Color-Swap models and other stochastic processes for non-growing scale-free networks

Tomas Hruz
Institute of Theoretical Computer Science, ETH Zurich.

Madhuresh Agrawal
Indian Institute of Technology, Kanpur.

Michal Natora
Institute for Software Engineering and Theoretical Computer Science, Berlin Institute of Technology.

(Dated: October 2007)

We present an experimental work and a concept based on complete, edge-colored graphs which goes towards the development of a non-growing scale-free process which would lack the disadvantages of the processes known so far i.e. a process that would i) stay in the class of simple graphs, ii) reach a scale-free degree distribution independently of the initial conditions, iii) have a constant number of edges and iv) that would not condensate to a state where a small fraction of nodes posses all edges between them. Within the concept of edge-colored graphs the interactions (like rewiring, edge creation and deletion) occur through edge color exchange. A particular instance of such a process called Color-Swap improves on the above problems, however it comes with a higher volatility of edges for particular nodes.

PACS numbers: 89.75.Hc, 89.75.Fb, 05.65.+b

I. INTRODUCTION

Non-growing scale-free networks represent an interesting class of complex networks. Compared to growing networks, which have been studied in many different scenarios with great success [1, 3, 8], they aim at modeling situations where a complex topology different from purely random is emerging but no growth process is present. There are many examples, mainly in biology (see [9] and the references therein), where instead of growth one can observe a self-organizing stochastic process which moves some network connections to different nodes, and the number of nodes and edges stays statistically bounded within some interval.

Although various aspects of the dynamic behavior of non-growing (equilibrium) networks were well studied [4] and several stochastic process candidates have been proposed, they provide the scale-free property only under special conditions. The discrepancies between real networks and most model designs can be summarized as follows: (i) Multigraph: the processes in [4, 9] allow multiple edges and/or self-loops between the nodes, whereas in most real networks this phenomena does not exist. (ii) Structural constrains: the stochastic process defined in [9] reveals very interesting aspects, however the scale-free character is a result of structural constrains and special initial conditions. (iii) Connection (edge) growth: the process defined in [7] would reach scale-free state, but the number of edges grows considerably before it stabilizes, and the network is scale-free only in certain range of node degree. (iv) Condensation: the networks go through a scale-free state but later condensate to a state where a small fraction of nodes posses all the edges [4].

In the present paper we summarize an experimental work aimed at developing a process which would lack the disadvantages of the processes known so far i.e. a process that would

(i) stay in the class of simple graphs - a process without multiple edges between the nodes and without self-loops.

(ii) reach a state with scale-free degree distribution independently of the process starting configuration.

(iii) have a constant number of edges and that

(iv) would not condensate to a state where a small fraction of nodes posses all edges.

As we show in the next sections, condensation can be avoided if a model of external force is included. However, we concentrate on model design, which would not condensate even without an external force. This means that the network evolution rules contain only interactions between the network elements explainable from the local features.

The paper is organized as follows. In the next section we describe computational experiments in a uniform way providing the process description, experimental results and short discussion. We start with known processes published elsewhere, and the section is concluded with a new process design method called Color-Swap. In the concluding section we discuss further possibilities stemming from our simulations.

II. SIMULATIONS

In this section we discuss in detail various processes which have been proposed to generate scale-free graphs. We give the complete process description followed by our simulation results. We also justify the problems associated with each process and finally present a process *Color-Swap* which tries to overcome most of the drawbacks with other approaches.

A. SESP

Consider as a starting point a simple graph $G(E, V)$ with edge set E, $|E| = L$ and vertex set V, $|V| = N$. The *Simple Edge Selection Process* (abbreviated as SESP) is defined by the following steps which are supposed to occur in one discrete time unit. The process is well known (see for example [4]), and its asymptotic dynamics was studied in considerable detail.

1. An edge, called E_i is selected uniformly randomly.
2. An end vertex, called V_i, of E_i is selected uniformly randomly. The other end vertex will be called V_j.
3. A vertex, called V_l, is selected with a probability proportional to $f(k)$ i.e. with probability $\frac{f(k)}{N\langle f\rangle}$ where k is the degree of V_l and $\langle f\rangle$ is the mean value of f, $\langle f\rangle \equiv \sum_s f(s)P(s)$.
4. The edge E_i is rewired from V_i to V_l.

