Generating constrained random graphs using multiple edge switches

Lionel Tabourier[a,c,d], Camille Roth[b,c,d], Jean-Philippe Cointet[d,c,e]

[a]*SPEC, CEA Saclay*
Orme des Merisiers, F-91191 Gif-sur-Yvette, France
[b]*CAMS, CNRS/EHESS*
54 bd Raspail, F-75006 Paris, France
[c]*Institute of Complex Systems of Paris-Ile-de-France*
59, rue Lhomond, F-75005 Paris, France
[d]*CREA, CNRS/Ecole Polytechnique*
ENSTA 32 bd Victor, F-75015 Paris, France
[e]*TSV, INRA*
65 Boulevard de Brandebourg
F-94205 Ivry-sur-Seine Cedex, France

Abstract

The generation of random graphs using edge swaps provides a reliable method to draw uniformly random samples of sets of graphs respecting some simple constraints, e.g. degree distributions. However, in general, it is not necessarily possible to access all graphs obeying some given constraints through a classical switching procedure calling on pairs of edges. We therefore propose to get round this issue by generalizing this classical approach through the use of higher-order edge switches. This method, which we denote by “k-edge switching”, makes it possible to progressively improve the covered portion of a set of constrained graphs, thereby providing an increasing, asymptotically certain confidence on the statistical representativeness of the obtained sample.

Key words: graph algorithms, random graphs, edge switching, Markov-chain mixing, constrained graphs

---

Introduction

The generation of random graphs respecting some constraints has two direct applications: the modeling of realistic network topology when empirical data are missing, and the confirmation of the role of a given set of constraints in the presence of some empirically observed topological and structural features (i.e. some *target observables*, such as in e.g. [17]). There is however currently no general approach to directly create uniformly random graph samples given arbitrary constraints, except for some very specific cases usually related to degree distributions (in this paper, *degree distribution* refers to a specific sequence of degrees, as opposed to a *probability distribution*).

Existing methods for generating random samples of a set of graphs $\mathcal{G}_\mathbf{C}$ respecting a given set of constraints $\mathbf{C}$ fall indeed into two broad categories:

- Either by directly building a graph of $\mathcal{G}_\mathbf{C}$ from scratch, i.e. randomly assigning links to pairs of nodes such that the overall constraint is respected. For instance, the configuration model as presented by [4] provides random graphs by connecting half-links on nodes such that each resulting graph respects a given prescribed degree distribution.
- Or by using an original graph which already belongs to $\mathcal{G}_\mathbf{C}$ and iteratively reshuffling edges of this graph while altogether remaining in $\mathcal{G}_\mathbf{C}$ in order to asymptotically converge, after a “sufficient” number of iterations, to a uniformly random element of $\mathcal{G}_\mathbf{C}$. This approach

---

Email addresses: `lionel.tabourier@ens-lyon.org` (Lionel Tabourier), `roth@ehess.fr` (Camille Roth), `cointet@polytechnique.edu` (Jean-Philippe Cointet)

of switching pairs of edges has been proposed for instance by [19] who aim at obtaining a random graph with a given degree distribution by switching pairs of links in an initial graph which already respects this constraint.

The asymptotical convergence is generally empirically appraised with respect to the target observables. Besides, approaches based on edge swaps implicitly assume that the number of nodes N, the number of edges M and the degree sequence are part of $\mathbf{C}$. In this case, we consider that $\mathbf{C}$ is the union of two subsets: $\mathbf{C} = \mathbf{C}^{\emptyset} \cup \mathbf{C}^{+}$, where $\mathbf{C}^{\emptyset}$ refers to the fundamental constraint forcing graphs to have M links, N nodes, a given degree sequence and to be of a certain type (simple graphs, multigraphs, etc.), while $\mathbf{C}^{+}$ refers to some additional and arbitrary set of constraints, depending on the context.

While the former method assuredly poses a new design challenge for every new kind of constraint — each set of constraints basically requires a new configuration model — on the other hand, the latter approach raises the issue of obtaining *uniformly random* elements of $\mathcal{G}_{\mathbf{C}}$. Put differently and as we will see below, this reshuffling approach, which initially requires at least one graph from $\mathcal{G}_{\mathbf{C}}$, does not guarantee in general that the final graph is *uniformly* chosen from the *whole* set $\mathcal{G}_{\mathbf{C}}$.

We propose to both (i) appraise the potential issues and drawbacks of random graph creation based on pairwise edge switching (Sec. 1), which is a relatively traditional method in the literature [8, 5, 23, 24, 19, 13, 20, 16, 10, 22, 2, 25, 7, 9, 14, 3, 6] and, then, (ii) introduce a method for producing random, simulation-based samples of graphs for arbitrary constraints $\mathbf{C}$, using higher-order edge switching processes (Sec. 2). We eventually present several practical and empirical illustrations in Sec. 3.

1. Edge swaps as a Markovian reshuffling process

Miklós *et al.* [15] showed that it is possible to use a *pairwise edge switching* reshuffling algorithm to generate a uniformly random sample of oriented graphs whose degree distributions are fixed. [2] later called this method "switching and holding" (*S&H*). More precisely, this edge switching method comes to randomly choosing two links in the current graph, checking whether swapping these links leads to a graph respecting the constraint and, if yes, carry out the corresponding swap, otherwise, "hold" the current graph and reiterate the procedure. Note that, as such, *S&H* differs from a simple switching method in that it focuses on the number of swap *trials* rather than the number of swaps.

This procedure may be described as a walk in a *Markov graph*. The Markov graph is a directed graph, allowing self-loops and multiple edges such that its set of nodes is exactly $\mathcal{G}_{\mathbf{C}}$. Arcs of the Markov graph are such that, (i) whenever a valid pairwise edge switch transforms $G_i \in \mathcal{G}_{\mathbf{C}}$ into $G_j \in \mathcal{G}_{\mathbf{C}}$, we draw an arc from G_i to G_j (and vice versa, mechanically), and (ii) whenever a pairwise edge switch transforms $G_i \in \mathcal{G}_{\mathbf{C}}$ into a graph which does not belong to $\mathcal{G}_{\mathbf{C}}$, we draw a self-loop from G_i to G_i. In this context, the reshuffling procedure is a random walk in the Markov graph, that is, a Markov chain [21] converging to an equilibrium distribution whose probabilities can be obtained from the transition matrix of the Markovian process. If the Markov graph has constant degrees (i.e. the in-degree and out-degree of all graphs of the Markov graph are all the same), the reshuffling process is uniform. If the Markov graph is connected, all possible graphs are reachable. If it is both connected and has constant degrees, the process leads to uniformly random elements of $\mathcal{G}_{\mathbf{C}}$. See an illustration on Fig. 1.

