

## Research Article

# An Iterative Scheme to Compute Size Probabilities in Random Graphs and Branching Processes

Paolo Serafini 

*Department of Mathematics, Computer Science, and Physics, University of Udine, Udine, Italy*

Correspondence should be addressed to Paolo Serafini; [paolo.serafini@uniud.it](mailto:paolo.serafini@uniud.it)

Received 25 September 2017; Revised 29 January 2018; Accepted 11 February 2018; Published 1 April 2018

Academic Editor: Fabrizio Riguzzi

Copyright © 2018 Paolo Serafini. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We deal with a functional equation that plays an important role in random graphs and in branching processes. In branching processes, the functional equation relates offspring probabilities to population size probabilities, while in random graph it relates degree probabilities to small component size probabilities. We present an iterative scheme that allows computing the size probabilities numerically. It is also theoretically possible to invert the iteration, although this inverse iteration is numerically unstable.

## 1. Introduction

Let  $G(x)$  and  $H(x)$  be two probability generating functions that are linked through the functional equation

$$H(x) = xG(H(x)). \quad (1)$$

Functions of this type occur in branching processes and in random graphs [1–6]. In branching processes,  $G(x)$  represents the probabilities of new offspring from a member of the population and  $H(x)$  represents the population size probabilities. In the configuration model [4] of random graphs,  $G(x)$  represents the excess degree probabilities of a vertex in small components and  $H(x)$  represents the small component size probabilities. Note that, in both cases,  $H(x)$  can be a defective generating function; that is,  $H(1) < 1$ .

Usually  $G(x)$  is given and  $H(x)$  has to be computed. Only in rare cases is it possible to find an explicit analytic expression of  $H(x)$ . However, a numerical iteration to compute the coefficients of  $H(x)$  is possible. To the best of our knowledge, such a question has not been investigated and it seems that the iteration we propose in this paper is new.

Interestingly enough, this iteration can be inverted; that is, from the size distribution, we can infer the degree probabilities. We present also this inverse iteration, although it has to be remarked that the inverse iteration is numerically unstable.

The paper is organized as follows. In Section 2, we provide the mathematical background by referring to the case of random graphs. Then, in Section 3, we present the main result, that is, the iteration to compute the size probabilities of the small components of the graph. The possibility of inverting this computation is presented in Section 4. Then, in Section 5, we point out how the same iteration can be used for a branching process. Some conclusions are presented in Section 6.

## 2. Mathematical Background

We first present our result by explicitly referring to random graphs in the configuration model for which the picture is more complex. In a later section, we show how to relate the iteration to branching processes. Hence, all definitions in this section and in Sections 3 and 4 are related to random graphs.

A random graph has assigned degree probabilities  $p_h$ ,  $h = 0, 1, \dots$ , and  $p_h$  is the probability that a randomly selected vertex has degree  $h$ . We recall that the degree of a vertex is the number of vertices adjacent to it. The study of random graphs through generating functions is asymptotic; that is, it assumes an infinite number of vertices. Let  $G_0(x)$  be the probability generating function of the degree distribution; that is,

$$G_0(x) = \sum_{h \geq 0} p_h x^h. \quad (2)$$

Let  $d = G_0'(1)$  be the average degree and

$$G_1(x) = \frac{G_0'(x)}{d} = \sum_{h \geq 0} q_h x^h, \quad (3)$$

where clearly

$$q_h = \frac{(h+1)p_{h+1}}{d}. \quad (4)$$

The  $q_h$  values are known as *excess degree probabilities*. Let  $H_0(x)$  and  $H_1(x)$  be two generating functions that can be expressed as power series as

$$\begin{aligned} H_0(x) &= \sum_{k \geq 1} s_k x^k, \\ H_1(x) &= \sum_{k \geq 1} r_k x^k, \end{aligned} \quad (5)$$

and they are defined by the equations

$$\begin{aligned} H_0(x) &= xG_0(H_1(x)), \\ H_1(x) &= xG_1(H_1(x)). \end{aligned} \quad (6)$$

Our aim is to compute the coefficients  $s_k$  and  $r_k$ .

