -\operatorname{sgn} C_j(t)$, so some cash is guaranteed to disappear at each state regardless of the cash present in those states. The cash that is guaranteed to disappear at any state $k\in\{1,...,n\}$ for each pair of states with respectively positive and negative cash is $2\min(p_{ik}|C_i(t)|, p_{jk}|C_j(t)|)$. This means that

$$\begin{aligned}
\left\|C(t+1)\right\|_1 &\leq \left\|C(t)\right\|_1 - \sum_{k\in\{1,\ldots,n\}} 2*\min(p_{ik}|C_i(t)|, p_{jk}|C_j(t)|) &\text{(19a)}\\
&\leq \left\|C(t)\right\|_1 - \sum_{k\in\{1,\ldots,n\}} 2*\min(p_{ik}\frac{1}{2n}\left\|C(t)\right\|_1, p_{jk}\frac{1}{2n}\left\|C(t)\right\|_1) &\text{(19b)}\\
&= \left\|C(t)\right\|_1 - \frac{2}{2n}\left\|C(t)\right\|_1 \sum_{k\in\{1,\ldots,n\}} \min(p_{ik}, p_{jk}) &\text{(19c)}\\
&\leq (1-\frac{2}{2n}n\min_{k,l\in\{1,\ldots,n\}} p_{lk})\left\|C(t)\right\|_1 &\text{(19d)}\\
&= (1-\min_{k,l\in\{1,\ldots,n\}} p_{lk})\left\|C(t)\right\|_1 &\text{(19e)}
\end{aligned}$$

for the one pair of states that is guaranteed to exist. Including a step similar to the one leading to equation (17) brings us to

$$\begin{aligned}
&\left\|C(t+1)\right\|_1 \leq \gamma\left\|C(t)\right\|_1 &\text{(20a)}\\
&\text{with } \gamma = 1 - m\min_{k,l\in\{1,\ldots,n\}} p_{lk} < 1 &\text{(20b)}\\
&\text{and } m = \min(|\{i\in G_t|\operatorname{sgn} C_i = +1\}|, |\{i\in G_t|\operatorname{sgn} C_i = -1\}|). &\text{(20c)}
\end{aligned}$$

This proofs that the MaxCMinC heuristic converges exponentially. $\square$

Remark. In general complete graphs, without any clear structure, m most likely will not be larger than 1.

Remark. Like equation (17), equation (20) gives only a rough estimation of the convergence rate.

4.1.3 Theta heuristic

As a reminder, the Theta heuristic is a heuristic that gives green light to all states that have an absolute amount of cash that exceeds a threshold. Following the reasoning from the beginning of the proof to Theorem

5 shows that is possible to choose the threshold of the Theta heuristic such that it generalizes the MaxCMinC heuristic. If the threshold is set to be smaller or equal to $\frac{1}{2n}\left\|C(t)\right\|_1$ green light is given to at least one state with positive cash and at least one state with negative cash. Indeed as the argument from the previous proof states that for both the maximal positive and the maximal negative cash $|C_i(t)| \geq \frac{1}{2n}\left\|C(t)\right\|_1$. This means that if $\theta_t \leq \frac{1}{2n}\left\|C(t)\right\|_1$ the same states that would have received green light from the MaxCMinC heuristic also get green light from the Theta heuristic along with, most probably, some other states. Moreover, the arguments made to proof the exponential convergence of the MaxCMinC heuristic hold regardless of what the other states do. Therefore, if the restriction on θ_t holds, Theorem 5 can be extended to include the Theta heuristic with the proof remaining valid.

Theorem 6 (Extension to Theorem 5)**.** *Under the same conditions as given in Theorem 5 the RLGL algorithm instead gives green light according to the Theta heuristic with $\theta_t \leq \frac{1}{2n}\left\|C(t)\right\|_1$. Then there is a $\gamma \in (0,1)$ such that $\left\|C(t)\right\|_1 \leq \gamma\left\|C(t)\right\|_1$ and $\gamma \leq 1 - \min_{k,l\in\{1,\ldots,n\}} p_{lk}$.*

Proof. Green light is give to all states i such that $|C_i| \geq \theta_t$. By the proof to Theorem 5:

$$\max_{i\in\{1,..,n\}} C_i(t) \geq \frac{1}{2n}\left\|C(t)\right\|_1 \geq \theta_t$$

and

$$|\min_{i\in\{1,..,n\}} C_i(t)| \geq \frac{1}{2n}\left\|C(t)\right\|_1 \geq \theta_t.$$

Therefore equations (19) and (20) hold here as well. □

Remark. Equation (20) gives an even rougher estimate for the Theta heuristic than it did for the MaxCMinC heuristic. This is as it also fails to include the cash that cancels out as a result of the extra states receiving green light from the Theta heuristic with respect to the MaxCMinC heuristic.

4.2 Mean-Field Stochastic Block Model

As previously mentioned, the estimates found for the convergence rate in the previous section are quite rough. It is possible to make these estimates more precise when more assumptions are made. In this section we make assumptions about the structure of the transition matrix. For this we use an simple model for random graphs that is also analysed in [2].