Edge switching methods have been used to generate random graph samples in various instances [19, 13, 22, 7, 9, 14, 3] and have been studied and improved in various directions [20, 16, 10, 2, 25]. To use such a switching method, one has nonetheless to ensure that all graphs of $\mathcal{G}_{\mathbf{C}}$ are present in the equilibrium distribution of the random walk with an identical probability, i.e. ensure that:

(i) all graphs of $\mathcal{G}_{\mathbf{C}}$ are *uniformly* drawable, and
(ii) all graphs of $\mathcal{G}_{\mathbf{C}}$ are *exhaustively* reachable.

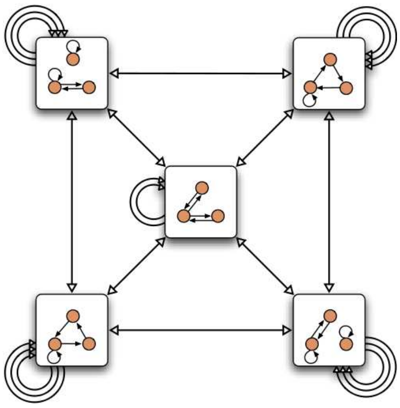

Figure 1: Simple Markov graph for a constraint on a graph of (i) three nodes with (ii) given in- and out-degree distributions and (iii) without multiple edges but possibly self-loops. Non-valid swaps are represented by self-loops in this Markov graph, which has thus a constant degree.

Uniformity is guaranteed by the *S&H* approach within a given connected portion of the Markov graph. While [15] show uniformity in the case of degree distribution constraints, the proof they mention in Appendix A of the same reference can easily be extended to any kind of constraint. A sketch of this proof is given by the following reasoning: "holding" on failed trials is equivalent to connecting a Markov graph node to itself as many times as there are failure possibilities. Thus, the in- and out-degree of all Markov graph nodes will be equal to the number of trials (both failed and successful ones), which is strictly the same for every graph of $\mathcal{G}_{\mathbf{C}}$, since it only depends on the constant number of links of graphs of $\mathcal{G}_{\mathbf{C}}$. Finally, for a random walk in a Markov graph where all nodes have the same in and out-degree, the probability of being on a given node is asymptotically uniform.

Exhaustivity relates to the issue of whether the whole Markov graph is connected, i.e. the existence of a path going from any node to any other node of the Markov graph. In Markov chain terminology, the chain is said to be *irreducible*. To our knowledge, existing theorems on exhaustivity concern simple constraints $\mathbf{C}$, essentially reduced to little more than the conservation of the original degree sequence: i.e. in the case of trees [5], graphs [8], connected graphs [23] and bi-connected graphs [24].

However in the general case of more elaborate constraints (e.g. [14, 3]), using the *S&H* method appears to be less legitimate, since no such exhaustivity theorems are known. For instance, Rao *et al.* [19] show that extending $\mathbf{C}$ by requiring the graph to have both directed edges and no self-loop makes it impossible, in some cases, to reach all graphs of $\mathcal{G}_{\mathbf{C}}$ by pairwise edge swaps. In particular, no pairwise edge switch could indeed transform one of the following adjacency matrices into the other one (forbidden self-loops are marked with a star):

$$\begin{pmatrix} 0^* & 1 & 0 \\ 0 & 0^* & 1 \\ 1 & 0 & 0^* \end{pmatrix} \not\leftrightarrow \begin{pmatrix} 0^* & 0 & 1 \\ 1 & 0^* & 0 \\ 0 & 1 & 0^* \end{pmatrix}$$

Convergence of the walk. In addition to these issues, convergence speed remains an open theoretical question [19, 12], often coped with using practical heuristics [10, 25]. As said before, the walk usually aims at randomly drawing an element of $\mathcal{G}_{\mathbf{C}}$ in order to check whether graphs of $\mathcal{G}_{\mathbf{C}}$ exhibit some properties on the target observables (and, implicitly, in order to check whether $\mathbf{C}$ could constitute a sufficient explanation for these observables). In other words, some measurements are carried out on graphs of $\mathcal{G}_{\mathbf{C}}$ so that the walk is generally considered to have performed a "sufficient" number of steps when those measurements on the target observables apparently plateau.

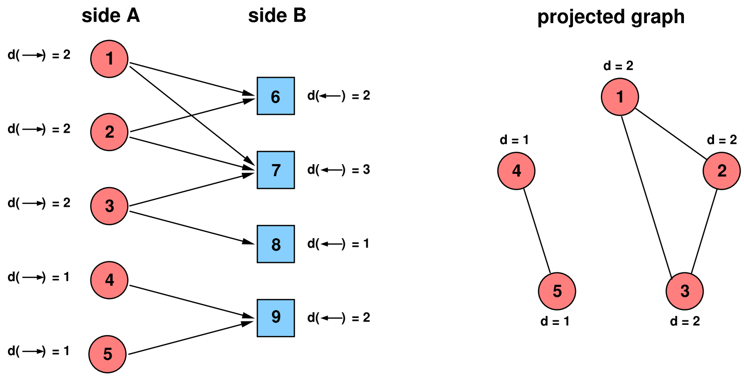

Figure 2: *On the left,* one possible realization of a graph drawn from $\mathcal{G}_{C_0}$: note that B-sided nodes of the bipartite graph (marked by squares) have out-degree zero and A-sided nodes (marked by circles) have in-degree zero. *On the right,* the corresponding projection of this bipartite graph onto side A.

2. Higher-order switching process

In this section, for the sake of clarity we focus on directed graphs, although it is effortless to formulate the whole argument for undirected graphs.

2.1. k-edge switching

In general, the disconnectedness of the Markov graph stems from the impossibility of transforming a graph into another graph by a simple pairwise switching. To overcome this issue, we propose an experimental method based on higher-order edge switchings: given $G \in \mathcal{G}_{\mathbf{C}}$, let us consider k edges $(a_i, b_i)_{i\in\{1,\ldots,k\}}$ corresponding to nodes $(a_1, \ldots, a_k, b_1, \ldots, b_k)$, possibly not all distinct. The k-edge switching process, henceforth called "*k-switch*", comes to randomly choosing one permutation σ among the $k!$ possible permutations of $(b_1, \ldots, b_k)$. The resulting graph is such that edges $(a_i, b_i)_{i\in\{1,\ldots,k\}}$ are replaced with $(a_i, \sigma(b_i))_{i\in\{1,\ldots,k\}}$ (for an example of pseudocode, see Alg. 1).