The motivation for the generating functions  $H_0(x)$  and  $H_1(x)$  derives from the analysis of the asymptotic properties of the random graph in the configuration model. If the graph is sufficiently dense, it exhibits the so-called *giant component*, that is, a connected component whose size asymptotically goes to infinity. The giant component, if present, is unique. The rest of the graph consists of an infinite number of finite trees, the so-called *small components* (see [3], among many possible references).

It can be shown that if  $p_h$  is the probability that a randomly chosen vertex (in the whole graph) has degree  $h$ , then  $s_k$  is the probability that a randomly chosen vertex belongs to a small component of size  $k$ , and  $r_k$  is the probability that, after choosing a random vertex  $i$  of degree at least one and then a random vertex  $j$  adjacent to  $i$ , the vertex  $j$  belongs to a small component of size  $k$  after removing the edge  $\{i, j\}$ .

If the giant component is present, the conditional probability  $\hat{p}_h$  of choosing in the small components a vertex of degree  $h$  is different from  $p_h$  and similarly for the excess degree probability  $\hat{q}_h$ . It can be shown that

$$\hat{p}_h = \frac{u^h}{v} p_h, \quad (7)$$

$$\hat{q}_h = u^{h-1} q_h,$$

where  $u$  is the solution of  $u = G_1(u)$  and  $v = G_0(u)$  is the fraction of vertices in the small components. We can briefly justify (7) by using Bayes' formula:

$$\hat{p}_h = \frac{\Pr\{S \mid D_h\} p_h}{\Pr\{S\}}, \quad (8)$$

with  $S$  being the random event of choosing a vertex in a small component and  $D_h$  being the random event of choosing a vertex of degree  $h$ . Clearly  $\Pr\{S \mid D_0\} = 1$  and consequently  $\hat{p}_0 = p_0/v$ . If  $h > 0$ ,  $\Pr\{S \mid D_h\}$  is the probability that *all* adjacent  $h$  vertices belong to a small component once we have removed the corresponding edges, and so its value is  $u^h$ . This explains the left expression in (7). To justify the right expression, we need to compute the average degree in the small components by taking the derivative of  $G_0(ux)/v$  (by using (7)) and computing it for  $x = 1$ ; that is,  $uG_0'(u)/v = udG_1(u)/v = u^2d/v$ . From this, we immediately get the expression at the right.

It turns out that using  $\hat{p}_h$  instead of  $p_h$  in the definition of  $G_0$  and  $G_1$  has the only effect of scaling the  $s_k$  values by the constant factor  $v$  and the  $r_k$  values by the constant factor  $u$ , which correspond to the conditional probability of choosing within the small components. In particular, we have  $H_0(1) = v$  and  $H_1(1) = u$  if we use  $p_h$  and  $q_h$  in the definition of  $G_0$  and  $G_1$ , respectively, whereas we have  $H_0(1) = 1$  and  $H_1(1) = 1$  if we use  $\hat{p}_h$  and  $\hat{q}_h$ .

We also define the probability  $t_k$  that a randomly selected small component has size  $k$ . Of course

$$\frac{s_k}{v} = \frac{kt_k}{\bar{t}}, \quad (9)$$

with  $\bar{t}$  being the average size of a small component. Here we have to discount  $s_k$  because the choice of small components necessarily conditions the choice within the small components.

### 3. Computing the Coefficients of $H_0(x)$ and $H_1(x)$

From the recursive equation,

$$H_1(x) = xG_1(H_1(x)) = x \sum_{h \geq 0} q_h H_1(x)^h, \quad (10)$$

and from

$$H_1(x) = \sum_{k \geq 1} r_k x^k \quad (11)$$

(necessarily  $r_0 = 0$ ), we derive

$$\begin{aligned} \sum_{k \geq 1} r_k x^k &= xq_0 + x \sum_{h \geq 1} q_h \left( \sum_{j \geq 1} r_j x^j \right)^h \\ &= xq_0 + \sum_{h \geq 1} q_h x^{h+1} \left( \sum_{j \geq 0} r_{j+1} x^j \right)^h, \end{aligned} \quad (12)$$

so that

$$\sum_{k \geq 1} r_k x^k = xq_0 + \sum_{h \geq 2} q_{h-1} x^h \left( \sum_{j \geq 0} r_{j+1} x^j \right)^{h-1}. \quad (13)$$

