ESAIM: PS 24 (2020) 138–147
https://doi.org/10.1051/ps/2019024

ESAIM: Probability and Statistics
www.esaim-ps.org

A NOTE ON PERFECT SIMULATION FOR EXPONENTIAL RANDOM GRAPH MODELS

Andressa Cerqueira[1,*], Aurélien Garivier[2]
and Florencia Leonardi[3]

Abstract. In this paper, we propose a perfect simulation algorithm for the Exponential Random Graph Model, based on the Coupling from the past method of Propp and Wilson (1996). We use a Glauber dynamics to construct the Markov Chain and we prove the monotonicity of the ERGM for a subset of the parametric space. We also obtain an upper bound on the running time of the algorithm that depends on the mixing time of the Markov chain.

Mathematics Subject Classification. 60K35,60J22.

Received April 27, 2018. Accepted November 6, 2019.

1. Introduction

In recent years there has been an increasing interest in the study of probabilistic models defined on graphs. One of the main reasons for this interest is the flexibility of these models, making them suitable for the description of real datasets, for instance in social networks [7] and neural networks [3]. The interactions in a real network can be represented through a graph, where the interacting objects are represented by the vertices of the graph and the interactions between these objects are identified with the edges of the graph.

In this context, the Exponential Random Graph Model (ERGM) has been vastly explored to model these type of interactions in real networks [9]. Inspired on the Gibbs distributions of Statistical Physics, one of the reasons of the popularity of ERGM can be attributed to the specific definition of the probability distribution that takes into account appealing properties of the graphs such as number of edges, triangles, stars, or any other subgraph count.

In terms of statistical inference, learning the parameters of a ERGM is not a simple task. Classical inferential methods such as maximum likelihood [5] or pseudolikelihood [11] have been proposed, but exact calculation of the estimators is computationally infeasible except for small graphs. In the general case, estimation is made by means of a Markov Chain Monte Carlo (MCMC) procedure [10]. It consists in building a Markov chain whose stationary distribution is the target law. The idea in this approach is to simulate a "long enough" random path,

Keywords and phrases: Exponential Random Graph, perfect simulation, coupling from the past, MCMC, Glauber dynamics.

[1] Statistics Department, Universidade Federal de São Carlos, São Carlos, Brazil.
[2] Unité de Mathématiques Pures et Appliquées, Laboratoire de l'Informatique du Parallélisme, Ecole Normale Supérieure de Lyon, Université de Lyon, Lyon, France.
[3] Statistics Department, Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, Brazil.

[*] Corresponding author: acerqueira@ufscar.br

Article published by EDP Sciences

so as to get an approximate sample. But this procedure suffers from a "burn-in" time that is difficult to estimate in practice and can be exponentially long, as shown in [1].

In order to get a sample of the stationary distribution of the chain, an alternative approach introduced by [8] is called *Coupling From The Past* (CFTP). This method yields a value distributed under the *exact* target distribution, and is thus called perfect (also known as exact) simulation. Like MCMC, it does not require computing explicitly the normalising constant; but it does not require any burn-in phase either, nor any detection of convergence to the target distribution.

In this work we present a perfect simulation algorithm for the ERGM. In our approach, we adapt a version of the CFTP algorithm, which was originally developed for spin systems [8] to construct a perfect simulation algorithm using a Markov chain based on the Glauber dynamics. For a specific family of distributions based on subgraph counts we prove that they satisfy the monotonicity property and then the CFTP algorithm is computationally efficient to generate a sample from the target distribution. Our method is related to the recent work [2] where a perfect sampling algorithm was proposed for not necessarily monotone models by constructing a pair of stochastically coupled bounding processes. However, this method is only computationally efficient when the subgraph counts are monotone increasing in edge addition. In terms of theoretical results, it is proved that the bounding algorithm stops with probability going to 1; however, there is no clue about the expected running time of the algorithm. The work presented here takes a slightly different point of view and was apparently derived independently at the same time. Our main contribution is that the expected running time of the perfect simulation algorithm is at most as large (up to logarithmic terms) than the mixing time of the Markov chain constructed using the Glauber dynamics. As a consequence, we argue CFTP is superior to MCMC for the simulation of the ERGM.

2. Definitions

2.1. Exponential random graph model

A graph is a pair (V, E), where $V \subset \mathbb{N}$ is a finite set of N vertices and $E \subset V \times V$ is a set of edges that connect pairs of vertices. In this work we consider undirected graphs; *i.e.*, graphs for which the set E satisfies the following properties:

- For all $i \in V$, $(i,i) \notin E$.
- If $(i,j) \in E$, then $(j,i) \in E$.

Let $\mathcal{G}_N$ be the collection of all undirected graphs with set of vertices V. A graph $(V,E) \in \mathcal{G}_N$ can be identified with a symmetric binary matrix $\mathbf{x} \in \{0,1\}^{V\times V}$, called adjacency matrix, such that $x(i,j) = 1$ if and only if $(i,j) \in E$. Thus from now on a graph will be denoted by its adjacency matrix.