It is immediate to see that neighbors of G in the Markov graph corresponding to a classical pairwise edge swap are also neighbors of G in the Markov graph corresponding to a k-switch, when considering a permutation that just swaps two b_i, $b_{i'}$. Similarly, when $k = 2$, we fall back on the *S&H* approach.

For increasing values of k, this procedure creates new links in the Markov graph and new neighbors appear (in the case of Fig. 1 it is easy to see that the Markov graph is complete for $k = 3$). More importantly, some potentially disconnected components of the Markov graph may thus become connected.

Illustration. To illustrate this higher-order switching process, let us consider the case of bipartite (or 2-mode) graphs. Such graphs are useful in the context of real-world networks, for example to study collaborations in social groups [17] or peer-to-peer exchange systems [11]. Nodes belong to one of two sides A and B, and links connect pairs of nodes from distinct sides only. It is possible to build monopartite (or 1-mode) graphs from the bipartite one by keeping only A (resp. B) nodes and linking them if they are connected to the same B (resp. A) node in the original bipartite structure, as pictured on Figure 2. These graphs are called *projections* of the original bipartite graph on side A (resp. B).

Consider a case consisting of a constraint $\mathbf{C}_0 = \mathbf{C}_0^{\emptyset} \cup \mathbf{C}_0^{+}$, on bipartite graphs such that:

(i) $\mathbf{C}_0^{\emptyset}$: *the bipartite graph contains no multiple link, it consists of two sides with fixed degree distributions:*
  - *"side A": 5 nodes, out-degree* $\{2, 2, 2, 1, 1\}$ *(and in-degree 0);*

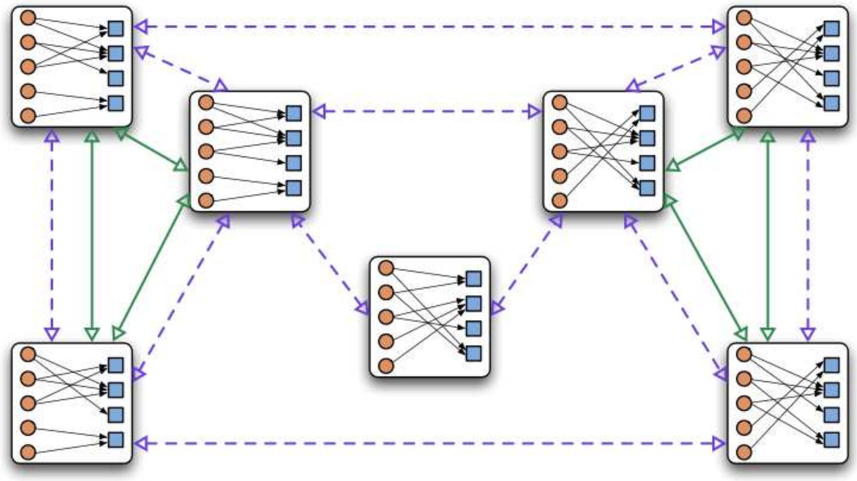

Figure 3: Markov graph of $\mathcal{G}_{\mathbf{C}_0}$ for various k-switching procedures: dashed blue arrows correspond to $k = 2$, plain green arrows to $k = 4$. For readability purposes, we simplified the representation by discarding self-loops and multiple edges of the Markov graph.

- *"side B": 4 nodes, in-degree* $\{3, 2, 2, 1\}$ *(and out-degree 0).*

(ii) $\mathbf{C}_0^+$*: the degree distribution of the projected graph on side A is fixed:* $\{2, 2, 2, 1, 1\}$.

Put shortly, this constraint consists in simultaneously imposing degree distributions on a bipartite graph and on one of its monopartite projections. An example of such a graph is represented Fig. 2. Given such a $\mathbf{C}_0$, Markov graphs corresponding to $\mathcal{G}_{\mathbf{C}_0}$ contain 7 nodes. The Markov graph for $k \geq 4$ is connected, while it actually consists of 3 disconnected components for $k \in \{2, 3\}$ — see Fig. 3.

We chose this practical case in part because the Markov graph is still small enough to be visualized for each value of k. In the remaining examples, it will not be possible anymore, and no theoretical proof is available; we therefore rely on experimental investigations.

2.2. Relationship between k and exhaustivity

There is an upper bound on k such that the Markov graph is assuredly connected and the underlying walk is exhaustive/irreducible. In particular, given two graphs G_1 and G_2 of $\mathcal{G}_{\mathbf{C}}$, there always exists a permutation of size at most M (the number of edges) such that G_1 is transformed into G_2.

Proof. The M edges of G_1 can be written as $\{(a_1, b_1); (a_2, b_2); ...; (a_M, b_M)\}$. Similarly, in G_2, because both M and degree sequence are fixed, we can write that M edges originate from the same family $(a_i)_{i \in \{1,...,k\}}$ to another family $(b'_i)_{i \in \{1,...,k\}}$, i.e. these edges can be written as $\{(a_1, b'_1); (a_2, b'_2); ...; (a_M, b'_M)\}$. Because the degree sequence is fixed, families of nodes b and b' contain exactly the same nodes repeated the same number of times. Thus, σ defined as $(b_1, b_2, ..., b_M) \xrightarrow{\sigma} (b'_1, b'_2, ..., b'_M)$ is then a valid M-switch permutation which does transform G_1 into G_2. $\square$

The number of connected components of the Markov graph is thus a monotonously decreasing function of k converging at most for $k = M$. As increasing k guarantees a better coverage of the Markov graph, the relevance of this method lies essentially in improving the confidence in the random mixing achieved by rewiring procedures — rather than addressing convergence speed issues.[1]

---

[1] In practice, increasing k comes however at the price of an increasingly slow convergence of the walk, in terms

2.3. Data storage format

One of the first requirements for the data format is to enable quick random selection of edges and subsequent edge switches, i.e. update of the graph. A straightforward option for drawing random links consists in using an array of edges, and picking a random integer lower or equal to the array size. To store the graph, by contrast, we opt for an adjacency list, especially because the operation of constraint checking often requires to access neighbors of a given node (which is possible in $O(\delta)$, where δ is the node degree). Eventually, we thus maintain and update two data structures: an adjacency list and an array. These two data strutures have a comparable size and are respectively most efficient for link selection and graph operations.