Stochastic Block Models (SBM) [13] are simple random graphs with a clustered structure. This graph has multiple clusters or block of states. Say it has $b \geq 2$ blocks, B_1, ..., B_b, with sizes N_1, ..., N_b. Without loss of generality we assume the blocks to be ordered in descending order with respect to the size, $N_1 \geq N_2 ... \geq N_b$. States within these blocks are connected with probability p and states that each belong to a different block are connected with probability q.

As the results of the previous section hold for complete graph, we specifically consider the Mean-Field SBM. This is a graph with edge weights corresponding to the probability of an edge and has transition probabilities corresponding to these edge weights. Take $n = \sum_{k\in\{1,...b\}} N_k$, then the transition probabilities, p_{ij} will be:

$$p_{ij} = \frac{p}{pN_k + q(n - N_k)} \text{ with } i,j \in B_k \text{ and } k \in \{1,...,b\} \tag{21}$$

and

$$p_{ij} = \frac{q}{pN_k + q(n - N_k)} \text{ with } i \in B_k,\ j \notin B_k \text{ and } k \in \{1,...,b\}. \tag{22}$$

Another assumption made in this section is that the initial step of the RLGL algorithm distributes the cash uniformly. The next subsections discuss how the knowledge given by these assumptions can be used to make the estimation of the convergence rate more precise.

4.2.1 MaxC heuristic

As we assume the initial step distributes the cash uniformly, the cash distribution among the states of the Mean-Field SBM will always remain uniform over the blocks. Indeed, all states within a block are identical according to the model, so as they will all behave in the same way they will also all have the same amount of cash after the initial step, $C_i(1) = C_j(1)\ \forall i, j \in B_k,\ k \in \{1, ..., b\}$. Therefore, if one state in a block receives green light from the MaxC heuristic, the entire block will receive green light. As such, all states in a block will continue to be identical for the whole duration of the algorithm, $C_i(t) = C_j(t)\ \forall i, j \in B_k,\ k \in \{1, ..., b\}$, $\forall t \geq 0$.

We first look at a model with only two blocks. If one block has a state with positive cash, all states in that block will have the same amount of positive cash and therefore the other block will only have states with negative cash so that the total cash will be zero. So if $\operatorname{sgn} C_i = +1$ for $\forall i \in B_1$, $\operatorname{sgn} C_j = -1\ \forall j \in B_2$. As such, the cash value of each state is half the total cash divided by the size of the block, $|C_i(t)| = \frac{1}{2N_1}\|C(t)\|_1$, $\forall i \in B_1$. This also means that all states in the smallest block have the absolute maximum amount of cash and, therefore, all receive green light from the MaxC heuristic. And as this situation will hold for all t, the MaxC heuristic will always give green light to the smallest block, as [2] has determined to be the optimal strategy for a Mean-Field SBM with two blocks.

The knowledge of the structure of the graph, as well as of the states that receive green light, given by the used model can be used to make the convergence rate more precise. We follow the logic of Cases 1 and 3 from the proof to Theorem 4 and know that the two blocks have cash of opposite sign. We, therefore, know that given $G_t = B_2\ \exists j,\ j \in B_1$ such that $|C_j(t)| = \frac{1}{2N_1}\|C(t)\|_1 \geq N_2 \frac{q}{qN_1+pN_2}|C_i(t)| = N_2 \frac{q}{qN_1+pN_2}\frac{1}{2N_2}\|C(t)\|_1$, with $i \in B_2$ an arbitrary state. But because all states in a block are identical this holds for all $j \in B_1$. As such the amount of cash disappearing in each state is $2\min(|C_j(t)|, N_2\frac{q}{qN_1+pN_2}|C_i(t)|) = 2\frac{q}{qN_1+pN_2}\frac{1}{2}\|C(t)\|_1$. As such the final equation becomes:

$$\|C(t+1)\|_1 = \|C(t)\|_1 - \sum_{j \in B_1} \frac{q}{qN_1 + pN_2}\|C(t)\|_1 \tag{23a}$$

$$= \|C(t)\|_1 - \frac{qN_1}{qN_1 + pN_2}\|C(t)\|_1 \tag{23b}$$

$$= \frac{pN_2}{qN_1 + pN_2}\|C(t)\|_1 . \tag{23c}$$

This gives the precise value for the convergence rate as opposed to the estimates of the previous section. It should also be noted that this value is equal to the one found in section 4 of [2] regardless of being derived by tightening the upper bound on the convergence rate instead of deriving it directly.

We now look at a model with b blocks where $b > 2$. In this case we know less than in a model with only 2 blocks. We can, however, still make the estimate of equation (18) tighter. It is still known that at least one whole block will receive green light, say this is block $B_k, k \in \{1, ..., b\}$. As such there are at least $N_k \geq N_b$ states that receive green light. And while still following the logic leading to equations (1) and (3) similar to the case of two, we know that $\exists j \in \{1, ..., n\}$ with $\operatorname{sgn} C_j(t) \neq \operatorname{sgn} C_i(t)\ \forall i \in B_k$ such that $|C_j(t)| \geq \sum_{i \in B_k} p_{ij}|C_i(t)|$ and as all states in a block are identical, $\exists B_l,\ l \in \{1, ..., b\}$ with $\operatorname{sgn} C_j(t) \neq \operatorname{sgn} C_i(t)\ \forall j \in B_l$ and $\forall i \in B_k$ such that $\forall j \in B_l$, $|C_j(t)| \geq N_k \frac{q}{pN_k + q(n-N_k)}|C_i(t)|$ for $i \in B_k$ an arbitrary state. Bringing all of this together gives