Let  $a_k^h$  be the coefficient of  $x^k$  in  $(\sum_{j \geq 0} r_{j+1} x^j)^h$ . Note that  $a_k^1 = r_{k+1}$  and in particular  $a_0^1 = r_1$ . From (13),

$$\begin{aligned} r_1 &= q_0, \\ r_k &= \sum_{h=2}^k q_{h-1} a_{k-h}^{h-1}, \quad k \geq 2. \end{aligned} \quad (14)$$

Hence, the computation of  $r_k$  requires the coefficients  $a_{k-2}^1, a_{k-3}^2, \dots, a_0^{k-1}$ . In turn, the computation of  $a_k^h$  requires the terms  $r_1, \dots, r_{k+1}$  and so to compute  $r_k$  we only need knowledge of  $r_1, \dots, r_{k-1}$ .

The recursion works as follows: initially  $r_1 = q_0$ , and then, for  $k = 2, 3, \dots$ , the following block is computed:

$$\begin{aligned} r_k &= \sum_{h=2}^k q_{h-1} a_{k-h}^{h-1}, \\ a_{k-1}^1 &= r_k, \\ a_{k-1}^h &= \sum_{j=0}^{k-1} a_j^{h-1} a_{k-1-j}^1 = \sum_{j=0}^{k-1} a_j^{h-1} r_{k-j}, \quad h = 2, \dots, k-1, \\ a_h^k &= \sum_{j=0}^h a_j^{k-1} r_{h+1-j} \quad h = 0, \dots, k, \end{aligned} \quad (15)$$

Note that  $a_0^k = a_0^{k-1} r_1 = a_0^{k-1} q_0$  and therefore  $a_0^k = q_0^k$ , where  $k$  is an upper index for  $a$  and a power exponent for  $q$ .

We also derive from

$$\begin{aligned} H_0(x) &= xG_0(H_1(x)), \\ H_0(x) &= \sum_{k \geq 1} s_k x^k \end{aligned} \quad (16)$$

the expression

$$\sum_{k \geq 1} s_k x^k = xp_0 + \sum_{h=2}^n p_{h-1} x^h \left( \sum_{j \geq 0} r_{j+1} x^j \right)^{h-1}, \quad (17)$$

so that

$$\begin{aligned} s_1 &= p_0, \\ s_k &= \sum_{h=2}^k p_{h-1} a_{k-h}^{h-1}, \quad k \geq 2. \end{aligned} \quad (18)$$

In this case, the computation is straightforward, since it involves all previously computed quantities.

Theoretically, the generating functions  $H_0(x)$  and  $H_1(x)$  involve an infinite series, but obviously only a finite number of coefficients can be computed. Hence, the computation has to be stopped after having computed the desired number of terms  $r_k$  and  $s_k$ . Since each term is computed only once and it is not the result of subsequent smaller and smaller additions, truncating the computation up to a certain index has no effect on the accuracy of the values we compute. In other words, if

we compute just a few terms, they are computed with the same accuracy as we had computed all coefficients.

It is clear from the definitions and the previous iteration that  $H_1(x)$  implies  $H_0(x)$ ; that is, once we know the  $r_k$  values, the  $s_k$  values are also known and implied by the  $r_k$  values. It is not difficult to see that the converse is also true. By differentiating (6) and using (3), we get

$$\begin{aligned} H_0'(x) &= \frac{H_0(x)}{x} + xdG_1(H_1(x))H_1'(x) \\ &= \frac{H_0(x)}{x} + dH_1(x)H_1'(x), \end{aligned} \quad (19)$$

and by integrating (19), we get

$$\frac{d}{2} H_1^2(x) = H_0(x) - \int_0^x \frac{H_0(\xi)}{\xi} d\xi, \quad (20)$$

that is,

$$\frac{d}{2} \left( \sum_{k \geq 1} r_k x^k \right)^2 = \sum_{k \geq 1} s_k x^k - \int_0^x \sum_{k \geq 1} s_k \xi^{k-1} d\xi, \quad (21)$$

which leads to the following identities term by term:

$$\frac{d}{2} \sum_{h=1}^{k-1} r_h r_{k-h} = s_k \left( 1 - \frac{1}{k} \right), \quad k \geq 2. \quad (22)$$