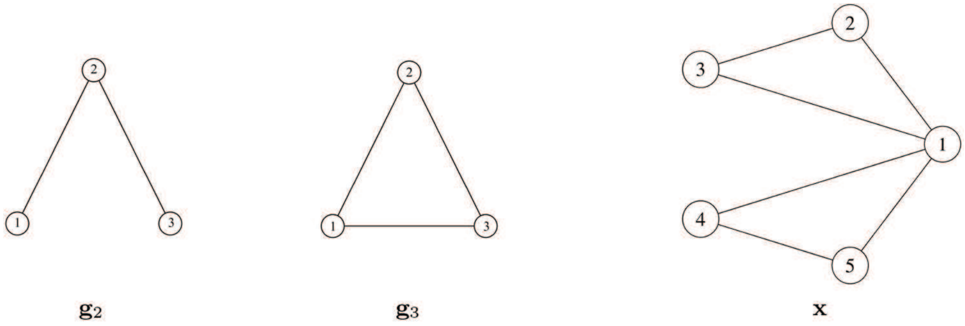
The *empty graph* $\mathbf{x}^{(0)}$ is given by $x^{(0)}(i,j) = 0$, for all $i,j \in V$, and the the *complete graph* $\mathbf{x}^{(1)}$ is given by $x^{(1)}(i,j) = 1$, for all $i,j \in V$.

We write $\mathbf{x}_{-ij}$ to represent the set of values $\{\mathbf{x}(l,k)\colon (i,j) \neq (l,k)\}$. For a given graph $\mathbf{x}$ and a set of vertices $U \subseteq V$, we denote by $\mathbf{x}(U)$ the subgraph *induced* by U, that is $\mathbf{x}(U)$ is a graph with set of vertices U and such that $\mathbf{x}(U)(i,j) = \mathbf{x}(i,j)$ for all $i,j \in U$.

In the ERGM model, the probability of a given graph in $\mathcal{G}_N$ is defined by its structure: its number of edges, of triangles, or any other subgraph count. To define subgraphs counts in a graph, let V_m be the set of all possible arrangements (ordering) of m distinct vertices of V, that is

$$V_m = \{\, (k_1, \ldots, k_m) : k_1, \cdots, k_m \text{ are distinct vertices of } V\}.$$

For $v \in V_m$, define $\mathbf{x}(v)$ as the subgraph of $\mathbf{x}$ induced by the set of elements appearing in v. For a graph $\mathbf{g}$ with vertices $\{1,\ldots,m\}$ and for $v \in V_m$, we say $\mathbf{x}(v)$ *contains* $\mathbf{g}$ (and we write $\mathbf{x}(v) \succeq \mathbf{g}$) if $\mathbf{g}(i,j) = 1$ implies $\mathbf{x}(v)(k_i,k_j) = 1$, for all $1 \leq i < j \leq m$. Then, for $\mathbf{x} \in \mathcal{G}_N$ and $\mathbf{g} \in \mathcal{G}_m$, $m \leq N$, we define the number of



$\mathbf{g}_2$ $\mathbf{g}_3$ $\mathbf{x}$

FIGURE 1. Examples of subgraphs $\mathbf{g}_2$ (2-star) and $\mathbf{g}_3$ (triangle) to count occurrences in a graph $\mathbf{x}$. In this example, $N_{\mathbf{g}_2}(\mathbf{x}) = 20$ and $N_{\mathbf{g}_3}(\mathbf{x}) = 12$.

subgraphs $\mathbf{g}$ in $\mathbf{x}$ by the counter

$$N_{\mathbf{g}}(\mathbf{x}) = \sum_{v \in V_m} \mathbb{1}\{\mathbf{x}(v) \succeq \mathbf{g}\}. \tag{2.1}$$

To illustrate the definitions above, consider a graph $\mathbf{x}$ with vertex set $V = \{1, 2, 3, 4, 5\}$ and the subgraphs $\mathbf{g}_2$ (2-star) and $\mathbf{g}_3$ (triangle) given in Figure 1. In this case V_3 is given by the set of all arrangements of three distinct vertices of V. Since in $\mathbf{x}$ there are two triangles formed with vertices $\{1, 2, 3\}$ and $\{1, 4, 5\}$, then $N_{\mathbf{g}_3}(\mathbf{x}) = 12$. In the same way, for the 2-star shape $\mathbf{g}_2$ we have that $N_{\mathbf{g}_2}(\mathbf{x}) = 20$.

Let $\mathbf{g}_1, \ldots, \mathbf{g}_s$ be fixed graphs, where $\mathbf{g}_i$ has m_i vertices, $m_i \le N$ and $\boldsymbol{\beta} \in \mathbb{R}^s$ a vector of parameters. By convention we set $\mathbf{g}_1$ as the graph with only two vertices and one edge. Following [1, 4] we define the probability of graph $\mathbf{x}$ by

$$p_N(\mathbf{x}|\boldsymbol{\beta}) = \frac{1}{Z_N(\boldsymbol{\beta})} \exp\left(\sum_{i=1}^{s} \beta_i \frac{N_{\mathbf{g}_i}(\mathbf{x})}{N^{m_i - 2}}\right), \tag{2.2}$$

where $Z_N(\boldsymbol{\beta})$ is a normalising constant, usually known as partition function.