$$\|C(t+1)\|_1 \leq \|C(t)\|_1 - \sum_{j \in B_l} \frac{2qN_k}{pN_k + q(n-N_k)} |C_i(t)| \tag{24a}$$

$$= \|C(t)\|_1 - N_l \frac{2qN_k}{pN_k + q(n-N_k)} |C_i(t)| \tag{24b}$$

$$= \|C(t)\|_1 - \frac{N_l}{n} \frac{2qN_k}{pN_k + q(n-N_k)} \|C(t)\|_1 \tag{24c}$$

$$\leq (1 - \frac{N_b}{n} \frac{2qN_k}{pN_k + q(n-N_k)}) \|C(t)\|_1 \tag{24d}$$

$$\leq \left(1 - \frac{2N_b}{n} \min_{k \in \{1,\ldots,b\}} \frac{qN_k}{pN_k + q(n-N_k)}\right) \|C(t)\|_1 . \tag{24e}$$

This convergence rate, despite still being an estimate, generally predicts a faster convergence than the one found in equation (18) and can therefore be considered a tighter bound. The convergence rate found in equation (17), on the other hand, could still relevant even in this case if green light is given to more than one block, which is relatively unlikely unless 2 blocks have the same size. If those blocks then each have cash of opposite sign, equation (17) will function as is with the value of m being equal to the size of the smallest block receiving green light. This would still be a better bound than one would expect for general complete graphs, where the value of m is not expected to be much larger than 1.

4.2.2 MaxCMinC heuristic

As also discussed in the previous section, all states in a block are identical and will therefore, at any time, have the same amount of cash. The MaxCMinC heuristic will therefore always give green light to at least two entire blocks, one with positive and one with negative cash.

Models with only two blocks are not very interesting to consider for the MaxCMinC heuristic. This is as, in this case, the MaxCMinC heuristic simply gives green light to all states. This gives the same result as when using the power iteration method, which has already been extensively analysed in existing literature. For this specific case, the Mean-Field SBM with 2 blocks, it can also be easily calculated to be:

$$\gamma = |\lambda_2| = \frac{(p^2 - q^2)N_1}{(pN_1 + qN_2)(pN_2 + qN_1)}. \tag{25}$$

For a larger number of blocks, $b > 2$, a similar reasoning holds as in the last part of the previous subsection. The original analysis from subsection 4.1.2 still holds. The only change is that we have a clearer indication on the value of m from equation (20). As green light is given to at least two blocks, this value, m, is as least the size of the smallest block receiving green light. This value is again bounded by the size of the smallest block so $m \geq N_b$.

5 Numerical results

The previous chapters provided theoretical bounds on the convergence rate of the MaxC heuristic and the MaxCMinC heuristic. This chapter tests how well these bounds hold in practice. The results of the implementations of these heuristics are compared to their theoretical bounds as well at to each other. For reference, they are also compared to the power iteration method. This method has been given a simple implementation as a special case of the RLGL algorithm, as proven in [2]. Power iteration can therefore be considered a heuristic that at any iteration t gives green light to all states.

The comparison is made by looking at the normalized error, $\frac{\|\hat{\pi_t}-\hat{\pi_{t-1}}\|_1}{\|\hat{\pi_2}-\hat{\pi_1}\|_1}$, set out against either t, the number of steps taken, or against an indication of the number of operations needed. In most cases, the number of states receiving green light at each step, $|G_t|$, is taken as the indication of the number of operations. This is as, specifically for complete graphs, any state receiving green light gives the same number of additional operations, making the total number of operations at each step proportional to $|G_t|$. The initial step taken by the algorithm is not taken into account as this is the same for all heuristics and therefore won't provide additional information. And as the heuristics have been proven to converge exponentially, the error is plotted on a semi log scale.

The results of section 5.1 only discuss the MaxC heuristic with a three state ergodic Markov chain without self loops as the example. This can be seen as the numerical continuation of section 3.3. Section 5.2 includes both heuristics for example Markov chains that are random walks on a complete graph. Two different models are used to create this Markov chain, each of which is discussed separately.

5.1 Three state ergodic Markov chain

Section 3.3 shows that, according to the theoretical values, the power iteration method would mostly outperform the MaxC heuristic when it comes to convergence rate. This does, however, not mean that this is also the case in practice. The MaxC heuristic mostly performs far better than its theoretical bound, often overtaking the power iteration method in the process. An example of this can be seen in figure 8.

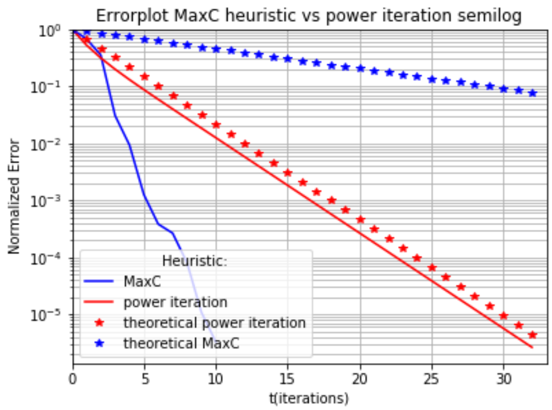
In this figure the normalized error progression of the MaxC heuristic is compared to its theoretical bound. The normalized error progression of the power iteration method for the same Markov chain, as well as its theoretical bound found in (9), are also shown. It can clearly be seen that the error progression of the power iteration nicely follows its theoretical bound, confirming the validity of the calculated bound.