2.4. Complexity

Carrying out a k-switch in $G \in \mathcal{G}_{\mathbf{C}}$ consists in:

1. Finding k random edges in G represented as an adjacency list, in $O(k)$;
2. k-switching their extremities into a resulting graph G', in $O(k)$;
3. Verifying that G' respects the constraint set, i.e. $G' \in \mathcal{G}_{\mathbf{C}}$, in $O(f_{\mathcal{G}_{\mathbf{C}}})$ related to a given design of the constraint check.

$\mathbf{C}$ should be such that there exists a tractable check on any graph of $\mathcal{G}_{\mathbf{C}}$.[2] In best cases when it is possible to check incrementally if $G' \in \mathcal{G}_{\mathbf{C}}$ relatively to the k switched edges only, $f_{\mathcal{G}_{\mathbf{C}}}$ at best belongs to $O(k)$. The complexity of doing n trials of k-switches is thus at least $O(nk)$.

Additionally, target observables have to be computed at regular intervals to monitor their asymptotical convergence. Those target observables shall also be chosen to be tractable. If, moreover, the observation frequency is chosen to be sufficiently low, constraint checking shall dominate the overall running time.

Algorithm 1: Pseudocode of the k-switching procedure in the case of a directed network with constraints: degree distributions, no self-loops, no multiple arcs and a set of constraints $\mathbf{C}^+$ (associated to the set $\mathcal{G}_{\mathbf{C}^+}$), which depends on the context.

```
input : Graph $G_0 = (V_0, E_0)$, stored as an array of adjacency lists; number of switching
        trials: $n$ ; size of the switches: $k$;
output: graph $G$ produced by $n$ attempts of switching;
$G = (V, E) \leftarrow G_0$ ;                                      // initialization
for $j \leftarrow 1$ to $n$ do
    draw randomly $k$ different arcs : $\{(a_i\ ,\ b_i)\}_{i \in I} \in E$ ;
    draw randomly $\sigma$ a permutation of the index set $I$ ;
    build the set of swapped arcs $\left\{\left(a_i\ ,\ b_{\sigma(i)}\right)\right\}_{i \in I}$ ;
    $E' \leftarrow E \cup \left\{\left(a_i\ ,\ b_{\sigma(i)}\right)\right\} \setminus \{(a_i\ ,\ b_i)\}$ ;
    define $G' = (V, E')$ ;
    define $\forall i \in I\ ,\ \mathcal{W}_i = \{b\ :\ \exists\ (a_i\ ,\ b) \in E\} \setminus \{b_i\}$ ;
    if $\forall i\ ,\ a_i \neq b_{\sigma(i)}$                         // test no self-loops
    and $\forall i\ ,\ b_{\sigma(i)} \notin \mathcal{W}_i$                  // test no multiple arcs
    and $G' \in \mathcal{G}_{\mathbf{C}^+}$                              // test constraint $\mathbf{C}^+$
    then $G \leftarrow G'$ ;
end
```

The reason why large values of k are not necessarily advisable actually lies in the possibility of k-switch *failures*, i.e. such that the resulting graph does not anymore belong to $\mathcal{G}_{\mathbf{C}}$ and thus the walk stays on the same graph at the next step. Odds of such failure depend in a complicated way on k: on one hand, when increasing k we are allowing new types of switches, therefore allowing

---

of switch trials, as detailed in the following subsection on complexity.

[2] Various optimizations of this very step are open to a discussion which depends on the chosen external set of constraints $\mathbf{C}$, but are obviously outside the scope of the present paper. In particular, we assume that $f_{\mathcal{G}_{\mathbf{C}}}$ is not e.g. exponential in N or M.

access to possibly more graphs from a given graph of $\mathcal{G}_{\mathbf{C}}$. On the other hand, many of these new possible k-switches are also likely to fail (i.e. fall on a graph which does not belong to $\mathcal{G}_{\mathbf{C}}$), because they alter more deeply the graph (i.e. more deeply than k'-switches for $k' < k$). In the end, the proportion of k-switch *successes* generally depends on k in a non-monotonous manner.

In practice, given an *a priori* fixed number of trials, we observe that the number of successful alterations tends to decrease sharply for large values of k (as shown below e.g. in Tab. 2). In other words, high-order alterations apparently make the walk stay longer on a given graph, although at the same time successful alterations reshuffle more strongly the graph. Put shortly, with increasing k, the walk is more likely to stagnate, but when it does not, it is more likely to lead to more different graphs.

2.5. Random graph sampling using k-switches

It is therefore hard to assess whether the mixing achieved by a k-switch-based walk of given length is more efficient or not for higher values of k. However, the number of connected components of the Markov graph is monotonously decreasing with k: increasingly connected portions of $\mathcal{G}_{\mathbf{C}}$ are visited with increasing values of k. Because of that, it is relevant to propose an asymptotical approach on k.

More precisely, a k-switch walk is stopped when some measures on $\mathcal{G}_{\mathbf{C}}$ apparently plateau to some values. Starting with the traditional case $k = 2$, we thus progressively increase k up to a "sufficient" value, i.e. such that the measurements appear to plateau from some k_0; as is classical in asymptotical convergence of simulation-based methods. As we will see in the following section, it seems empirically that even very small values of k are often satisfactory.

3. Illustrations on practical cases

In addition to the earlier toy example $\mathbf{C}_0$ shown on Fig. 3 on an extremely small graph, we now illustrate this asymptotical approach on four practical cases for various kinds of constraints. For the sake of clarity, we gathered in Appendix 3.4 the descriptions of constraint checking algorithms and their respective complexity. Note that, here, we only consider constraints on graphs without multiple edges; the higher-order switching approach may nonetheless be used in the context of multigraphs.

3.1. Constraint based on oriented and colored triangles

We first suggest a quite fictitious constraint $\mathbf{C}_1$ such that:

(i) $C_1^{\emptyset}$: *the graph is directed and made of N nodes, each one having one outgoing and one incoming arc;*

(ii) C_1^{+}:
  - *nodes are equally divided into 3 groups of $N/3$ nodes, each denoted with a color: red (R), green (G), or blue (B);*
  - *the graph is made of $N/3$ isolated and oriented cycles of 3 nodes (i.e. N isolated triangles such that each node points to a single other node of the triangle).*

Graphs of $\mathcal{G}_{\mathbf{C}_1}$ are thus such that each node exactly has an in-degree of 1 and an out-degree of 1. Suppose we want to randomly draw an element of $\mathcal{G}_{\mathbf{C}_1}$ using k-switches, starting with an initial graph G_0 such that each triangle is "R-G-B-oriented", i.e. a red node points to a green one which points to a blue one which points to the red one.