For simplicity of notation, and without loss of generality, from now on we will assume that the vertex set is given by $V = \{1, \ldots, N\}$. Observe that the set $\mathcal{G}_N$ is equipped with a partial order given by

$$\mathbf{x} \preceq \mathbf{y} \quad \text{if and only if} \quad \mathbf{x}(i, j) = 1 \quad \text{implies} \quad \mathbf{y}(i, j) = 1. \tag{2.3}$$

Moreover, this partial ordering has a maximal element $\mathbf{x}^{(1)}$ (the complete graph) and a minimal element $\mathbf{x}^{(0)}$ (the empty graph). Considering this partial order, we say $p_N(\cdot\,|\boldsymbol{\beta})$ is *monotone* if the conditional distribution of $\mathbf{x}(i, j) = 1$ given $\mathbf{x}_{-ij}$ is a monotone increasing function. That is, $p_N(\cdot\,|\boldsymbol{\beta})$ is monotone if, and only if $\mathbf{x} \preceq \mathbf{y}$ implies

$$p_N(\mathbf{x}(i, j) = 1|\boldsymbol{\beta}, \mathbf{x}_{-ij}) \;\le\; p_N(\mathbf{y}(i, j) = 1|\boldsymbol{\beta}, \mathbf{y}_{-ij}). \tag{2.4}$$

Let $\mathbf{E} = \{(i, j) : i, j \in V \text{ and } i < j\}$ be the set of possible edges in a graph with set of vertices V. For any graph $\mathbf{x} \in \mathcal{G}_N$, a pair of vertices $(i, j) \in \mathbf{E}$ and $a \in \{0, 1\}$ we define the modified graph $\mathbf{x}_{ij}^a \in \mathcal{G}_N$ given by

$$\mathbf{x}_{ij}^a(k, l) = \begin{cases} x(k, l)\,, & \text{if } (k, l) \ne (i, j)\,; \\ a\,, & \text{if } (k, l) = (i, j)\,. \end{cases}$$

Then, Inequality (2.4) is equivalent to

$$\frac{p_N(\mathbf{x}^1_{ij}\,|\boldsymbol{\beta})}{p_N(\mathbf{x}^0_{ij}\,|\boldsymbol{\beta})} \leq \frac{p_N(\mathbf{y}^1_{ij}\,|\boldsymbol{\beta})}{p_N(\mathbf{y}^0_{ij}\,|\boldsymbol{\beta})}. \tag{2.5}$$

Moreover, if this inequality holds for all $\mathbf{x} \preceq \mathbf{y}$ then the distribution is monotone. Monotonicity plays a very important role in the development of efficient perfect simulation algorithms. The following proposition gives a sufficient condition on the parameter vector $\boldsymbol{\beta}$ under which the corresponding ERGM distribution is monotone.

Proposition 2.1. *Consider the ERGM given by (2.2). If $\beta_i \geq 0$ for all $i \geq 2$ then the distribution $p_N(\cdot\,;\boldsymbol{\beta})$ is monotone.*

The proof of Proposition 2.1 is given in Section 6.

3. Coupling from the past

In this section we recall the construction of a Markov chain with stationary distribution $p_N(\cdot\,;\boldsymbol{\beta})$ using a local update algorithm called *Glauber dynamics*. This construction and the monotonicity property given by Proposition 2.1 are the keys to develop a perfect simulation algorithms for the ERGM.

3.1. Glauber dynamics

In the Glauber dynamics the probability of a transition from graph $\mathbf{x}$ to graph $\mathbf{x}^a_{ij}$ is given by

$$p(\mathbf{x}_{ij} = a|\boldsymbol{\beta}, \mathbf{x}_{-ij}) = \frac{p_N(\mathbf{x}^a_{ij}|\boldsymbol{\beta})}{p_N(\mathbf{x}^0_{ij}|\boldsymbol{\beta}) + p_N(\mathbf{x}^1_{ij}|\boldsymbol{\beta})} \tag{3.1}$$

and any other (*non-local*) transition is forbidden.

For the ERGM, this can be written

$$p(\mathbf{x}_{ij} = 1|\boldsymbol{\beta}, \mathbf{x}_{-ij}) = \frac{1}{1 + \exp\left\{-2\beta_1 + \sum_{k=2}^{s} \frac{\beta_k}{N^{m_k - 2}} \left(N_{\mathbf{g}_k}(\mathbf{x}^0_{ij}) - N_{\mathbf{g}_k}(\mathbf{x}^1_{ij})\right)\right\}} \tag{3.2}$$

and

$$p(\mathbf{x}_{ij} = 0|\boldsymbol{\beta}, \mathbf{x}_{-ij}) = 1 - p(\mathbf{x}_{ij} = 1|\boldsymbol{\beta}, \mathbf{x}_{-ij}).$$

It can be seen easily that the Markov chain with the Glauber dynamics (3.2) has an invariant distribution equal to $p_N(\cdot\,;\boldsymbol{\beta})$. It can be implemented with the help of the local update function $\phi : \mathcal{G}_N \times \mathbf{E} \times [0,1] \to \mathcal{G}_N$ defined as

$$\phi(\mathbf{x}, (i,j), u) = \begin{cases} \mathbf{x}^0_{ij}, & \text{if } u \leq p(\mathbf{x}_{ij} = 0|\boldsymbol{\beta}, \mathbf{x}_{-ij}); \\ \mathbf{x}^1_{ij}, & \text{otherwise.} \end{cases} \tag{3.3}$$

A transition in the Markov chain from graph $\mathbf{x}$ to graph $\mathbf{y}$ is obtained by the random function

$$\mathbf{y} \;=\; \phi(\mathbf{x}, \mathbf{e}, \mathbf{u})\,, \qquad \text{with } \mathbf{e} \sim \text{Uniform}(\mathbf{E}),\ \mathbf{u} \sim \text{Uniform}(0,1).$$