Figures 8c and 8d are zoomed in versions of figures 8a and 8b respectively that show the the line depicting the MaxC heuristic more clearly. Figures 8a and 8c show the error progression set out against t, the number of iterations of the algorithm. Figures 8b and 8d show the same error progression set out against an indication for the number of operations instead. In this case, the number of states receiving green light each iteration is used as an indication for this value, this is done as, in this case, any state receiving green light will send cash to the same number of states as any other and lead to the same number of additional operations. Therefore $|G_t|$ is proportional to the number of operations used in an iteration and can be used to indicate it.

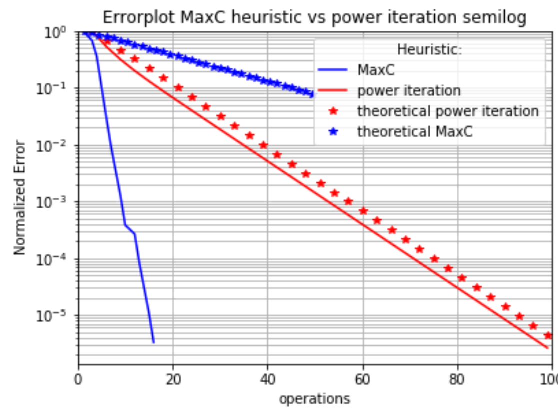
The transition probability matrix, P, used in this example is of a simple three state Markov chain with no self loops:

$$P = \begin{bmatrix} 0 & p & 1-p \\ q & 0 & 1-q \\ r & 1-r & 0 \end{bmatrix} \tag{26}$$

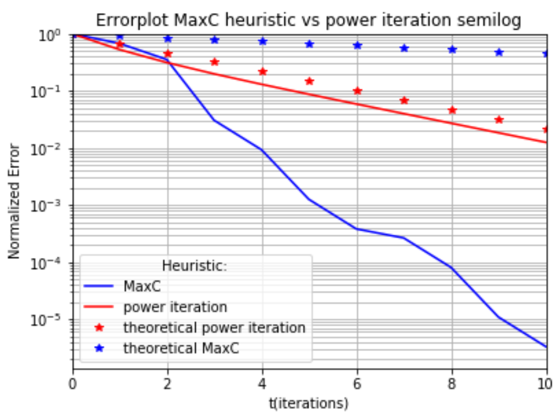
with p, q and r randomly picked from a Uniform Distribution ranging between 0 and 1. For these matrices, where no transition probabilities are likely to be the same and there are no zeros outside of the diagonal, the



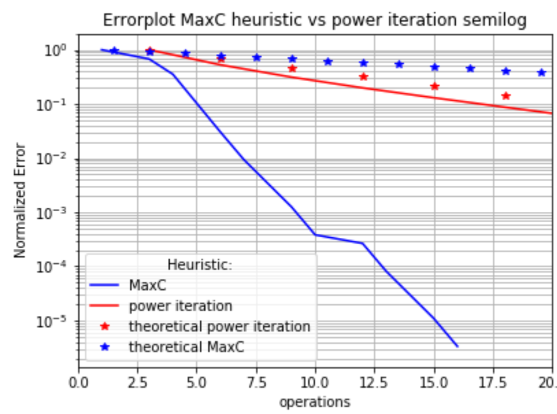
(a) Error progression of the MaxC heuristic, the power iteration method and their theoretical estimates against the number of iteration, t.



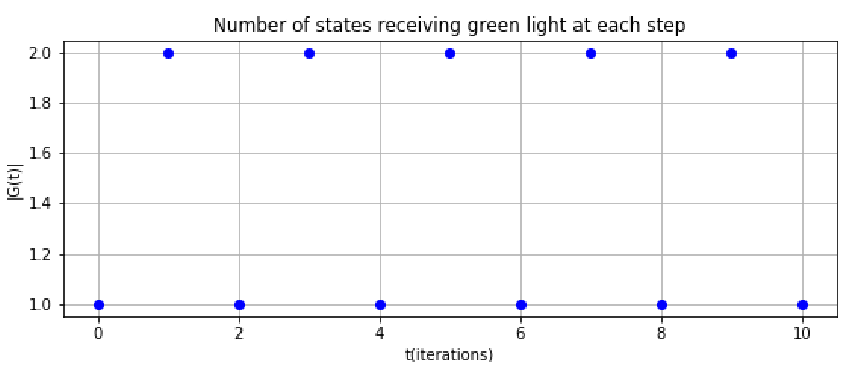
(b) Error progression of the MaxC heuristic, the power iteration method and their theoretical estimates against an indication of the number of operations, $|G_t$.



(c) Zoomed in version of 8a that shows the line of the MaxC heuristic more clearly.