The process keeps the number of nodes and edges in the network constant, and the function $f(k)$ models preferential attachment; in particular if $f(k) = k$, the nodes in step 3 are selected with higher probability if they have higher degree. However, the process generates multiple connections and self-loops, which means that it does not preserve the simple graph network structure. Multiple connections are created, because the edge is rewired irrespective of already existing connections between V_j and V_l. On the other hand, self-loops emerge when vertex V_l is the same vertex as V_j.

We simulated the process with $f(k) = k$ under different network dimensions and initial conditions and observed that it ends up in condensation with all edges confining to one single vertex. (see Fig. 2). This can be understood qualitatively with the following argument. A vertex with degree 0 can never be selected as (a) random chosen end (like V_i or V_j) of a randomly selected edge E_i (b) preferentially chosen vertex V_l since $f(0) = 0$. Hence once a vertex becomes isolated in the SESP process it remains isolated throughout and the number of such isolated vertices keeps on increasing with the process ending up in condensation after sufficiently large number of rewiring operations.

One interesting possibility how to stop the condensation is to include a model of external force, that would keep the low degree nodes gaining edges. To achieve this effect a nonlinear function f(k) can be used. We simulated the simplest possibility using $f(k) = k, k > 0$ and $f(0) = \alpha$ to give some finite positive selection preference to isolated nodes. On simulating for large networks (see Fig. 3) we observed that SESP does not condensate anymore. Also the number of isolated nodes was quite acceptable. But because of a slow convergence to equilibrium, we needed to simulate it for small networks (see Fig. 4) to achieve a sufficiently high number of rewiring operations compared to the size of the network. We observed that for small networks too, the process does not condensate but the number of isolated nodes is quite high. Moreover, the process builds many small stars coming from nodes with very low degree, which are disconnected from the rest of the network. Therefore a good model of external force, which would stop the condensation without disconnecting the network needs more research.

B. SGESP

The natural extension of the SESP process, which would keep the process within the class of simple graphs is to introduce a test whether there is an edge between the target node, and if one is detected, the rewiring would not take place. The process called *Simple Graph Edge Selection Process* (abbreviated as SGESP) can be defined as follows (see also Fig. 1):

1. An edge, called E_i, is selected uniformly randomly.

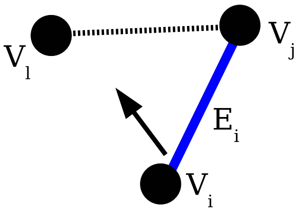

FIG. 1: (color online) The SESP process unconditionally rewires a connection from V_i to V_l. In contrast, according to the SGESP process the rewiring only occurs if V_l is not an end vertex of edge E_i and if V_l is not directly connected by an edge to V_j.

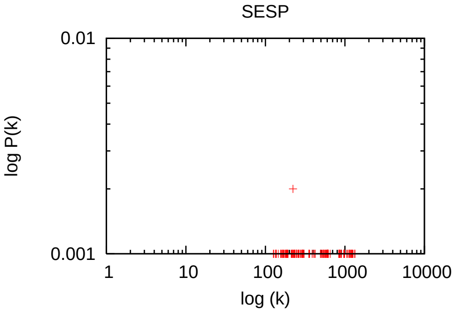

FIG. 2: (color online)Simulation experiment for a network with N=1000, L=2000. The x-axis represents degree values and the y-axis the values of $P(k)$, both axes are in logarithmic scale. The simulation time was 2.10^6 iterations. The figure shows the simulation results for $f(k) = k$. To obtain the resulting graph a time averaging over 15 instances separated with 10^4 states was computed. The SESP process condensates and all edges are rewired to a few nodes (finally to only one node) as self-connections.