For $k = 2$, the only possible k-switch is identity, so that in the Markov graph it is not possible to leave G_0. For $k = 3$, possible k-switches reshuffle links within a given triangle, as illustrated on Fig. 4; the associated walk can only lead to "R-G-B-oriented" and "R-B-G-oriented" triangles. For $k = 4$, link exchanges are possible between triangles, so that eventually all combinations of colored triangles are possible (including non trichromatic triangles "R-R-R", "R-G-G", etc.).[3]

---

[3] The corresponding Markov graph is thus connected for $k = 4$, which hence happens much before $k = M$.

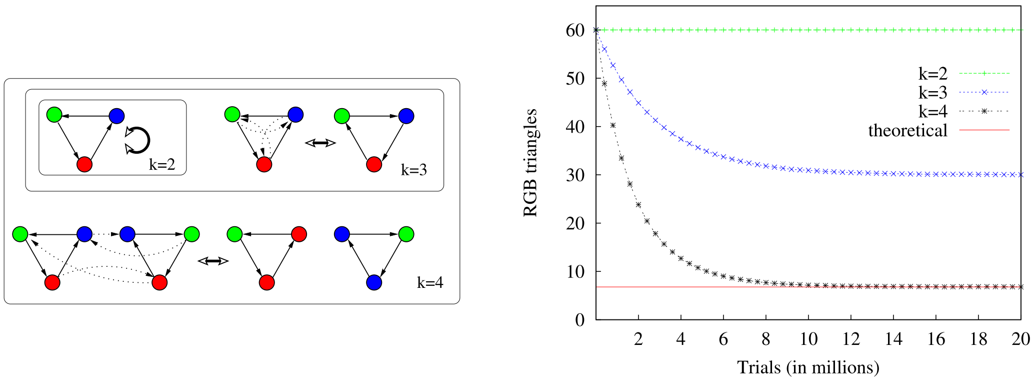

Figure 4: *Left:* Illustration of the increasing possibilities of k-switches for $k \in \{2, 3, 4\}$ in the case of "R-B-G" triangles. *Right:* Number of "R-B-G" triangles with respect to the number of k-switch trials, for $k \in \{2, 3, 4\}$ (averages and corresponding confidence intervals computed over 10 000 simulations for each k).

Table 1: Proportion of triangles of each type with respect to k, averaged over 10000 completed simulations consisting of 10^8 trials, including the respective mean number of effectively successful k-switches. The last column features the theoretical average value over all graphs of $\mathcal{G}_{\mathbf{C}_1}$.

	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	*Theoretical* $\langle \mathcal{G}_{\mathbf{C}_1} \rangle$
R-R-R	0.	0.	0.036	0.036	0.036	0.036
G-G-G	0.	0.	0.036	0.036	0.036	0.036
B-B-B	0.	0.	0.036	0.036	0.036	0.036
R-G-G	0.	0.	0.111	0.111	0.111	0.111
R-B-B	0.	0.	0.111	0.111	0.111	0.111
G-G-B	0.	0.	0.111	0.111	0.111	0.111
G-B-B	0.	0.	0.111	0.111	0.111	0.111
R-R-B	0.	0.	0.111	0.111	0.111	0.111
R-R-G	0.	0.	0.111	0.111	0.111	0.111
R-B-G	0.	0.500	0.113	0.113	0,113	0.113
R-G-B	1.000	0.500	0.113	0.113	0,113	0.113
Successes	0	997 ± 74	2643 ± 108	2067 ± 132	936 ± 55	-

Considering a trivial target observable which is the proportion of triangles of a given color-orientation, we now compare the performance of k-switch-based walks for $k \in \{2, 3, 4, 5, 6\}$. Using simulations on graphs of $N = 180$ nodes, we consider the plateauing values of each walk, as shown on Fig. 4. We then gather in Tab. 1 the various averages of such values obtained over 10000 simulations for each k. We see that average values plateau for $k = 4$ which generally fits well the theoretical values, which can be analytically computed for $\mathbf{C}_1$ (see also Tab. 1). However, values obtained for $k = 2$ (classical *S&H*) and $k = 3$ are significantly different from the theoretical values, indicating that the corresponding Markov processes are unable to reach every graph of the set $\mathcal{G}_{\mathbf{C}_1}$. In particular, the classical *S&H* method cannot be used in the case of $\mathbf{C}_1$ to generate a random sample, whereas the multiple edges switching method with $k \geq 4$ is reliable.

Such apparently arbitrary constraints can actually be relevant when considering e.g. complex molecular edifices modeled as graphs linking molecules according to chemical constraints [18].

3.2. Constraint based on correlations of degrees

We now consider constraint $\mathbf{C}_2$ imposing that:

- $\mathbf{C}_2^{\emptyset}$*: the graph is directed, without self-loops nor multiple edges and has a fixed degree sequence,*
- $\mathbf{C}_2^{+}$*: the distribution of out-degree correlations between pairs of connected nodes is fixed. In other words, the number of links connecting nodes of some out-degree to nodes of some (possibly distinct) out-degree remains the same across the set of graphs.*

The practical interest of this constraint becomes explicit in the empirical case of a hyperlink citation network. In qualitative terms, this constraint should in effect help in appraising how much correlations in citing activities (in terms of out-degrees) explain the existence of cyclic citation patterns (in terms of directed triangles). To this end, we start with an initial graph G_0 extracted from the 50,000 first web pages from the network database used in [1][4], we denote this database *WWW*. We carry out one billion trials in each walk corresponding to k-switches for $k \in [2, 6]$. We measure the average number of directed triangles (i.e. oriented cycles of length 3) of graphs of $\mathcal{G}_{\mathbf{C}_2}$ thereby estimating how much $\mathbf{C}_2$ contributes to this kind of topological patterns. Results are gathered on Tab. 2 and Fig. 5.

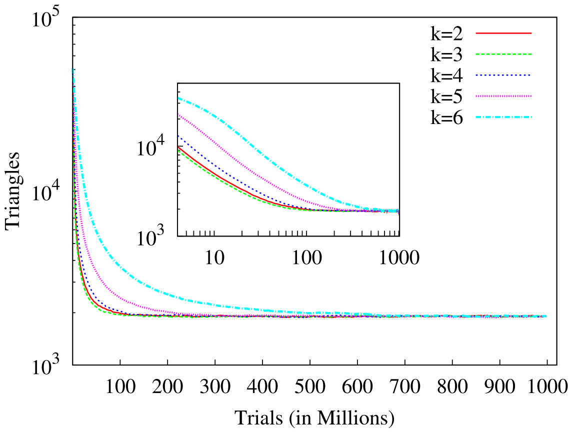

Figure 5: Number of directed triangles with respect to the number of k-switch trials ($k \in [2, 6]$).