Define for $m \leq n$ the random vector $\mathbf{e}^n_m = (\mathrm{e}_m, \mathrm{e}_{m+1}, \ldots, \mathrm{e}_n)$ and $\mathbf{u}^n_m = (u_m, u_{m+1}, \ldots, u_n)$, with $\mathrm{e}_k \in \mathbf{E}$ and $u_k \in [0,1]$. For random vectors $\mathbf{e}^{-1}_{-n} = (\mathrm{e}_{-n}, \mathrm{e}_{-n+1}, \ldots, \mathrm{e}_{-1})$ and $\mathbf{u}^{-1}_{-n} = (u_{-n}, u_{-n+1},\, \ldots, u_{-1})$, with $\mathrm{e}_{-k} \in \mathbf{E}$

and $u_{-k} \in [0,1]$, we define the random map F^0_{-n}

$$F^0_{-1}(\mathbf{x}, \mathbf{e}^{-1}_{-1}, \mathbf{u}^{-1}_{-1}) = \phi(\mathbf{x}, e_{-1}, u_{-1})\,, \text{and}$$
$$F^0_{-n}(\mathbf{x}, \mathbf{e}^{-1}_{-n}, \mathbf{u}^{-1}_{-n}) = F^0_{-n+1}(F^0_{-1}(\mathbf{x}, \mathbf{e}^{-n}_{-n}, \mathbf{u}^{-n}_{-n}), \mathbf{e}^{-1}_{-n+1}, \mathbf{u}^{-1}_{-n+1})\,, \quad \text{for } n \geq 2\,.$$

The CFTP protocol of [8] relies on the following elementary observation. If, for some time $-n$ and for some uniformly distributed $\mathbf{e}^{-1}_{-n} = (\mathrm{e}_{-n} \ldots, \mathrm{e}_{-1})$ and $\mathbf{u}^{-1}_{-n} = (u_{-n}, \ldots, u_{-1})$, the mapping $F^0_{-n}(\cdot, \mathbf{e}^{-1}_{-n}, \mathbf{u}^{-1}_{-n})$ is *constant*, that is if it takes the same value at time 0 on all possible graphs $\mathbf{x}$, then this constant value is readily seen to be a sample from the invariant measure of the Markov chain. Moreover, if F_{-n} is constant for some positive n, then it is also constant for all larger values of n. Thus, it suffices to find a value of n large enough so as to obtain a constant map, and to return the constant value of this map. For example, one may try some arbitrary value n_1, check if the map $F^0_{-n_1}(\cdot, \mathbf{e}^{-1}_{-n_1}, \mathbf{u}^{-1}_{-n_1})$ is constant, try $n_2 = 2n_1$ otherwise, and so on...

Since the state space has the huge size $|\mathcal{G}_N| = 2^{\frac{N(N-1)}{2}}$, it would be computationally intractable to compute the value of $F^0_{-n}(\mathbf{x}, \mathbf{e}^{-1}_{-n}, \mathbf{u}^{-1}_{-n})$ for all $\mathbf{x} \in \mathcal{G}_N$ in order to determine if they coincide.

This is where the monotonicity property helps: it is sufficient to inspect the maximal and minimal elements of $\mathcal{G}_N$. If they both yield the same value, then all other initial states will also coincide with them, and the mapping will be constant.

To apply the monotone CFTP algorithm, we use the partial order defined by (2.3) and the extremal elements $\mathbf{x}^{(0)}$ and $\mathbf{x}^{(1)}$. The resulting procedure, which simulates the ERGM using the CFTP protocol, is described in Algorithm 1.

Algorithm 1 CFTP for ERGM.

```
Input: $\mathbf{E}$, $\phi$
Output: $\mathbf{x}$
$upper(i,j) \leftarrow 1$, for all $(i,j) \in \mathbf{E}$
$lower(i,j) \leftarrow 0$, for all $(i,j) \in \mathbf{E}$
$n \leftarrow 0$
while upper $\neq$ lower do
    $n \leftarrow n+1$
    $upper(i,j) \leftarrow 1$, for all $(i,j) \in \mathbf{E}$
    $lower(i,j) \leftarrow 0$, for all $(i,j) \in \mathbf{E}$
    Choose a pair of vertices $\mathrm{e}_{-n}$ randomly on $\mathbf{E}$
    Simulate $u_{-n}$ with distribution uniform on $[0,1]$
    upper $\leftarrow F^0_{-n}(upper, \mathbf{e}^{-1}_{-n}, \mathbf{u}^{-1}_{-n})$
    lower $\leftarrow F^0_{-n}(lower, \mathbf{e}^{-1}_{-n}, \mathbf{u}^{-1}_{-n})$
end while
$\mathbf{x} \leftarrow upper$
Return: $n$, $\mathbf{x}$
```

Let

$$T^{\text{stop}}_N = \min\{n > 0 : F^0_{-n}(\mathbf{x}^{(0)}, \mathbf{e}^{-1}_{-n}, \mathbf{u}^{-1}_{-n}) = F^0_{-n}(\mathbf{x}^{(1)}, \mathbf{e}^{-1}_{-n}, \mathbf{u}^{-1}_{-n})\} \tag{3.4}$$

be the stopping time of Algorithm 1. Proposition 3.1 below guarantees that the law of the output of Algorithm 1 is the target distribution $p_N(\cdot; \boldsymbol{\beta})$.

Proposition 3.1. *For T^{stop}_N defined in (3.4) we have $\mathbb{P}(T^{stop}_N < \infty) = 1$ and then the graph $\mathbf{x}$ returned by Algorithm 1 has law $p_N(\cdot; \boldsymbol{\beta})$.*

4. Convergence speed: why CFTP is better

By the construction of the perfect simulation algorithm it is expected that the stopping time T_N^{stop} is related to the mixing time of the chain. In Theorem 4.1 below we provide an upper bound on this quantity.