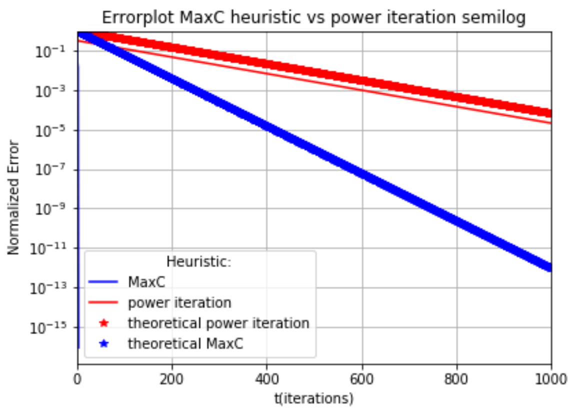
(d) Zoomed in version of 8b that shows the line of the MaxC heuristic more clearly.



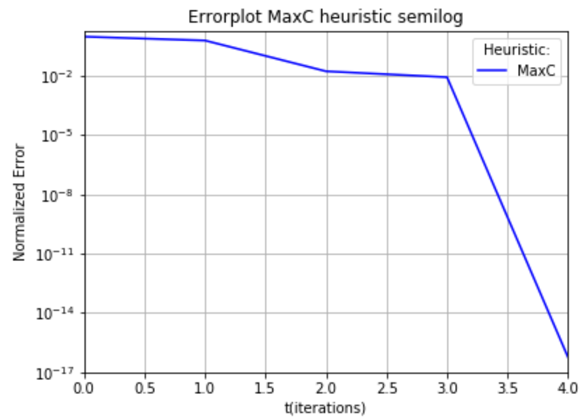
(e) The progression of $|G_t|$

Figure 8: Results for a three state ergodic Markov chain with no self loops and no further zero transition probabilities.

number of states receiving green light from the MaxC heuristic follows a clear pattern, as was also discussed in section 3.3. The theoretical analysis shows that after the initial step green light will be alternately given to 1 or 2 states. This pattern is also observed in the numerical experiments when looking at the progression of $|G_t|$ as can be seen in figure 8e. This knowledge is used for the theoretical estimation from figures 8b and 8d of the number of operations by estimating it as 1.5 operation per iteration. This means that, when looking at the number of operations used in each iteration, the MaxC heuristic is two times faster than the power iteration method. This difference can clearly be seen when comparing figure 8a with figure 8b.



(a) Error progression of the MaxC heuristic, the power iteration method and their theoretical estimates against the number of iterations, t. Please note that the blue line representing the MaxC heuristic is just barely visible next to the left axis of the plot.



(b) Zoomed in version of 9a that shows the behaviour of the MaxC heuristic more clearly.

Figure 9: Results for a three state ergodic Markov chain with no self loops and one additional zero transition probability.

Figure 9 shows another interesting example of how the MaxC heuristic can behave. As discussed in section 3.2, it is possible for the MaxC heuristic to give rise to a non decreasing step followed by convergence after a fixed number of steps. A non decreasing step can only arise if there are additional transition probabilities that are equal to zero. For this purpose the transition probability matrix of equation (26) has been modified by choosing $p = 0$. The behavior of the MaxC heuristic can most clearly be seen in figure 9b which has zoomed in on the first four iterations. At $t = 3$ a steep descend can be observed after barely changing from the step before. When looking at the value of C and the order of green light given by the heuristic at these steps, it can be seen that the first state receiving green light from the MaxC heuristic is the second state, this is at $t = 1$ after the initial step when it has the largest negative amount of cash. It can be confirmed that a non decreasing step was taken directly thereafter, in this case when green light was given to the first and the third state. Followed this step at $t = 2$ is a step giving green light to only 1 state, the third state in this case, that leads to convergence. This clearly illustrates the theory.

Another unique feature of the transition probability matrix used in this example is that $r \approx 0.97$ making it nearly decomposable. It is known that the power iteration method does not perform well in these cases which is supported by the theoretical convergence rate found for this case. Using equation (9) it can be found that $|\lambda_2| \approx 0.99$. This is worse than the theoretical bound found for the MaxC heuristic as can also be seen in figure 9a. This makes this a case in which the MaxC heuristic can be expected to perform better than the power iteration method even if the guaranteed convergence after a non decreasing step does not occur. This can be also be seen from the blue line in figure 9 that descents much faster than the red line, even before the non decreasing step.

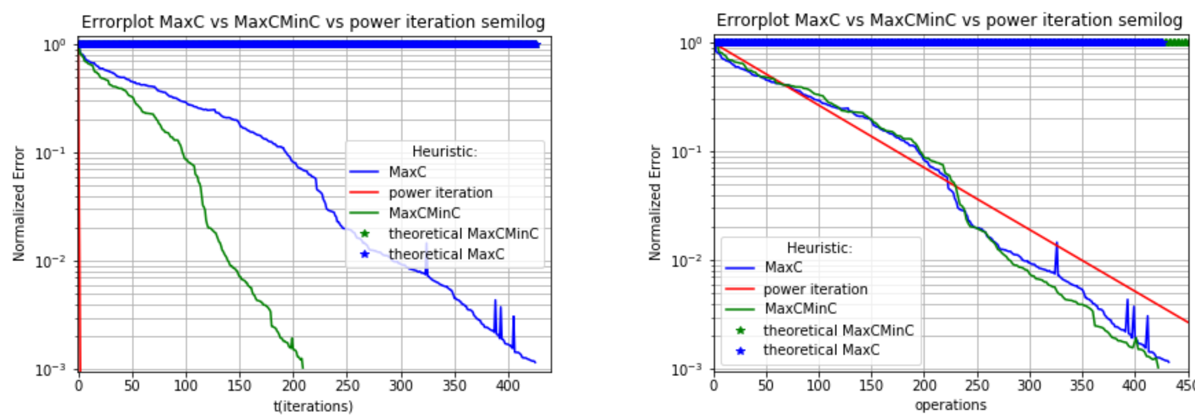
This completes the numerical analysis on three state ergodic Markov chains with no self loops. The next section discusses the numerical analysis for several Markov chains that are random walks on complete graphs.