For  $k = 2$ , we get in particular

$$r_1 = \sqrt{\frac{s_2}{d}}, \quad (23)$$

and, for  $k > 2$ , we have

$$\frac{d}{2} \left( 2r_1 r_{k-1} + \sum_{h=2}^{k-2} r_h r_{k-h} \right) = s_k \left( 1 - \frac{1}{k} \right), \quad (24)$$

which allows writing

$$r_{k-1} = \frac{2(s_k/d)(1 - 1/k) - \sum_{h=2}^{k-2} r_h r_{k-h}}{2r_1}, \quad (25)$$

so that all  $r_k$  values can be recursively computed once we know  $d$ . We note that  $p_1 > 0$  implies that  $q_0 = r_1 > 0$ . Hence, the recursion is well defined if  $p_1 > 0$ , which is an almost necessary assumption if we investigate the presence of small components. Hence, knowledge of the  $s_k$  values implies knowledge of the  $r_k$  values.

If we do not know  $d$ , we may still compute  $d$  from the recursion. We first note that all  $r_k$  depend on  $d$  through the factor  $1/\sqrt{d}$ . Therefore, we initially guess the value  $d = 1$  and compute tentative values  $\tilde{r}_k$ . Since  $\sum_k r_k = u$ , we find the correct value for  $d$  as  $d = (\sum_k \tilde{r}_k / u)^2$  and so we have the correct values:

$$r_k = \frac{\tilde{r}_k}{\sum_k \tilde{r}_k} u. \quad (26)$$

#### 4. Inferring the Degree Probabilities from the Component Size Probabilities

We may also consider the inverse problem of finding  $G_0(x)$  and  $G_1(x)$  from  $H_0(x)$  and  $H_1(x)$ , that is, computing the degree distribution which gives rise to a particular small component size distribution. This problem presents interesting features. Arbitrary degree distributions of  $r_k$  and  $s_k$  may not be feasible; that is, there may be no degree distribution that can lead to those values.

Formally, the recursion can be easily inverted; that is, knowing the  $r_k$  values, we can compute the  $p_k$  values and  $q_k$  values. Indeed, from (14), we have

$$r_k = q_{k-1}a_0^{k-1} + \sum_{h=2}^{k-1} q_{h-1}a_{k-h}^{h-1}; \quad (27)$$

that is,

$$q_{k-1} = \frac{r_k - \sum_{h=2}^{k-1} q_{h-1}a_{k-h}^{h-1}}{a_0^{k-1}}. \quad (28)$$

Computing the  $a_h^k$  values is straightforward once we know the  $r_k$  values. From the  $q_k$  values, we easily deduce the  $p_k$  values, apart from the fact that  $p_0$  cannot be derived from the  $q_k$  values. However,  $p_0 = s_1$ , and so it is known a priori. Note also that  $r_1 > 0$  implies that  $a_0^{k-1} > 0$ .

There is, however, a subtle point to be settled. Let us assume that a giant component may be present but we do not know the  $u$  and  $v$  values. Then, it is simpler to work with the conditional probabilities within the small components. Starting from the (conditional)  $s_k$  probabilities, we compute the  $r_k$  values as explained in the previous section but by using the normalization  $\sum_k \tilde{r}_k = 1$ . This way, we actually compute  $\hat{q}_h$  and  $\hat{p}_h$ . Then, from (7), we get  $p_h$  and  $q_h$ . The unknowns  $u$  and  $v$  are computed by imposing  $\sum_h p_h = 1$  and  $\sum_h q_h = 1$ , which is equivalent to solving  $G_1(u^{-1}) = u^{-1}$  and  $G_0(u^{-1}) = v^{-1}$  with  $G_0$  and  $G_1$  defined on  $\hat{p}_h$  and  $\hat{q}_h$ .