Given vectors $\mathbf{e}_1^n = (\mathrm{e}_1, \mathrm{e}_2, \dots, \mathrm{e}_n)$ and $\mathbf{u}_1^n = (u_1, u_2 \dots, u_n)$ define the forward random map $F_0^n(\mathbf{x}, \mathbf{e}_1^n, \mathbf{u}_1^n)$ as

$$\begin{aligned} &F_0^1(\mathbf{x}, \mathbf{e}_1^1, \mathbf{u}_1^1) = \phi(\mathbf{x}, e_1, u_1)\,; \\ &F_0^n(\mathbf{x}, \mathbf{e}_1^n, \mathbf{u}_1^n) = F_0^1(F_0^{n-1}(\mathbf{x}, \mathbf{e}_1^{n-1}, \mathbf{u}_1^{n-1}), \mathbf{e}_n^n, \mathbf{u}_n^n)\,, \quad \text{for } n \geq 2. \end{aligned} \tag{4.1}$$

For $\mathbf{x} \in \mathcal{G}_N$, let the Markov chain $\{Y_n^{\mathbf{x}}\}_{n\in\mathbb{N}}$ be given by

$$\begin{aligned} &Y_0^{\mathbf{x}} = \mathbf{x}\,; \\ &Y_n^{\mathbf{x}} = F_0^n(\mathbf{x}, \mathbf{e}_1^n, \mathbf{u}_1^n)\,, \quad n \geq 1 \end{aligned} \tag{4.2}$$

and denote by $\mathrm{p}_n^{\mathbf{x}}$ the distribution of this Markov chain at time n. Now define

$$\bar{d}(n) = \max_{\mathbf{x},\mathbf{z}\in\mathcal{G}_N} \parallel \mathrm{p}_n^{\mathbf{x}} - \mathrm{p}_n^{\mathbf{z}} \parallel_{\text{TV}},$$

where $\|\cdot\|_{\text{TV}}$ stands for the total variation distance.

Let T_N^{mix} be given by

$$T_N^{\text{mix}} = \min\left\{ n > 0 : \bar{d}(n) \leq \frac{1}{e} \right\}. \tag{4.3}$$

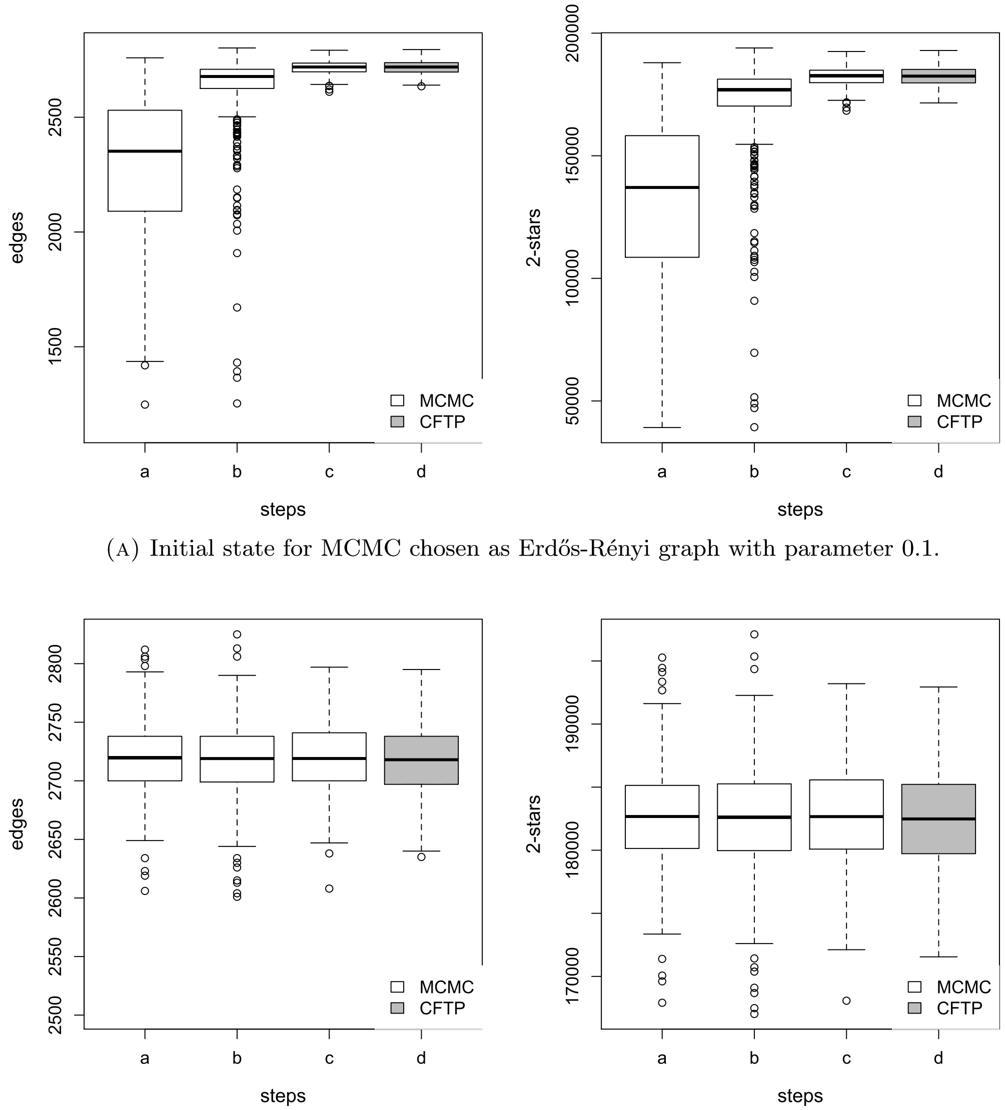
The following theorem shows that the expected value of T_N^{stop} is upper bounded by T_N^{mix}, up to a explicit multiplicative constant that depends on the number of vertices of the graph.