5.2 Complete graphs

Where the previous section only discussed the MaxC heuristic, this one will also include the MaxCMinC heuristic. This is done for two different types of complete graphs, each discussed in its own subsection. In all cases the normalized error, $\frac{\|\hat{\pi_t} - \hat{\pi_{t-1}}\|_1}{\|\hat{\pi_2} - \hat{\pi_1}\|_1}$, is set out both against t, the number of iterations, and against an indication for the number of operations. As we only consider complete graphs, there is no difference in degree depending on what state we look at. Therefore, as in the previous section, the number of states receiving green light at each iteration, $|G_t|$, is used as an indication for the number of operations.

5.2.1 Uniformly Distributed transition probabilities

The first model used as an example is a very simple one where each transition probability is taken from a Uniform distribution ranging between 0 and 1. After this the rows are normalized to make the transition probabilities sum to 1. This can be done for any number of states, which was set to 250 for the example used in figure 10.



(a) Error progression of the MaxC heuristic, the MaxCMinC heuristic, their theoretical estimates and the power iteration method against the number of iterations, t.

(b) Error progression of the MaxC heuristic, the MaxCMinC heuristic, their theoretical estimates and the power iteration method against an indication of the number of operations, $|G_t|$.

Figure 10: Results for a complete graph with $n = 250$ and uniformly distributed transition probabilities. The green stars are mostly hidden by the blue stars.

Like in the previous section, the left figure, 10a plots the normalized error against t, the number of iterations, and the right figure, 10b, plots it against the indication for the number of operations, the number of states receiving green light.

One of the first things that can be seen in these figures is that the theoretical value found for the convergence rate is indeed a very rough bound. For both the MaxC heuristic and the MaxCMinC heuristic this value is very close to 1. This causes both theoretical lines to almost completely overlap and appear practically horizontal in the figure.

In figure 10a it can be seen that the MaxCMinC heuristic converges faster than the MaxC heuristic while at the same time performing more stably. Both are, however, clearly outperformed by the power iteration method, the line that can just barely be observed on the very left side of the figure.

If the number of operations is considered, as can be seen in figure 10b, the large gap between the heuristics disappears. The MaxC heuristic and the MaxCMinC heuristic use approximately the same number of operations to reach convergence. With the later doing so in less iterations. Both of them use less operations in total than the power iteration method, making them more efficient.

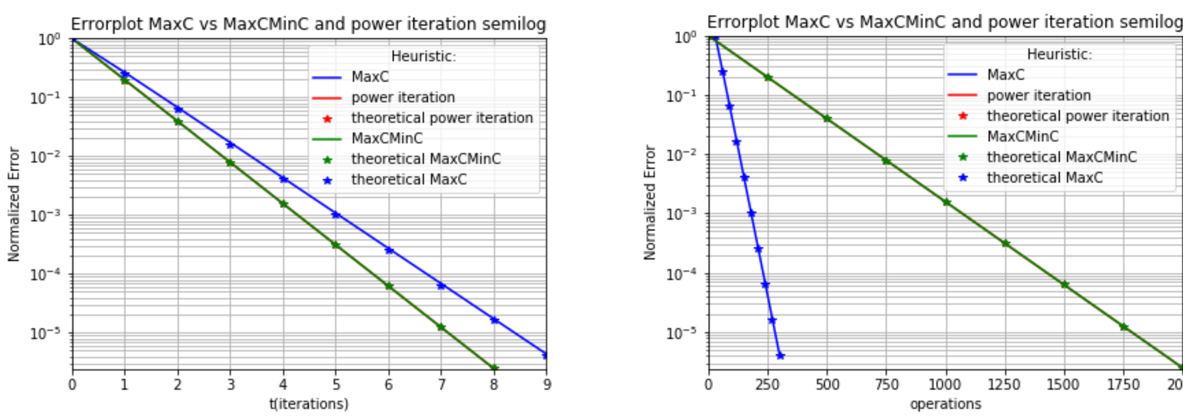
As was hypothesized in chapter 4, the MaxC heuristic almost always gives green light to only one state. In the same way, the MaxCMinC heuristic almost always gives green light to two states. Choosing $m = 1$ in equations (18) and (20), which we use to produce the theoretical lines of the MaxC heuristic and the MaxCMinC heuristic respectively, is therefore not only the worst case scenario but also the most common one.

In the next section it can be seen that this depends on the structure of the graph, as also discussed in theory in chapter 4

5.2.2 Mean-Field Stochastic Block Model

The other model we use as an example is the Mean-Field SBM, as described in section 4.2.