Table 2: Number of directed triangles with respect to k, averaged over 50 completed walks consisting of 1 billion trials, and respective number of effectively successful k-switches. Standard deviation are generally negligible and never exceed 5% of the observed mean.

Target observables	Starter G_0	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$
	$50.77{\cdot}10^3$	$1.92{\cdot}10^3$	$1.91{\cdot}10^3$	$1.91{\cdot}10^3$	$1.92{\cdot}10^3$	$1.91{\cdot}10^3$
	$31.70{\cdot}10^4$	$2.90{\cdot}10^4$	$2.88{\cdot}10^4$	$2.89{\cdot}10^4$	$2.90{\cdot}10^4$	$2.88{\cdot}10^4$
	15,423	59	56	58	58	59
Successes	-	$6.96{\cdot}10^7$	$8.22{\cdot}10^7$	$5.28{\cdot}10^7$	$2.50{\cdot}10^7$	$1.00{\cdot}10^7$

In spite of their diverse convergence speeds and success rates, $\forall k \in \{2, 3, 4, 5, 6\}$ walks converge to a same average number of such directed triangles. As is usually the case with random graphs with constraints, and contrarily to the previous example, we are trying to empirically estimate the theoretical average of this measure on $\mathcal{G}_{\mathbf{C}_2}$. We therefore assume that the plateauing of limit measures for increasing k is a sufficient indication that this empirical estimate can be trusted, which is classical with simulation-based convergence — similarly, the user may also decide to extend simulations to higher values of k. These results suggest that the reshuffling process is likely to cover well $\mathcal{G}_{\mathbf{C}_2}$ even for $k = 2$, i.e. traditional edge swaps. As such, the k-switch approach provides an increasing confidence in the simulation estimate of this measure. Qualitatively, because average observable values for $\mathcal{G}_{\mathbf{C}_2}$ do not match those of G_0, we have additional confidence in the interpretation that correlations in citation activities does not suffice to explain cyclic citation patterns.

To get some insights on how the convergence process varies with input size, we implement the algorithm on smaller samples of this dataset: the first 20,000 and 10,000 pages. Corresponding results are gathered on Table 3, providing information about computational requirements in the various cases[5]. As will also be the case in the next examples, it seems to be difficult to find any obvious relationship between input size and the number of trials necessary to observe the convergence.

---

[4] Available from http://www.barabasilab.com/rs-netdb.php

Table 3: Experimental values obtained for constraint $\mathbf{C}_2$ on different inputs (with N: number of nodes, M: number of arcs): minimum k measured to obtain a uniformly random sample, approximate amount of trials needed for convergence, maximum memory space used during the process.

Input	N	M	k threshold	approximate number of trials	memory used
WWW-50K	50,000	143,592	2	$\sim$ 1000m	13 MB
WWW-20K	20,000	63,224	2	$\sim$ 250m	8 MB
WWW-10K	10,000	36,970	2	$\sim$ 250m	5 MB

3.3. Constraint based on triangles

As said above, it is straightforward to apply the method with constraints on undirected graphs. $\mathbf{C}_3$, and $\mathbf{C}_4$ below, are of this kind.

$\mathbf{C}_3 = \mathbf{C}_3^{\emptyset} \cup \mathbf{C}_3^{+}$ is such that:

- $\mathbf{C}_3^{\emptyset}$*: the graph is undirected, with a fixed degree distribution, has no multiple edges nor self-loops*
- $\mathbf{C}_3^{+}$*: the number of (undirected) triangles remains the same.*

The interest of $\mathbf{C}_3$ can be illustrated in the case of a collaboration network. The amount of distinct motifs of size four will be our target observables. In that case, $\mathbf{C}_3$ practically aims at checking whether the size and shape of the close neighborhood of a scientist in this field is related to the cohesiveness between agents — that is, more precisely, to check how the tendency to do triangular interactions influences the number and connectedness of neighbors at distance 1 and 2.

G_0 is an undirected graph of collaborations between scientists extracted from the Anthropological Index Online database.[6] The dataset we use focuses on a specific subfield consisting of Scandinavian archeology-related papers published over the period 2000–2009: nodes are paper authors, links feature collaborations between authors in these papers. G_0 contains 273 individuals and 280 links.

Results of the corresponding exploration of the random graph space defined by $\mathbf{C}_3$ are gathered on Fig. 6 and Tab. 4 for motifs of size four, for which there is significant variation from G_0 for $k > 2$. More importantly, these diverging results do not appear when using $k = 2$, but only appear from $k > 2$, being then similar for all $k \in \{3, 4, 5, 6\}$. Thus, the usual *S&H* method — unlike the generalized switching method with $k \geq 3$ — cannot be used to generate a uniformly random subset of $\mathcal{G}_{\mathbf{C}_3}$ on this particular dataset: the obtained sample would be significantly biased. In other words, only by going beyond $k = 2$ makes it possible to show that $\mathbf{C}_3$ is not sufficient to explain the particular shape of the neighborhood of these agents in this empirical network.

On Table 5 we gather results on the convergence process on larger collaboration databases extracted from the AIO database in other geographical area, namely the British Isles and the whole of Europe, over the same period of time. Qualitative results on the relationship between $\mathbf{C}_3$ and target observables hold, yet there is, again, no obvious relationship between convergence and input size & type.

3.4. Constraint based on connected components

Finally, $\mathbf{C}_4$ addresses the issue of connected components. $\mathbf{C}_4$ is such that:

---

[5] Computations have been made using a standard computer (2x2.33GHz processor, 2GB memory).

[6] Available from http://aio.anthropology.org.uk/aiosearch

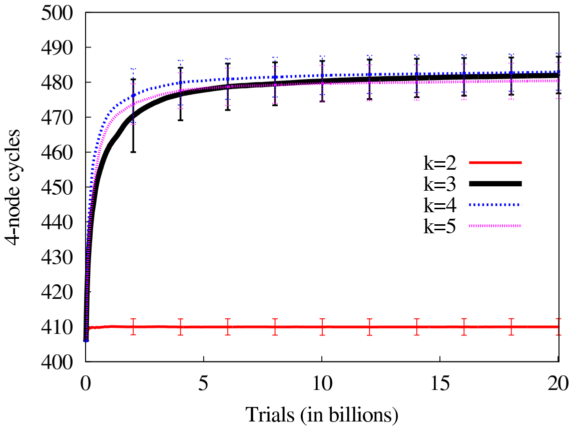

Figure 6: Cumulative mean number of 4-nodes cycles for $\mathbf{C}_3$.