Theorem 4.1. *Let T_N^{mix} be the mixing time defined by (4.3) for the forward map (4.1). Then*

$$\mathbb{E}[T_N^{\text{stop}}] \;\leq\; 2\big(\log\big(\tfrac{N(N-1)}{2}\big) + 1\big)\, T_N^{\text{mix}}. \tag{4.4}$$

The proof of this relation is inspired from an argument given in [8], and is presented in Section 6. The upper bound in Theorem 4.1 on the mixing time of the chain to obtain an estimate of the expected value of T_N^{stop}, when they are available. As an example, we cite the results obtained by [1], where they present a study of the mixing time of the Markov chain constructed using the Glauber dynamics for ERGM as we consider here. They show that for models where $\boldsymbol{\beta}$ belongs to the high temperature regime the mixing time of the chain is $\Theta(N^2 \log N)$. On the other hand, for models under the low temperature regime, the mixing time is exponentially slow.

Observe that the mixing time T_N^{mix} is directly related to the "burn-in" time in MCMC, where the non-stationary forward Markov chain defined by (4.1) approaches the invariant distribution. This can be observed in practice in Figure 2: the convergence time for both MCMC and CFTP are of the same order of magnitude, as suggested by Theorem 1. In this figure we compare some statistics of the graphs obtained by MCMC at different sample sizes of the forward Markov chain and by CFTP at its convergence time. As it is usual in applications of the CFTP method, we run the algorithm using the time interval of 1500 steps between iterations. The model is defined by (2.2), where $\mathbf{g}_2$ is chosen as the graph with 3 vertices and 2 edges, with parameters $(\beta_1, \beta_2) = (-1.1, 0.4)$, and the mean proportion of edges in the graphs is approximately $p = 0.86$. We observe in Figure 2a that MCMC remains a long time strongly dependent on the initial value of the chain when this value is chosen in a region of low probability. On the other hand, when the initial value is chosen appropriately, then MCMC and CFTP have comparable performances, as shown in Figure 2b.



(A) Initial state for MCMC chosen as Erdős-Rényi graph with parameter 0.1.

(B) Initial state for MCMC chosen as Erdős-Rényi graph with parameter 0.8.

FIGURE 2. Boxplots of number of edges (*left*) and number of 2-stars (*right*) for the ERGM with $N = 80$ and parameter $(\beta_1, \beta_2) = (-1.1, 0.4)$. The MCMC algorithm was run for $n = 80.000$ time steps (respectively $n = 100.000$ (b) and $n = 130.000$ (c)) and two different initial states. The results for the CFTP algorithm at the time of convergence are shown in (d), with mean 130.500 time steps and with standard deviation 13209.

5. DISCUSSION

In this paper we proposed a perfect simulation algorithm for the Exponential Random Graph Model, based on the Coupling From The Past method of [8]. The justification of the correctness of the CFTP algorithm is generally restricted to the monotonicity property of the dynamics, given by Proposition 4.1, and to prove that T_N^{stop} is almost-surely finite, but with no clue about the expected time to convergence. Here, in contrast, we prove a much stronger result. Not only do we upper-bound the expectation of the waiting time, but we show that

CFTP compares favourably with standard MCMC forward simulation. The waiting time of CFTP is (up to a logarithmic factor) not larger than the time one has to wait before a run of the chain approaches the stationary distribution. Moreover, CFTP has the advantage of providing a draw from the exact stationary distribution, in contrast to MCMC which may keep for a long time a strong dependence on the initial state of the chain, as shown in the simulation example of Figure 2. We thus argue that for these models the CFTP algorithm is a better tool than forward MCMC simulation in order to get a sample from the Exponential Random Graph Model.

6. Proofs

Proof of Proposition 2.1. Let $\mathbf{x}$, $\mathbf{z} \in \mathcal{G}_N$ such that $\mathbf{x} \preceq \mathbf{z}$. For a pair of vertices (i,j) we define a new count that considers only the arrangements in V_{m_k} that contain the vertices i and j, that is

$$N_{\mathbf{g}_k}(\mathbf{x},(i,j)) = \sum_{\substack{v\in V_{m_k}\\ i,j\in v}} \mathbb{1}\{\mathbf{x}(v) \succeq \mathbf{g}_k\}. \tag{6.1}$$

Since $\mathbf{x}^1_{ij}$ is different from $\mathbf{x}^0_{ij}$ only in the edge (i,j), we have that $\mathbf{x}^0_{ij} \preceq \mathbf{x}^1_{ij}$ implying that $N_{\mathbf{g}_k}(\mathbf{x}^0_{ij}) \le N_{\mathbf{g}_k}(\mathbf{x}^1_{ij})$. In the same way, we have that $N_{\mathbf{g}_k}(\mathbf{z}^0_{ij}) \le N_{\mathbf{g}_k}(\mathbf{z}^1_{ij})$. So,

$$\begin{aligned} 0 \le N_{\mathbf{g}_k}(\mathbf{x}^1_{ij}) - N_{\mathbf{g}_k}(\mathbf{x}^0_{ij}) = N_{\mathbf{g}_k}(\mathbf{x}^1_{ij},(i,j))\\ 0 \le N_{\mathbf{g}_k}(\mathbf{z}^1_{ij}) - N_{\mathbf{g}_k}(\mathbf{z}^0_{ij}) = N_{\mathbf{g}_k}(\mathbf{z}^1_{ij},(i,j))\,. \end{aligned} \tag{6.2}$$