The first example is a graph that only has 2 blocks and can be found in figure 11 along with further parameter details. One of the first points to note is that red and the green lines completely overlap. This is not surprising, as we know that the MaxCMinC heuristic gives green light to both blocks, which is effectively the same as the power iteration method.



(a) Error progression of the MaxC heuristic, the MaxCMinC heuristic, the power iteration method and their theoretical estimates against the number of iterations, t.

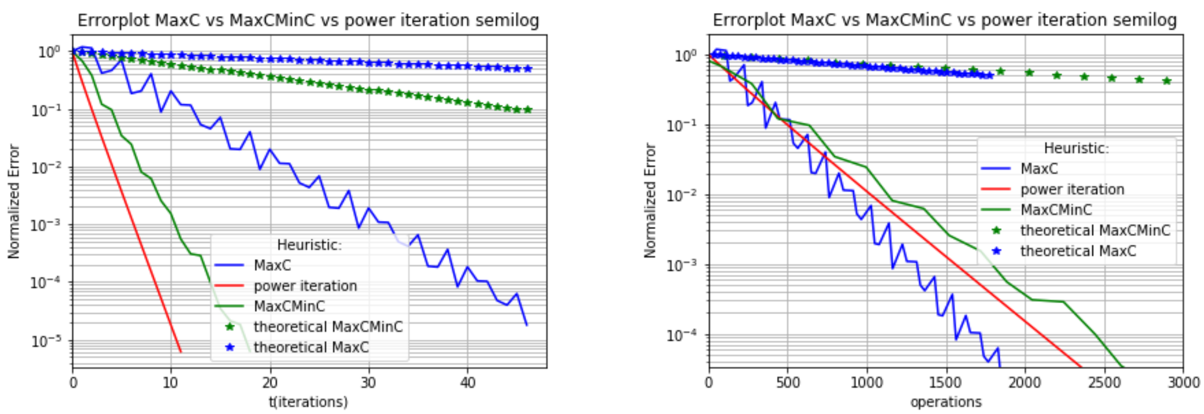
(b) Error progression of the MaxC heuristic, the MaxCMinC heuristic, the power iteration method and their theoretical estimates against an indication of the number of operations, $|G_t|$.

Figure 11: Results for a Mean-Field SBM with 2 blocks, $N_1 = 220$ and $N_2 = 30$, $p = 0.90$ and $q = 0.37$. The red line and stars completely overlap with the green line and stars.

The second point to note is that the theoretical line nicely follows the numeric line. This once again indicates that the theoretical values for the convergence rate are exact for this example and not estimations. This goes for both in terms of t, the number of iteration, as in terms of the number of operations. It is guaranteed that, on one hand, the MaxCMinC heuristic will always give green light to all states like the power iteration method. On the other hand, the MaxC heuristic is guaranteed to always give green light to all states of the smallest block. This can also nicely be observed in the numerical example.

This figure therefore clearly illustrates what we already knew from theory; The power iteration method, and therefore the MaxCMinC heuristic, is faster than the MaxC heuristic in terms of the number of iteration. If the number of operations are taken into account, however, the MaxC heuristic far outperforms the others.

The second example is one that has 4 block and can be found in figure 12 along with further details about the parameters. Firstly note that the theoretical line for this example could not be made as precise as in the previous one. On the other had, it still gives a better image than the theoretical lines available in figure 10, where all transition probabilities where chosen from a Uniform distribution.



(a) Error progression of the MaxC heuristic, the MaxCMinC heuristic, their theoretical estimates and the power iteration method against the number of iterations, t.

(b) Error progression of the MaxC heuristic, the MaxCMinC heuristic, their theoretical estimates and the power iteration method against an indication of the number of operations, $|G_t|$.

Figure 12: Mean-Field SBM with 4 blocks, $N_1 = 137$, $N_2 = 62$, $N_3 = 28$ and $N_4 = 23$, $p = 0.64$ and $q = 0.25$.

It can also be seen that the MaxC heuristic gives a slightly unstable line and takes more iteration to reach convergence than the power iteration method and the MaxCMinC heuristic. On the other hand, it can again be seen that it takes fewer operations than both the others.

Another interesting result is how the heuristics give green light in this example. The MaxC heuristic cyclically gives green light to the three smallest blocks starting with the smallest one. The MaxCMinC heuristic, on the other hand, gives green light to the same block that is given green light by the MaxC heuristic in addition to the largest block. This is not necessarily surprising as, in this case, the largest block contains all the positive cash while the negative cash is distributed over the three smaller blocks. This knowledge was also used to give an better estimate to the number of operations used for the theoretical line.

6 Conclusions and future research

The goal of this research was to obtain conditions under which the MaxC heuristic converges and to obtain its convergence rate. In chapters 3 and 4 these results have been analytically derived. It has been proven that the MaxC heuristic converges exponentially both in the constricted case of a three state ergodic Markov chain with no self loops and in the more general case of Markov chains that are a random walk on complete graphs with any number of states.

Finding an exact convergence rate is hard. Therefore, estimates have been found, see equations (10), (17) and (18), along with the areas in which they might be lagging. Also, it has been shown that these estimates can be made more precise and in some cases even be made exact by imposing further restrictions on the Markov chain. See equations (24) and (23) respectively, that where obtained by using the Mean-Field SBM as the model for the graph.