However, the inverse recursion is numerically unstable, and, unless we use exact data, it can produce absurd outcomes, like probabilities outside the range  $[0, 1]$ . The reason of the instability is clear from (28), where we have a difference in the numerator and the denominator is getting smaller and smaller with  $q_0^k$ . As a simple exercise, suppose that we wonder which degree distribution gives rise to a size distribution of the small components of exponential type; that is,

$$t_k = (1 - \beta) \beta^{k-1}, \quad k \geq 1, \quad (29)$$

with  $0 < \beta < 1$ . Hence, we have  $\bar{t} = 1/(1 - \beta)$  and

$$s_k = \frac{kt_k}{\bar{t}} = k(1 - \beta)^2 \beta^{k-1}. \quad (30)$$

We remark that these  $s_k$  are conditional probabilities. Now we have to compute the  $r_k$  values from the  $s_k$  values. As explained in the previous section, we initially fix  $d = 1$  and compute from (25) the tentative values:

$$\tilde{r}_k = \sqrt{\frac{2}{\beta}} (1 - \beta) \beta^k, \quad (31)$$

for which  $\sum_k \tilde{r}_k = \sqrt{2\beta}$ . Hence,  $d = 2\beta$ , implying the correct values:

$$r_k = (1 - \beta) \beta^{k-1}. \quad (32)$$

If we carry out the computation in (28) symbolically, we get

$$\begin{aligned} \hat{q}_0 &= 1 - \beta, \\ \hat{q}_1 &= \beta, \\ \hat{q}_k &= 0, \quad k > 1, \end{aligned} \quad (33)$$

from which

$$\begin{aligned} \hat{p}_0 &= (1 - \beta)^2, \\ \hat{p}_1 &= 2(1 - \beta)\beta, \\ \hat{p}_2 &= \beta^2, \\ \hat{p}_k &= 0, \quad k > 2. \end{aligned} \quad (34)$$

From (7), we have

$$\begin{aligned} p_h &= \frac{v}{u^h} \hat{p}_h, \\ q_h &= \frac{1}{u^{h-1}} \hat{q}_h. \end{aligned} \quad (35)$$

By imposing  $\sum_h q_h = 1$ , we get

$$\begin{aligned} u(1 - \beta) + \beta &= 1 \implies \\ u &= 1, \end{aligned} \quad (36)$$

and also  $v = 1$ . Hence, there is no giant component in this case.

Now assume that we have experimental data from which we infer the values:

$$\begin{aligned} t_1 &= 0.667499, \\ t_2 &= 0.221782, \\ t_3 &= 0.0739131, \\ t_4 &= 0.024639, \\ t_5 &= 0.00821253, \\ t_6 &= 0.00273827, \\ t_7 &= 0.000912502, \\ t_8 &= 0.000304167. \end{aligned} \quad (37)$$

(these data have been generated by slightly perturbing the previous theoretical values with  $\beta = 1/3$ ). The previous computation leads to

$$\begin{aligned}
 \hat{q}_0 &= 0.667002, \\
 \hat{q}_1 &= 0.333269, \\
 \hat{q}_2 &= 0.0000610268, \\
 q_3 &= -0.0000590538, \\
 \hat{q}_4 &= 0.0000833483, \\
 \hat{q}_5 &= -0.000133577, \\
 \hat{q}_6 &= 0.00018942, \\
 \hat{q}_7 &= -0.00542309.
 \end{aligned} \tag{38}$$

Not only are there negative values but also the absolute value of  $q_k$  is increasing with  $k$  showing an amplifying effect of error propagation. Therefore, a lot of care should be exerted in order to carry out computations on experimental data. This can be matter of further investigation and is beyond the scope of this paper.