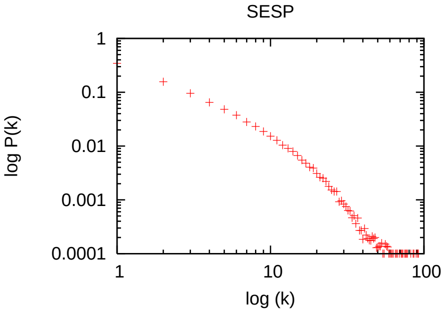

FIG. 3: (color online)Simulation experiment for a network with N=10000, L=20000. The x-axis represents degree values and the y-axis the values of $P(k)$, both axes are in logarithmic scale. The processes were simulated over 2.10^7 iterations to the equilibrium state. The figure shows the simulation results for the SESP process with $f(k) = k$ but $f(0) = \alpha$ with $\alpha = 5$. To obtain the resulting graph a time averaging over 15 instances separated with 10^5 states was computed. Introducing α avoids the condensation of SESP process and also in case of large networks $P(0) = 0.075$ (i.e. the number of isolated nodes), which is quite acceptable.

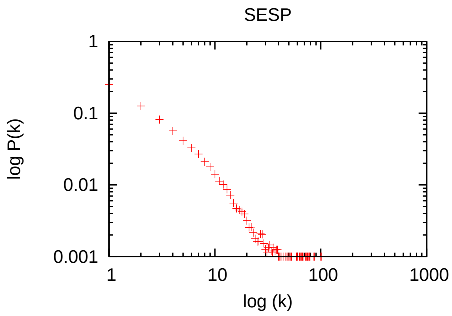

FIG. 4: (color online)Simulation experiment for a network with N=1000, L=2000. The meaning of the axes is the same as in Figure 2 and the same averaging procedure was used. The figure shows the simulation results for the SESP process with $f(k) = k$ but $f(0) = \alpha$ with $\alpha = 5$. Though introducing α helps in avoiding the condensation of SESP process, the number of isolated nodes i.e $P(0) = 0.21$ is quite high.

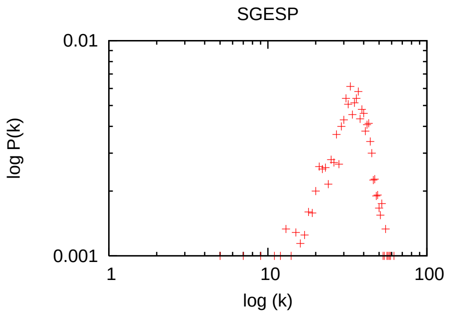

FIG. 5: (color online)Simulation experiment for a network with N=1000, L=2000. The meaning of the axes is the same as in Figure 2 and the same averaging procedure was used. The figure shows the simulation results for the SGESP process with $f(k) = \frac{k}{N\langle f \rangle}$. The SGESP process condensates to a highly interconnected kernel.

2. An end vertex, called V_i, of E_i is selected uniformly randomly. The other end vertex will be called V_j.
3. A vertex, called V_l, is selected with a probability proportional to $f(k)$ as in step 3 of the SESP process.
   (a) Check if V_l is an end vertex of E_i or if V_l is directly connected to V_j. If so, skip the next step.
4. The edge E_i is rewired from V_i to V_l.

The only difference to the SESP process is the test in the step 3a. However, to express analytically what this test means is difficult if not impossible in the full generality as the study in [6] shows. Therein, we introduced the process to study in detail the possibilities of SESP modifications that would keep the process in the class of simple graphs.

We simulated the process with $f(k) = k$ under different network dimensions and initial conditions and observed that it ends up in a highly interconnected kernel - a certain form of condensation. The experimental results for the range of parameters in Figure 5 also point to the analysis in [5], especially in relation to the analysis of the dense network kernel.

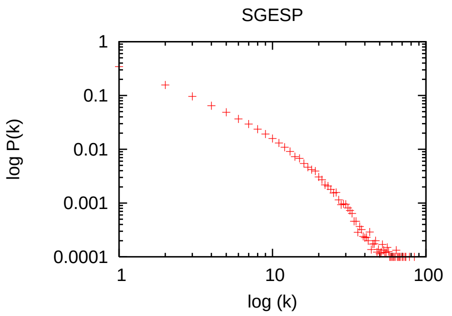

FIG. 6: (color online)Simulation experiment for a network with N=10000, L=20000. The meaning of the axes is the same as in Figure 3 and the same averaging procedure was used. The figure shows the simulation results for the SGESP process with $f(k) = k$ but $f(0) = \alpha$ with $\alpha = 5$. Introducing α avoids the condensation of SESP process and also $P(0) = 0.07$ which is quite acceptable.