Another way in which the results for the MaxC heuristic have been generalized is by extending the heuristic itself. The results gave rise to a new heuristic, the MaxCMinC heuristic. The results of the MaxC heuristic have been extended for this new heuristic. This proves exponential convergence if the graph that the Markov chain is a random walk on is a complete graph for the MaxCMinC heuristic.

For this heuristic, as well, estimates of the convergence rate have been found, see equation (20). This along with where these rates might be lagging and they can be made slightly more precise by giving a better estimate for m when considering the Mean-Field SBM.

The results for the MaxCMinC heuristic have, in turn, been used to also proof exponential convergence for the Theta heuristic when certain bounds are imposed on the value of θ_t.

These analytical results have been tested in numerical experiments. The heuristics have also been compared to each other and to the well known power iteration method for reference. In chapter 5 these experiments have been discussed. In summary, these experiments have shown the limitation of the estimates made for the convergence rate. At the same time they also show that the potential of the MaxC heuristic, as well as the MaxCMinC heuristic, is greater than the theoretical estimates can show.

With this the goal of this research has been met, but many directions for future research remain open. On one hand it would be interesting to look at more different structures of the graph, such as graphs that have a cyclic structure or sparse graphs with the addition of the reset condition of PageRank. It would also be interesting to look if the estimate of the convergence rate can be made more precise in these cases as well and if the addition of other conditions could be used to give better estimates in general. On the other hand, it could also be interesting if, with possible addition of other conditions, the requirement of the graphs being complete can be relaxed.

In the original work, [2], even more heuristics have been proposed. These, as well as some still to be discovered heuristics, would also benefit from a thorough analysis. It would be interesting to know the strong as well as the weak points of all these different heuristics and clearly map out when each of these would be the most useful.

On a completely different note, it would also be very interesting to research the effects of the initial step on both the convergence of the algorithm, as well as on the progression of the History, H, over time. Some of the results benefited from assuming the use of an uniform initial step. Furthermore, from additional experiments concerning the three state Markov chains the possible relation between the initial step and possibility of H becoming equal or converging to 0 could be devised. This also gave rise to the hypothesis that the History won't converge to 0 when an uniform initial step is used. This could prove useful to research, as the History becoming 0 is still an not understood breaking point for the algorithm.

Acknowledgements

I would first like to thank my main supervisor, Nelly Litvak, for her help, feedback and motivation during my assignment. Talking to her always gave me motivation to continue with the project even if thinks weren't going as planned. I would also like to thank Konstantin Avrachenkov for his help and feedback when my main supervisor wasn't available and thereafter. And also lots of appreciation for Matthias Schlottbom en Werner Scheinhardt for joining the graduation assessment committee. I would lastly also like to thank my group from the EEMCS Graduation Support Group for their support and help in getting out of bed and start working on my thesis every morning.

References

[1] Lawrence Page and Sergey Brin. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 4 1998.

[2] Konstantin Avrachenkov, Patrick Brown, and Nelly Litvak. Red Light Green Light Method for Solving Large Markov Chains. *ArXiv*, 2008.02710, 8 2020.

[3] Frank McSherry. A Uniform Approach to Accelerated PageRank Computation. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, pages 575–582, New York, NY, USA, 2005. Association for Computing Machinery.

[4] Atsushi Suzuki and Hideaki Ishii. Distributed Randomized Algorithms for PageRank Based on a Novel Interpretation. In *Proceedings of the American Control Conference*, volume 2018-June, pages 472–477. Institute of Electrical and Electronics Engineers Inc., 8 2018.

[5] Atsushi Suzuki and Hideaki Ishii. PageRank Computation via Web Aggregation in Distributed Randomized Algorithms. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 1856–1861, 2019.

[6] Dohy Hong. Optimized on-line computation of PageRank algorithm. *ArXiv*, 1202.6158, 2 2012.

[7] Julie Nutini, Mark Schmidt, Issam H. Laradji, Michael Friedlander, and Hoyt Koepke. Coordinate Descent Converges Faster with the Gauss-Southwell Rule Than Random Selection. *ArXiv*, 1506.00552, 6 2015.

[8] I S Dhillon, P Ravikumar, and A Tewari. Nearest neighbor based greedy coordinate descent. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011, NIPS 2011*, 2011.

[9] Zeyuan Allen-Zhu, Zheng Qu, Peter Richtárik, and Yang Yuan. Even Faster Accelerated Coordinate Descent Using Non-Uniform Sampling. *ArXiv*, 1512.09103, 12 2015.

[10] Serge Abiteboul, Mihai Preda, and Gregory Cobena. Adaptive On-Line Page Importance Computation. In *Proceedings of the 12th International Conference on World Wide Web*, WWW '03, pages 280–290, New York, NY, USA, 2003. Association for Computing Machinery.

[11] Nelly Litvak and Philippe Robert. Analysis of an on-line algorithm for solving large Markov chains. In *Proceedings of the 3rd International Conference on Performance Evaluation Methodologies and Tools*, page 19. ICST, 2008.

[12] Nelly Litvak and Philippe Robert. A scaling analysis of a cat and mouse Markov Chain. *Annals of Applied Probability*, 22(2):792–826, 4 2012.

[13] Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 6 1983.