We show a second example for the inverse computation. Assume that

$$\begin{aligned}
 r_{2k+1} &= \frac{3^{k+1}}{2^{4k+2}} C_k, \\
 r_{2k+2} &= 0, \\
 k &= 0, 1, \dots,
 \end{aligned} \tag{39}$$

where  $C_k$  are the Catalan numbers. If we carry out the computation in (28) symbolically, we get

$$\begin{aligned}
 \hat{q}_0 &= \frac{3}{4}, \\
 \hat{q}_1 &= 0, \\
 \hat{q}_2 &= \frac{1}{4}, \\
 \hat{q}_k &= 0, \quad k > 2,
 \end{aligned} \tag{40}$$

from which

$$\begin{aligned}
 \hat{p}_1 &= \frac{9}{10}, \\
 \hat{p}_2 &= 0, \\
 \hat{p}_3 &= \frac{1}{10}, \\
 \hat{p}_k &= 0, \quad k > 3,
 \end{aligned} \tag{41}$$

so that

$$\begin{aligned}
 q_0 &= \frac{3}{4}u, \\
 q_1 &= 0,
 \end{aligned}$$

$$\begin{aligned}
 q_2 &= \frac{1}{4u}, \\
 q_k &= 0, \quad k > 2,
 \end{aligned} \tag{42}$$

and

$$\begin{aligned}
 p_1 &= \frac{9v}{10u}, \\
 p_2 &= 0, \\
 p_3 &= \frac{v}{10u^3}, \\
 p_k &= 0, \quad k > 3.
 \end{aligned} \tag{43}$$

The normalization yields  $u = 1/3$  and  $v = 5/27$ , so that  $p_1 = p_3 = 1/2$ . Again, we show how perturbed data can lead to strange outcomes. If we perturb the data as

$$\begin{aligned}
 r_1 &= 0.746705, \\
 r_3 &= 0.141068, \\
 r_5 &= 0.0529938, \\
 r_7 &= 0.02473, \\
 r_9 &= 0.0130018, \\
 r_{11} &= 0.00728396, \\
 r_{13} &= 0.00430417, \\
 r_{15} &= 0.00261343, \\
 r_{17} &= 0.00163685, \\
 r_{19} &= 0.00104289,
 \end{aligned} \tag{44}$$

with  $r_k = 0$  for the other indices, we get

$$\begin{aligned}
 \hat{q}_0 &= 0.746705, \\
 \hat{q}_2 &= 0.253005, \\
 \hat{q}_4 &= -0.000988595, \\
 \hat{q}_6 &= -0.000552489, \\
 \hat{q}_8 &= 0.00141413, \\
 \hat{q}_{10} &= -0.00223664, \\
 \hat{q}_{12} &= 0.00360276, \\
 \hat{q}_{14} &= -0.00597299, \\
 \hat{q}_{16} &= 0.0098459, \\
 q_{18} &= -0.0155087.
 \end{aligned} \tag{45}$$

We see again the same inconsistencies and the amplifying effect. In any case, we may note that the values with odd index are correctly computed as null values.

## 5. Branching Processes

Now we define  $q_h$ ,  $h = 0, 1, \dots$ , as the probability that a member of the population generates  $h$  offspring. We are interested in computing the probability  $r_k$  that the population will eventually have  $k$  members, starting from a population consisting of one member. If  $G(x)$  and  $H(x)$  are the probability generating functions of  $q_h$  and  $r_k$ , respectively, then the following functional equation holds:

$$H(x) = xG(H(x)). \quad (46)$$

Hence, the same relations (13) and (14) hold as well as the recursion (15). This time there are no  $s_k$  coefficients to be computed and the picture is simplified. We may still view a branching process like a random graph. However, while in random graphs we pick up randomly any vertex within the small components, in branching processes, the small components are rooted trees and we pick up the roots randomly. Hence, the  $r_k$  values we state here for a branching process can be related to the  $t_k$  values of random graphs.