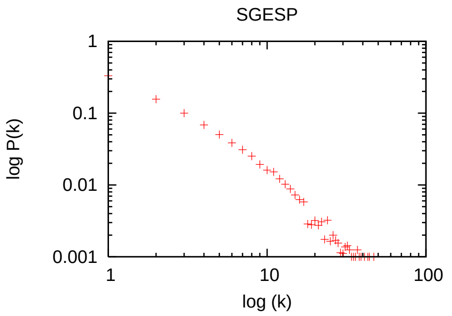

FIG. 7: (color online)Simulation experiment for a network with N=1000, L=2000. The meaning of the axes is the same as in Figure 2 and the same averaging procedure was used. The figure shows the simulation results for the SGESP process with $f(k) = k$ but $f(0) = \alpha$ with $\alpha = 5$. For SGESP, even for small networks $P(0) = 0.07$ which is quite acceptable.

Similarly as with SESP we introduced an external force with modified $f(k)$ i.e $f(k) = k, k > 0$ and $f(0) = \alpha$ to give some finite positive selection preference to isolated nodes too. On simulating for both large networks (see Fig. 6) as well as small networks (see Fig. 7) we observed that SGESP does not condensate anymore. Also the number of isolated nodes was quite acceptable in both cases. However, the connectivity problems as with SESP would need more research.

C. VADE

VADE, which stands for "Vertex based Addition and Deletion of Edges" was introduced in [6] and it aims at approximating the edge selection rule of SGESP with vertex selection rules. The VADE process is defined as (See Fig. 8)

1. With probability q do the following:
   (a) choose a vertex V_i with a probability proportional to $f_1(k)$;

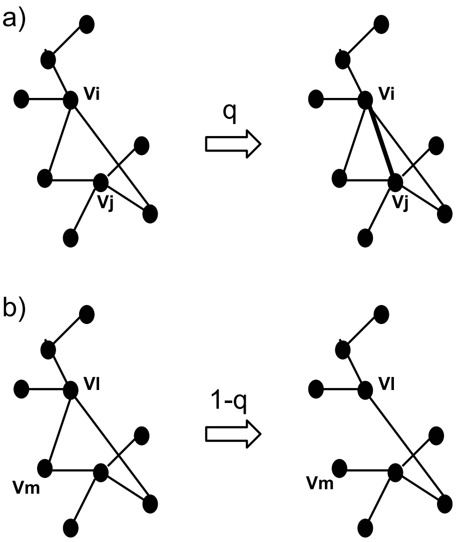

FIG. 8: Vade Process: (a) V_i and V_j are chosen based on the function $f_1(k)$ and $f_2(k)$ respectively. An edge is then introduced between V_i and V_j. (b) V_l and V_m are chosen based on the function $g_1(k)$ and $g_2(k)$ respectively. Edge $V_l - V_m$ is then deleted. Step (a) is performed with probability q while step (b) is performed with probability $1 - q$.

(b) choose a vertex V_j with a probability proportional to $f_2(k)$;

(c) if $V_i = V_j$ or if V_i *is* directly connected to V_j, skip the next step;

(d) add an edge between V_i and V_j.

2. With probability $1 - q$ do the following:

   (a) choose a vertex V_l with a probability proportional to $g_1(k)$;

   (b) Choose a vertex V_m with a probability proportional to $g_2(k)$;

   (c) if $V_l = V_m$ or if V_l is *not* directly connected to V_m, skip the next step;

   (d) delete the edge between V_l and V_m.

The VADE process is similar to the SGESP process in several ways. First, because it contains preferential selection parameters (f_1, f_2, g_1, g_2), in addition, the number of edges stays stable in a narrow interval (if the parameter q is chosen appropriately) approximating the constant number of edges in the SGESP process, and it also preserves the simple graph structure. The preferential selection parameters allow a large flexibility in the process behavior and they can be set to generate a degree distribution which is similar to the one generated by the SGESP process (Compare Fig. 9 and Fig. 7), however similarly as with SESP and SGESP the force term $f(0) > 0$ is necessary to achieve a stable scale-free state.