Since $\mathbf{x} \preceq \mathbf{z}$ we have that $N_{\mathbf{g}_k}(\mathbf{x}^1_{ij},(i,j)) \le N_{\mathbf{g}_k}(\mathbf{z}^1_{ij},(i,j))$. For $\beta_k \ge 0$, $k = 2,\cdots,s$, we have that

$$\begin{aligned} &\sum_{k=2}^{s} \frac{\beta_k}{N^{m_k-2}} \left(N_{\mathbf{g}_k}(\mathbf{x}^1_{ij}) - N_{\mathbf{g}_k}(\mathbf{x}^0_{ij}) - N_{\mathbf{g}_k}(\mathbf{z}^1_{ij}) + N_{\mathbf{g}_k}(\mathbf{z}^0_{ij})\right) \;=\\ &\sum_{k=2}^{s} \frac{\beta_k}{N^{m_k-2}} \left(N_{\mathbf{g}_k}(\mathbf{x}^1_{ij},(i,j)) - N_{\mathbf{g}_k}(\mathbf{z}^1_{ij},(i,j))\right) \;\le\; 0. \end{aligned} \tag{6.3}$$

Finally,

$$\begin{aligned} \frac{p_N(\mathbf{y}^1_{ij}\,|\boldsymbol{\theta})}{p_N(\mathbf{y}^0_{ij}\,|\boldsymbol{\theta})}\frac{p_N(\mathbf{z}^0_{ij}\,|\boldsymbol{\theta})}{p_N(\mathbf{z}^1_{ij}\,|\boldsymbol{\theta})} \;&=\; \exp\left\{\sum_{k=2}^{s} \frac{\beta_k}{N^{m_k-2}} \left(N_{\mathbf{g}_k}(\mathbf{x}^1_{ij},(i,j)) - N_{\mathbf{g}_k}(\mathbf{z}^1_{ij},(i,j))\right)\right\}\\ &\le 1 \end{aligned}$$

and this concludes the proof. □

Proof of Proposition 3.1. The proof that $\mathbb{P}(T^{\text{stop}}_N < \infty) = 1$ follows by a simple modification of Theorem 1(ii) in [2] and the Borel-Cantelli Lemma. Then the exactness of Algorithm 1 to produce a sample from the target distribution follows from Theorem 2 in [8]. □

Proof of Theorem 4.1. Define the stopping time of the forward algorithm by

$$\widetilde{T}^{\text{stop}}_N = \min\{n > 0 : F^n_0(\mathbf{x}^{(0)},\mathbf{e}^n_1,\mathbf{u}^n_1) = F^n_0(\mathbf{x}^{(1)},\mathbf{e}^n_1,\mathbf{u}^n_1)\}.$$

To prove the theorem claim we use the random variable $\widetilde{T}^{\text{stop}}_N$, since $\widetilde{T}^{\text{stop}}_N$ and T^{stop}_N have the same probability distribution. In fact,

$$\begin{aligned} \mathbb{P}(T^{\text{stop}}_N > n) &= \mathbb{P}(F^0_{-n}(\mathbf{x}^{(0)},\mathbf{e}^{-1}_{-n},\mathbf{u}^{-1}_{-n}) \ne F^0_{-n}(\mathbf{x}^{(1)},\mathbf{e}^{-1}_{-n},\mathbf{u}^{-1}_{-n}))\\ &= \mathbb{P}(F^n_0(\mathbf{x}^{(0)},\mathbf{e}^n_1,\mathbf{u}^n_1) \ne F^n_0(\mathbf{x}^{(1)},\mathbf{e}^n_1,\mathbf{u}^n_1) = \mathbb{P}(\widetilde{T}^{\text{stop}}_N > n)\,. \end{aligned} \tag{6.4}$$

Let $\{Y_n^1\}_{n\in\mathbb{N}}$ and $\{Y_n^0\}_{n\in\mathbb{N}}$ be the Markov chains obtained by (4.2) with initial states given by $\mathbf{x}^{(0)}$ and $\mathbf{x}^{(1)}$, respectively.

Given a graph $\mathbf{y}$, define by $l(\mathbf{y})$ the length of the longest strictly increasing sequence of graphs, according to the order given by (2.3), such that the last element in the sequence is $\mathbf{y}$. That is, $l(\mathbf{y})$ is the largest value of $n \in \mathbb{N}$ such that $\mathbf{x}^{(0)} \precneqq \mathbf{z}_1 \precneqq \cdots \precneqq \mathbf{z}_{n-2} \precneqq \mathbf{y}$.