Iteration (15) can also be carried out in exact arithmetic, thus producing results from which closed formulas can be inferred. As a simple example, suppose that  $q_0 = q_1 = q_2 = 1/3$  ( $q_i = 0$  for  $i > 2$ ). Then, by applying (15), we obtain a sequence whose first terms are

$$\left\{ \frac{1}{3}, \frac{1}{9}, \frac{2}{27}, \frac{4}{81}, \frac{1}{27}, \frac{7}{243}, \frac{17}{729}, \frac{127}{6561}, \frac{323}{19683}, \frac{835}{59049}, \frac{2188}{177147}, \dots \right\}. \quad (47)$$

We may guess that the denominator grows as the powers of 3, and so if we multiply the  $n$ th term of (47) by  $3^n$ , we obtain the new sequence

$$\{1, 1, 2, 4, 9, 21, 51, 127, 323, 835, 2188, \dots\}. \quad (48)$$

By looking at [7], we discover that these are the Motzkin numbers whose  $n$ th term is in closed form:

$$M_n = \sum_{k=0}^{\lfloor n/2 \rfloor} C_k \binom{n}{2k}, \quad (49)$$

with  $C_k$  being the  $k$ th Catalan number. Hence,

$$r_n = 3^{-n} M_{n-1}, \quad (50)$$

and we have found another combinatorial interpretation of the Motzkin numbers besides the many listed in [7]. The reason for  $n - 1$  as subscript is due to the fact that the first index of the sequence  $r_n$  is  $n = 1$ , whereas Motzkin numbers in (48) as defined above start from  $n = 0$ .

The same considerations about inferring the probabilities  $q_k$  from the probabilities  $t_k$  ( $= r_k$ ) can be applied also to branching processes. The example with  $t_k = (1 - \beta)\beta^{k-1}$  is almost trivial if we have in mind a branching process.

## 6. Conclusions

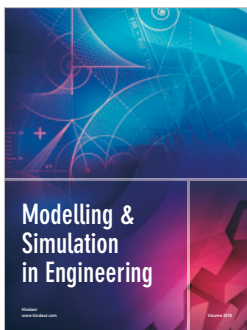
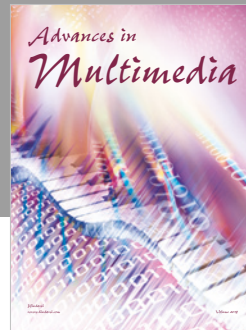
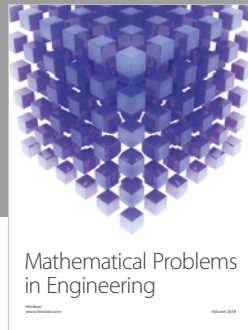
In this paper, we have presented an iterative scheme to compute the coefficients of a generating function that plays an important role in random graphs and in branching processes. The generating function is related to the population size probabilities for a branching process and to the small component size probabilities for random graphs. We also show that the iteration can be inverted; that is, for a branching process, from the population size probabilities, one can infer the offspring probabilities, but the inverse iteration is numerically unstable.

## Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

## References

- [1] N. Alon and J. H. Spencer, *The Probabilistic Method*, John Wiley & Sons, 2004.
- [2] N. Baumann and S. Stiller, "Network Models," in *Network analysis: methodological foundations*, U. Brands and T. Erlebach, Eds., vol. 3418 of *Lecture Notes in Computer Science*, pp. 341–372, Springer, Berlin, Germany, 2005.
- [3] M. E. J. Newman, *Networks: an introduction*, Oxford University Press, 2010.
- [4] M. E. J. Newman, S. H. Strogatz, and D. J. Watts, "Random graphs with arbitrary degree distributions and their applications," *Physical Review E*, vol. 64, Article ID 026118, 2001.
- [5] Y. Shang, "Impact of self-healing capability on network robustness," *Physical Review E*, vol. 91, Article ID 042804, 2015.
- [6] Y. Shang, "Effect of link oriented self-healing on resilience of networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2016, no. 8, Article ID 083403, 2016.
- [7] N. J. A. Sloane, "The on-line encyclopedia of integer sequences: sequence A001006," <http://oeis.org/A001006>, 2015.



Hindawi

Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