In Figure 10 the number of edges for the VADE process is plotted. The experiments show that if the parameter q of VADE process is correctly set, the number of edges will stay bounded in a very narrow interval around the initial number of edges.

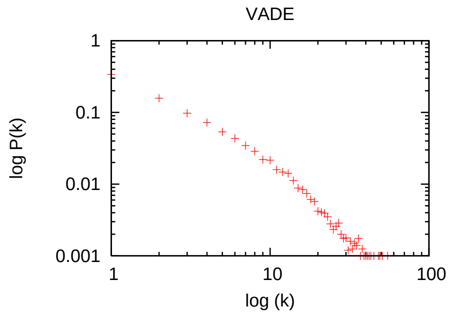

FIG. 9: (color online) Simulation experiment for VADE process. The meaning of the axes is the same as in Figure 2 and the same averaging procedure was used. The VADE process was simulated with the parameters: $N = 1000, q = 0.004, f_1(k) = k, f_1(0) = 1000, f_2(k) = k, g_1 = 1, g_2 = 1$. The initial number of edges was $L = 2000$.

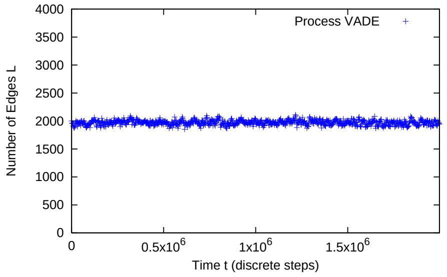

FIG. 10: (color online) The figure shows the simulation results for the VADE process with the same parameters as in Figure 9. The x-axis represents the discrete time and the y-axis the number of edges, both axes are in linear scale. The simulation experiments show that number of edges quickly becomes stable around $2000 = 1L$.

D. Merge Regenerate

The Merge-Regenerate model [7] is based on *merging* and *regeneration* of nodes. From the process definition it follows that the process would stay in the class of simple graphs, however the bound on the number of edges is not provided. As our simulations under different network dimensions and different initial conditions have shown the number of edges stabilizes but on the much higher number as was the initial condition. For example having the parameters $N = 1000, L = 1000, \langle r \rangle = 6$ the final number of edges is about 12000 i.e $12L$.

E. Park Lai Ye

The process defined in [9] achieves the scale-free character because of specific initial conditions and because of structural constrains which do not allow to create isolated nodes and multiple-edges. However, self-loops can occur therefore the process would not stay in the class of simple graphs. As the experiments show the process attains quickly the scale free state and stays stable with constant number of edges.

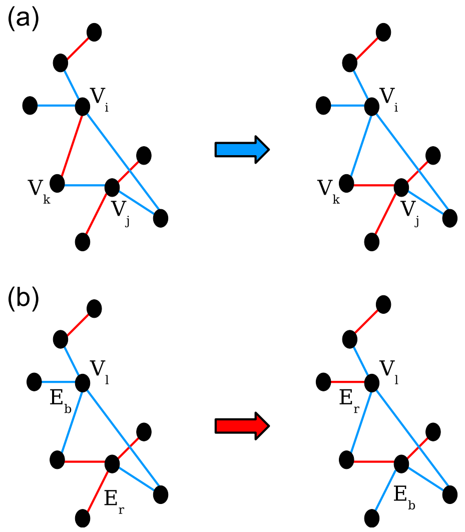

FIG. 11: (color online) Color-Swap process: (a) Blue Rewiring Phase: V_i and V_j are two randomly selected vertices. $V_j - V_k$ is swapped with $V_i - V_k$. The other blueNbr of V_j already has an edge with V_i and thus the swap won't bring any change in its case. (b) Red Rewiring Phase: V_l is selected preferentially based on its *blueDegree*. E_b is a uniformly randomly selected blue edge incident on V_l. It is swapped with a uniformly randomly selected red edge E_r. Black edges have not been shown in the figure

F. Color-Swap Models

The Color-swap class of processes is defined on complete edge-colored simple graph G. The edge rewiring, creation and deletion are implemented using color exchange between the graph edges.