Table 4: Mean number of motifs of size four after 20 simulations of 10 billion trials on G_0 from the AIO database.

Target observables	Starter G_0	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$
	2794	2799 ± 4	2907 ± 53	2933 ± 32	2942 ± 64	2894 ± 42
	406	410 ± 3	483 ± 6	483 ± 5	481 ± 5	482 ± 6
	730	734 ± 3	843 ± 9	841 ± 10	841 ± 6	840 ± 8
	108	108 ± 0	120 ± 2	120 ± 2	120 ± 2	119 ± 2
Successes (in millions)	-	79	166	96	34	8

Table 5: Experimental values obtained for constraint $\mathbf{C}_3$ on different inputs.

Input	N	M	k threshold	approximate number of trials	memory used
Scandinavia	273	280	3	$\sim 20,000$m	2 MB
British Isles	807	1020	2	$\sim 10,000$m	2 MB
Europe	12112	9090	2	$\sim 100,000$m	3 MB

- $\mathbf{C}_4^{\emptyset}$*: the graph is undirected, with a fixed degree distribution, has no multiple edges nor self-loops*
- $\mathbf{C}_4^{+}$*: distribution of the sizes of connected components remains the same*

G_0 is an undirected graph built from human metabolic pathways listed in the Biocyc database[7]: each node is a protein, and each link connects any two proteins involved in the same biochemical pathway. It features 679 nodes and 11 030 links. $\mathbf{C}_4$ aims at checking whether the existence of islands of pathways, as represented by connected components, is correlated with the presence of particular local, short-range interactions patterns between specific proteins.

Simulation results are featured on Tab. 6: averages of statistical variables obtained over corresponding explorations of $\mathcal{G}_{\mathbf{C}_4}$ do not match those of G_0. This suggests that $\mathbf{C}_4$ is not a possible explanation for the presence of 3- and 4-sized local patterns in this metabolic pathway network.

In this case, going beyond $k = 2$ did not yield any particular improvement on the random mixing process results, yet provided a stronger confidence on the random exploration of $\mathcal{G}_{\mathbf{C}_4}$.

Again, we run the algorithm on other network datasets: biochemical pathways of *Aquifex aeolicus* (denoted *aaeo*) and *Burkholderia pseudomallei* (*bpse*), see Table 7. Qualitative results hold too, while there is still no obvious relationship between convergence features and input size & type.

---

[7] http://www.biocyc.org

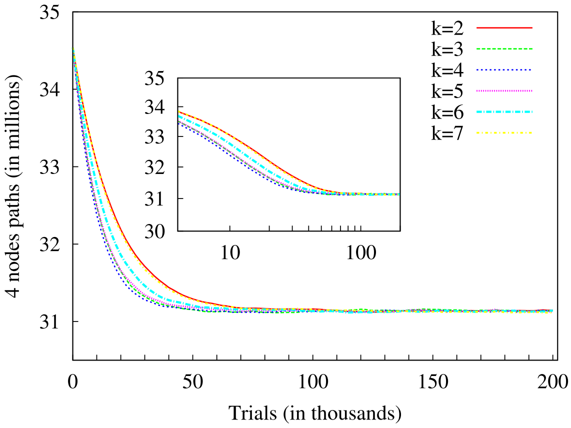

Figure 7: Number of undirected 4-nodes paths with respect to the number of k-switch trials ($k \in [2, 7]$) for $\mathbf{C}_4$.

Table 6: Mean number of patterns of size 3 and 4 on 50 simulations involving 200 000 trials on G_0 for 'Pathways'.

Target observables	Starter G_0	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$
△	$161.3 \cdot 10^3$	$51.7 \cdot 10^3$	$51.7 \cdot 10^3$	$51.7 \cdot 10^3$	$51.7 \cdot 10^3$	$51.7 \cdot 10^3$	$51.7 \cdot 10^3$
◈	$2070 \cdot 10^3$	$178 \cdot 10^3$	$178 \cdot 10^3$	$178 \cdot 10^3$	$178 \cdot 10^3$	$178 \cdot 10^3$	$177 \cdot 10^3$
⟨	$34.5 \cdot 10^6$	$31.1 \cdot 10^6$	$31.1 \cdot 10^6$	$31.1 \cdot 10^6$	$31.1 \cdot 10^6$	$31.1 \cdot 10^6$	$31.1 \cdot 10^6$
Successes	-	42,300	60,400	52,800	38,100	25,500	20,400

Table 7: Experimental values obtained for constraint $\mathbf{C}_4$ on different inputs.

Input	N	M	k threshold	approximate number of trials	memory used
aaeo	264	1,193	2	$\sim 20,000$	2 MB
Human	679	11,030	2	$\sim 200,000$	3 MB
bpse	1,447	20,620	2	$\sim 500,000$	6 MB

Conclusion

Pairwise edge swapping methods, such as *S&H*, are relevant to generate uniformly random samples of graphs in some simple cases, such as degree distributions. As constraints get stronger than just degree distributions, pairwise edge swaps may not be appropriate since the corresponding Markov graph is likely to be disconnected. We therefore proposed a higher-order switching method, denoted "k-edge switching", which makes it possible to tackle this issue by improving progressively the connectedness of the Markov graph of the corresponding walk.

While this approach guarantees that it is theoretically possible to navigate uniformly throughout the whole Markov graph for some value of k, for high values of k the process is likely to be empirically less and less practicable. As such, this approach nonetheless constitutes an easily implementable method to incrementally explore larger portions of the Markov graph; thereby obtaining an increasing, asymptotically certain confidence on the representativeness of the obtained sample. In particular, this method potentially generates random graphs for any type of constraint preserving degree distributions. It also makes it possible to incrementally check the robustness of results obtained using traditional edge swaps with $k = 2$ (which have no reason to yield valid results as such), thereby improving the confidence on the Markov graph exploration achieved by 2-switches.

Put simply, when average measurements on the reshuffled graphs tend to plateau for some successive values of k, we suggest that it is empirically sensible to assume that the walk covers a reasonably representative portion of the graph set $\mathcal{G}_{\mathbf{C}}$ — as such constituting a useful extension of edge swapping random graph generation approaches. In this respect, an interesting perspective for the present work would be to find classes of constraints **C** for which some low values of k guarantee the connectedness of the k-switch Markov graph.

Acknowledgements. We are grateful to Clémence Magnien and Fernando Peruani for interesting discussions, thank the anonymous reviewers for their constructive feedback, and acknowledge useful comments from Hugues Chaté and Niloy Ganguly. This work was partly supported by the Future and Emerging Technologies programme FP7-COSI-ICT of the European Commission through project QLectives (grant no.: `231200`) and by the French ANR through grant "Webfluence" #ANR-08-SYSC-009.