By the construction of the process and the monotonicity property we have that $Y_n^0 \preceq Y_n^1$ for all n. Therefore $l(Y_n^0) \leq l(Y_n^1)$, and if in addition $Y_n^0 \neq Y_n^1$ then $l(Y_n^0) + 1 \leq l(Y_n^1)$. Consequently, it holds that

$$\begin{aligned}\mathbb{P}(\widetilde{T}_N^{\text{stop}} > n) &= \mathbb{P}(Y_n^0 \neq Y_n^1) = \mathbb{P}(l(Y_n^0) + 1 \leq l(Y_n^1)) \\ &\leq \mathbb{E}[l(Y_n^1) - l(Y_n^0)] = |\mathbb{E}_{\mathrm{p}_n^1}[l(Y)] - \mathbb{E}_{\mathrm{p}_n^0}[l(Y)]| \\ &\leq \| \mathrm{p}_n^1 - \mathrm{p}_n^0 \|_{\mathrm{TV}} \, [\max_{\mathbf{x}\in\mathcal{G}_N} l(\mathbf{x}) - \min_{\mathbf{x}\in\mathcal{G}_N} l(\mathbf{x})] \leq \bar{d}(n) \max_{\mathbf{x}\in\mathcal{G}_N} l(\mathbf{x})\,.\end{aligned} \tag{6.5}$$

Since the update function of the Markov chain given by (3.3) can add only one edge at a time, we have that

$$\max_{\mathbf{x}\in\mathcal{G}_N} l(\mathbf{x}) \;=\; l(\mathbf{x}^{(1)}) \;=\; \frac{N(N-1)}{2}.$$

Thus

$$\bar{d}(n) \;\geq\; \frac{2}{N(N-1)}\,\mathbb{P}(\widetilde{T}_N^{\text{stop}} > n)\,.$$

Since $\mathbb{P}(\widetilde{T}_N^{\text{stop}} > n)$ is submultiplicative ([8], Thm. 6) we have that

$$\mathbb{E}[\widetilde{T}_N^{\text{stop}}] \leq \sum_{i=1}^{\infty} n\mathbb{P}(\widetilde{T}_N^{\text{stop}} > in) \leq \sum_{i=1}^{\infty} n[\mathbb{P}(\widetilde{T}_N^{\text{stop}} > n)]^i = \frac{n}{\mathbb{P}(\widetilde{T}_N^{\text{stop}} \leq n)}. \tag{6.6}$$

Then, (6.5) and (6.6) imply that

$$\mathbb{E}[\widetilde{T}_N^{\text{stop}}] \;\leq\; \frac{n}{1 - \bar{d}(n)\frac{N(N-1)}{2}}\,. \tag{6.7}$$

Set $n = \left(\log(\frac{N(N-1)}{2}) + 1\right) T_N^{\text{mix}}$. Using the submultiplicative property of $\bar{d}(n)$ ([6], Lem. 4.12) we have that

$$\bar{d}(n) \;\leq\; [\bar{d}(T_N^{\text{mix}})]^{\log(\frac{N(N-1)}{2})+1} \;\leq\; \frac{2}{eN(N-1)} \tag{6.8}$$

and by (6.7) and (6.8) we conclude that

$$\mathbb{E}[\widetilde{T}_N^{\text{stop}}] \;\leq\; \frac{\left(\log\left(\frac{N(N-1)}{2}\right) + 1\right) T_N^{\text{mix}}}{1 - \frac{1}{e}} \;\leq\; 2\left(\log\left(\frac{N(N-1)}{2}\right) + 1\right) T_N^{\text{mix}}.$$

□

Acknowledgements. This work was produced as part of the activities of FAPESP's Research, Innovation and Dissemination Center for Neuromathematics, grant 2013/07699-0, and FAPESP's project "Model selection in high dimensions: theoretical properties and applications", grant 2019/17734-3, São Paulo Research Foundation. AC was supported by grant 2015/12595-4, São Paulo Research Foundation and FL was partially supported by a CNPq's research fellowship (309964/2016-4). This work was completed while AC was a PhD student at Instituto de Matemática e Estatística, Universidade de São Paulo, and a Visiting Student at Institut de Mathématiques de Toulouse, Université Paul Sabatier. She thanks the support and hospitality of both institutions.

References

[1] S. Bhamidi, G. Bresler and A. Sly, Mixing time of exponential random graphs. *Ann. Appl. Probab.* **21** (2011) 2146–2170.

[2] C.T. Butts, A perfect sampling method for exponential family random graph models. *J. Math. Sociol.* **42** (2018) 17–36.

[3] A. Cerqueira, D. Fraiman, C.D. Vargas and F. Leonardi, A test of hypotheses for random graph distributions built from EEG data. *IEEE Trans. Netw. Sci. Eng.* (2017).

[4] S. Chatterjee, P. Diaconis *et al.*, Estimating and understanding exponential random graph models. *Ann. Stat.* **41** (2013) 2428–2461.

[5] C.J. Geyer and E.A. Thompson, Constrained Monte–Carlo maximum likelihood for dependent data. *J. Roy. Stat. Soc. Ser. B (Methodological)* **54**, (1992) 657–699.

[6] D.A. Levin, Y. Peres and E.L. Wilmer, Markov chains and mixing times. American Mathematical Soc. (2009).

[7] M.E. Newman, D.J. Watts and S.H. Strogatz, Random graph models of social networks. *Proc. Natl. Acad. Sci.* **99** (2002) 2566–2572.

[8] J.G. Propp and D.B. Wilson, Exact sampling with coupled markov chains and applications to statistical mechanics. *Random Struct. Algor.* **9** (1996) 223–252.

[9] G. Robins, P. Pattison, Y. Kalish and D. Lusher, An introduction to exponential random graph (p*) models for social networks. *Social Netw.* **29** (2007) 173–191.

[10] T.A. Snijders, Markov chain Monte Carlo estimation of exponential random graph models. *J. Soc. Struct.* **3** (2002) 1–40.

[11] D. Strauss and M. Ikeda, seudolikelihood estimation for social networks. *J. Am. Stat. Assoc.* **85** (1990) 204–212.