1. *The Graph Representation for Color-Swap process*

Let $G(V, L_{blue} \cup L_{red} \cup L_{black}) = K^{|V|}$ be a simple edge-colored graph with 3 colors - blue, red and black. We consider the following notation and color semantics:

- L_{blue} equals number of Blue Edges - These are the true edges i.e the final degree distribution of the graph depends on their distribution only.
- L_{red} equals number of Red Edges - Red edges represent potential or different connections, they might be useful in modeling networks with various connection types.
- $L_{black} = \binom{N}{2} - L_{blue} - L_{red}$ equals number of black edges - These are non edges and their role is to occupy the vacant sites, to ensure that graph is fully connected (in terms of these three kinds of edges).

2. *Color-Swap Process Description Notation*

- *blueDegree*: blueDegree of a vertex V_i is defined as number of blue edges incident on it.
- *redDegree*: redDegree of a vertex V_i is defined as number of red edges incident on it.
- *blueNbr*: V_j is called blueNbr of a vertex V_i if $\exists$ a blue edge V_j-V_i
- *redNbr*: V_j is called redNbr of a vertex V_i if $\exists$ a red edge V_j-V_i
- *swap*: Swapping an edge E_i with some edge E_j just means interchanging their colors.

Let us describe the complete details of the process (See Fig. 11)

1. *Initialization* : Initialize the Graph $G(V, L_{blue} \cup L_{red} \cup L_{black}) = K^{|V|}$ as a simple graph.
2. *Blue Rewiring Phase*: Randomly select two nodes V_i and $V_j \neq V_i$. For each blueNbr V_k of V_j swap edge V_j-V_k with V_k-V_i.
3. *Red Rewiring Phase*: Select a vertex V_l using $f(k_{blue}) = \frac{k_{blue}}{2L_{blue}}$ where k_{blue} is blueDegree of the vertex V_l. Now uniformly randomly select a blue edge E_b incident on it and swap it with uniformly randomly chosen red edge E_r.
4. Repeat Steps 2-3 till the equilibrium is obtained

We simulated the process for various network dimensions and initial conditions and have observed that the process always generates scale-free graphs (See Fig. 12, Fig. 13 and Fig. 14).

The color-swap process improves on the problems (i) - (iv) mentioned in the introduction. In particular, due to the mechanism of color exchange, it stays naturally in the class of simple graphs and it controls the growth of hubs leading to a scale-free structure. The number of edges stays constant because of the process design. The color-swap process generates scale-free networks independent of the initial condition and the convergence speed to the scale-free equilibrium state is very fast.

However, the process comes with a higher volatility of edges for particular nodes. Also the fraction of nodes with blue degree 0 is higher than for the processes in [7, 9], but the process does not condensate. If models of real networks with high local edge stability are sought more research on edge-colored models would be needed. On the other hand this class of models is flexible to express complex interactions between the network nodes.

It should be noted that the color-swap model omits condensation (and reaches a scale-free equilibrium state) without using an external force, i.e. the selection function $f(k_{blue})$ is strictly linearly preferential.

III. SUMMARY AND CONCLUSION

We presented some known processes from the literature that generate non-growing scale-free networks under certain conditions. One can say that all of them have some disadvantageous properties, namely (i)-(iv) described in the introduction. A summary of our simulation results for all process can be found in Table I.

We also presented the Color-swap process which omits the mentioned drawbacks, and at the same time introduces an interesting concept of colored edges. Of course the question

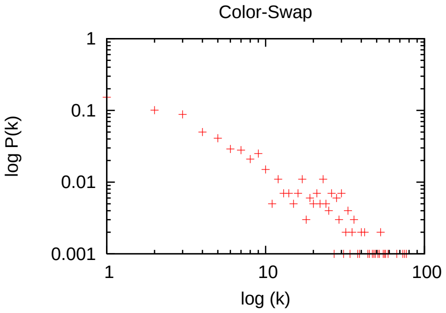

FIG. 12: (color online) Simulation experiment for Color-Swap process. The meaning of the axes is the same as in Figure 2 and no averaging has been done. The Color-Swap process was simulated with the parameters: $N = 1000, L_{blue} = 3000, L_{red} = 3000$.