APPENDIX: Constraint checking algorithms and complexities

In this Appendix, we describe briefly some possible algorithms for implementing tests corresponding to the above-described constraints.

Constraints $\mathbf{C}_1$ *and* $\mathbf{C}_3$

Constraint $\mathbf{C}_1$ may be implemented by testing whether a switch trial creates as many triangles as it destroys. For each arc (a_i, b_i) involved in a switch trial, we may list which oriented triangles are being created and destroyed by looking for the out-neighbors of b_i which are also in-neighbors of a_i before and after the switch trial. The same goes for $\mathbf{C}_3$, except for the fact that triangles are not oriented.

A random link has a probability proportional to δ to be connected to a node of degree δ, and we have to go through the list of neighbors for each neighbor of b_i. The same goes with a_i, so that the comparison of both lists of neighbors has eventually an average complexity in $O(\overline{\delta}^4)$, where $\overline{\delta}$ is the mean degree. This yields an overall complexity in $O(nk\overline{\delta}^4)$, where n is the number of trials. Note that $\overline{\delta}$ is always equal to 1 in the case of $\mathbf{C}_1$.

Constraint $\mathbf{C}_2$

The test corresponding to this specific constraint can be implemented as follows: after storing at the beginning of the process the out-degree of each node, the user checks at each trial that for any couple of degrees (δ_1, δ_2), links whose extremities have degrees δ_1 and δ_2 are created and destroyed in equal numbers. This specific test can be done in constant time, yielding an overall time complexity of the algorithm in $O(nk)$.

Constraint $\mathbf{C}_4$

A very simple (yet not optimal) way to implement this constraint test is to check, for each link involved in a switch, the size of the connected component it belongs to before and after the switch. This can be done in $O(M)$ by using a breadth first search algorithm. This induces a global complexity in $O(nkM)$.

References

[1] Albert, R., Jeong, H., and Barabasi, A.-L. 1999. Diameter of the world wide web. *Nature 401*, 130–131.

[2] Artzy-Randrup, Y. and Stone, L. 2005. Generating uniformly distributed random networks. *PRE 72,* 5, 056708.

[3] Bansal, S., Khandelwal, S., and Meyers, L. 2008. Evolving Clustered Random Networks. *Arxiv preprint cs.DM/0808.0509*.

[4] Bender, E. and Canfield, E. 1978. The asymptotic number of labeled graphs with given degree sequences. *J. Combin. Theory Ser. A 24,* 3, 296–307.

[5] Colbourn, C. 1977. *Graph generation.* University of Waterloo.

[6] Coolen, A., De Martino, A., and Annibale, A. 2009. Constrained Markovian dynamics of random graphs. *Journal of Statistical Physics 136,* 6, 1035–1067.

[7] Cooper, C., Dyer, M., and Greenhill, C. 2006. Sampling regular graphs and a peer-to-peer network. *Combinatorics, Probability and Computing 16,* 04, 557–593.

[8] Eggleton, R. 1973. Graphic sequences and graphic polynomials: a report. *Infinite and Finite Sets 1*, 385–392.

[9] Feder, T., Guetz, A., Mihail, M., and Saberi, A. 2006. A local switch Markov chain on given degree graphs with application in connectivity of peer-to-peer networks. In *Proc. of FOCS*. Vol. 6. 69–76.

[10] Gkantsidis, C., Mihail, M., and Zegura, E. 2003. The markov chain simulation method for generating connected power law random graphs. In *Proc. 5th Workshop on Algorithm Engineering and Experiments (ALENEX).*

[11] Guillaume, J., Latapy, M., and Le-Blond, S. 2005. Statistical analysis of a P2P query graph based on degrees and their time-evolution. *Distributed Computing-IWDC 2004*, 439–465.

[12] Guruswami, V. 2000. Rapidly mixing markov chains: A comparison of techniques. MIT Laboratory for Computer Science. Available on cs.washington.edu/homes/venkat/pubs/papers.html.

[13] Kannan, R., Tetali, P., and Vempala, S. 1997. Simple Markov-chain algorithms for generating bipartite graphs and tournaments. In *Proceedings of the eighth annual ACM-SIAM symposium on Discrete algorithms.* Society for Industrial and Applied Mathematics Philadelphia, PA, USA, 193–200.

[14] Mahadevan, P., Krioukov, D., Fall, K., and Vahdat, A. 2006. Systematic topology analysis and generation using degree correlations. In *Proc. SIGCOMM'06.* ACM.

[15] Miklós, I. and Podani, J. 2004. Randomization of presence-absence matrices: comments and new algorithms. *Ecology Archives 85,* 1, 86–92. Appendix A available on http://esapubs.org/archive/ecol/E085/001/appendix-A.htm.

[16] Milo, R., Kashtan, N., Itzkovitz, S., Newman, M., and Alon, U. 2003. On the uniform generation of random graphs with prescribed degree sequences. *Arxiv preprint cond-mat/0312.028*.

[17] Newman, M. 2004. Coauthorship networks and patterns of scientific collaboration. *Proceedings of the National Academy of Sciences of the United States of America 101,* Suppl 1, 5200.

[18] Ralaivola, L., Swamidass, S., Saigo, H., and Baldi, P. 2005. Graph kernels for chemical informatics. *Neural Networks 18,* 8, 1093–1110.

[19] Rao, A., Jana, R., and Bandyopadhyay, S. 1996. A Markov chain Monte Carlo method for generating random (0, 1)-matrices with given marginals. *Sankhyā: The Indian Journal of Statistics, Series A*, 225–242.

[20] Roberts, J. 2000. Simple methods for simulating sociomatrices with given marginal totals. *Social Networks 22,* 3, 273–283.

[21] Sinclair, A. 1993. *Algorithms for random generation and counting: a Markov chain approach.* Springer.

[22] Stauffer, A. and Barbosa, V. 2005. A study of the edge-switching Markov-chain method for the generation of random graphs. *Arxiv preprint cs.DM/0512.105*.

[23] Taylor, R. 1980. Constrained switchings in graphs. *Combinatorial Mathematics 8*, 314—336.

[24] Taylor, R. 1982. Switchings constrained to 2-connectivity in simple graphs. *SIAM Journal on Algebraic and Discrete Methods 3*, 114.

[25] Viger, F. and Latapy, M. 2005. Efficient and simple generation of random simple connected graphs with prescribed degree sequence. *Lecture Notes in Computer Science 3595*, 440.