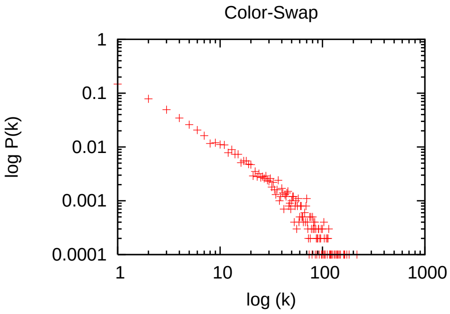

FIG. 13: (color online) Simulation experiment for Color-Swap process. The meaning of the axes is the same as in Figure 2 and no averaging has been done. The Color-Swap process was simulated with the parameters: $N = 10000, L_{blue} = 30000, L_{red} = 30000$.

arises if colored edges should be regarded as a new constraint or as an additional feature. Although we do not present any specific examples of existing networks which are explicitly modeled by the Color-swap process, we believe that rewiring processes based on colored edges might indeed be appropriate for modeling some real networks in which various connection types are present. For example one can imagine to represent the real connections of a technological network (such as the Internet) by blue edges, whereas the back-up links are modeled by the potential/red edges. The color models can also be generalized for more than three colors and it is imaginable to use them in order to model different types of protein interactions in biological networks.

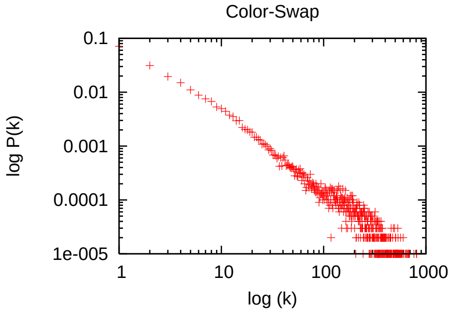

FIG. 14: (color online) Simulation experiment for Color-Swap process. The meaning of the axes is the same as in Figure 2 and no averaging has been done. The Color-Swap process was simulated with the parameters: $N = 100000, L_{blue} = 300000, L_{red} = 300000$.

[1] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Rev. Modern Phys.*, 74:47–97, 2002.

[2] M. Baiesi and S. S. Manna. Scale-free networks from a hamiltonian dynamics. *Physical Review E*, 43(047103), 2003.

[3] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(509), 1999.

[4] S. N. Dorogovtsev and J. F. F. Mendes. *Evolution of networks*. Oxford University Press, 2003.

[5] S. N. Dorogovtsev, A. M. Povolotsky, and A. N. Samukhin. Organization of complex networks without multiple connections. *arXiv.org*, (cond-mat/0505193), 2005.

[6] Tomas Hruz, Michal Natora, and Madhuresh Agrawal. Higher-order distributions and nongrowing complex networks without multiple connections. *Physical Review E*, 77(046101), 2008.

[7] B.J. Kim, A. Trusina, P. Minnhagen, and K. Sneppen. Self organized scale-free networks from merging and regeneration. *Eur. Phys. J. B*, 43:369–372, 2005.

[8] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.

[9] Kwangho Park, Ying-Cheng Lai, and Nong Ye. Self-organized scale-free networks. *Physical Review E*, 72(026131), 2005.

	SESP	SGESP	VADE	Merge-Regenerate	Park-Lai-Ye	Color-Swap
Simple Graph	No	Yes	Yes	Yes	No	Yes
Isolated Nodes	Yes	Yes	Yes	No	No	Yes
Edges	Constant	Constant	Bounded	Too many	Constant	Constant
Large Networks	Yes	Yes	Yes	Yes	Yes	Yes
Small Networks	Too many isolated nodes	Yes	Yes	Yes	No	Yes
Rate of attaining equilibrium	Slow	Slow	Slow	Fast	Fast	Fast

TABLE I: Comparison of processes on selected parameters. The row called "Simple Graph" describes whether the process stays in the class of simple graphs, the row called "Isolated Nodes" describes whether the process creates more nodes with degree 0 than was the initial condition, the row "Edges" tells whether the number of edges is bounded or grows. The rows "Large Networks" and "Small Networks" indicate whether the network reaches the scale-free state for very small ($N \sim 1000$) and very large ($N \gg 1000$), and the last row describes the convergence speed.