Random walk with memory on complex networks

Lasko Basnarkov,[1,2,*] Miroslav Mirchev,[1] and Ljupco Kocarev[1,2]
[1]*Faculty of Computer Science and Engineering, SS. Cyril and Methodius University, P.O. Box 393, 1000 Skopje, Macedonia*
[2]*Macedonian Academy of Sciences and Arts, P.O. Box 428, 1000 Skopje, Macedonia*

(Received 3 July 2019; revised 15 June 2020; accepted 16 October 2020; published 30 October 2020)

We study random walks on complex networks with transition probabilities which depend on the current and previously visited nodes. By using an absorbing Markov chain we derive an exact expression for the mean first passage time between pairs of nodes, for a random walk with a memory of one step. We have analyzed one particular model of random walk, where the transition probabilities depend on the number of paths to the second neighbors. The numerical experiments on paradigmatic complex networks verify the validity of the theoretical expressions, and also indicate that the flattening of the stationary occupation probability accompanies a nearly optimal random search.

DOI: 10.1103/PhysRevE.102.042315

I. INTRODUCTION

The pursuit for appropriate models of the nontrivial interconnections between the units of real systems has led to the emergence of complex network theory as one of the most fruitful fields in modern science. Instead of being regular, or purely random [1], the graph of connections between the items rather frequently possesses characteristics like the small-world property [2] and power-law degree distribution [3]. These topological features have strong implications on the dynamics which might be present in the system. A list of such dynamical processes on complex networks of interacting units can include synchronization [4], consensus formation [5], disease spreading [6], and so on.

The random walk is one of the most pervasive concepts in natural sciences which is applied in studies of diverse phenomena ranging from simple animal strategies for food location [7,8] to complex human interactions resulting in stock price variations [9], or evolution of research interests [10]. A recent paper [11] contains a nice review of the topic and a long list of references. A large portion of dynamical processes on complex networks like the PageRank algorithm [12], various types of searching [13,14], or community detection [15] is based on or related to the random walk. A random searching process in a complex network is formulated as follows: starting from an arbitrary node, or source i, sequentially hop from a node to one randomly chosen neighbor until reaching some previously defined target node j. The performance of a searching procedure is measured in terms of the number of steps needed to get from i to j and the related quantity is known as first passage time. Due to the stochastic nature of picking the nodes in the sequence, sometimes one can be very lucky and rather quickly find the target, while in most of the trials the number of steps would be larger than the number of nodes in the network for a typical source-target pair. Therefore, a more informative quantity is the average number needed to complete the task—the mean first passage time (MFPT)—obtained by averaging across all possible realizations of the random choices.

On the other side, there are efficient deterministic searching algorithms which rely on information about the underlying graph structure. In such approaches, when one has knowledge of the full structure of the graph, the shortest paths are used, and then one needs the smallest number of steps to reach the target. However, for very large systems, like the World Wide Web, or in dynamical environments like mobile sensor networks, keeping and updating all necessary topological information might be a serious issue. Then one could turn towards strategies based on local information only. The classical uniform random walk (URW) needs the smallest amount of information—only the number of neighbors (the degree k_i) of each node i. Within this approach, the probabilities for choosing among the neighbors of some node i are taken to be identical and equal to the inverse of its degree $p = 1/k_i$. However, this procedure greatly increases the time to completion of the task, which is another type of inconvenience. The searching can be improved when the local information extends the node degrees. For example, it was shown that, for a certain type of small-world networks, a random target can be found rather quickly by using local information only [16,17]. Knowledge of the identities of the direct or maybe more distant neighbors also enhances the searching [18].

There are various alternatives for modification of the URW aimed at speeding up its searching capabilities. Some of these works provided enhancements while others also presented connections with related problems in other fields. For example, as a counterpart of path integrals, the maximal entropy random walk was introduced as a modification of URWs which assigns equal probabilities to all paths with equal length starting from a certain node [19]. In another approach, the Lévy random walk, which allows for jumps toward more distant nodes besides the (first) neighbors, was proven to decrease the expected time needed to visit all nodes in a network [20]. The combination of the local diffusion and knowledge of the topology has recently been applied to study the routing

*lasko.basnarkov@finki.ukim.mk

of neural signals [21]. Biasing of the random walk has been shown to be useful in sampling of complex networks as well [22]. Another important achievement was the demonstration that biasing of the URW, by preferring the less connected neighbors, can improve random searching in complex networks [23]. In the same contribution, it was uncovered that inverse-degree-based biasing of the random walk also leads to uniform stationary occupation probability. In a related work, it was obtained that the improvement is greatest when the probability to jump to a neighbor is inversely proportional to its degree [24].

In this work we explore the potential for searching improvement by considering memory-based random walks on complex networks since it relies on information that extends the immediate neighborhood. First, we develop a theoretical framework for analytical calculation of the MFPT between any pair of nodes when the random walk has a memory of one step. Then we apply it for determination of MFPT for one particular searching algorithm which aims to provide nearly equal chances of visiting second neighbors. We numerically show that searching enhancement is also accompanied with flattening of the stationary distribution of the visiting frequency as is the case of the inverse-degree-based biasing of the random walk. The co-occurrence of the improved searching and nearly uniform stationary distribution is found even for the memory-based and inverse-degree-based random walk on directed complex networks as well.

The remainder of the text is organized as follows. In Sec. II we present general theoretical framework for studying random search with random walk with memory. Then in Sec. III we introduce an algorithm for random search with memory of one step. The results from the numerical experiments and their analysis are provided in Sec. IV. The paper finishes with the conclusions.

II. MEAN FIRST PASSAGE TIME OF RANDOM WALK WITH MEMORY ON COMPLEX NETWORKS

Consider a connected complex network with N nodes and connections encoded in the adjacency matrix $\mathbf{A}$. For simplicity, we study a discrete-time walk, where the next node in the sequence is chosen randomly, with time-invariant transitions, or jump probability, which depend only on the current node and the node visited immediately before it. This represents a random walk with a memory of one step, which can be straightforwardly generalized to cases with longer memory. To be more specific, assume that the random walker at a certain time step has moved from node r to its neighbor s. For one-step memory, the probability of proceeding towards some neighbor t from s is given as $p(t|s,r)$ which means that it depends only on the current node s and the previously visited one r, but not on those preceding r.[1] This kind of random walk can be suitably studied with a related Markov chain with states conveniently denoted with rs, which are pairs of consecutively visited nodes r and s. The transition probabilities from state rs to st in this chain are thus $p_{rs,st} = p(t|s,r)$, which can be compactly organized in the respective transition-probability matrix $\mathbf{P}$.

When the transition-probability matrix $\mathbf{P}$ of the related Markov chain is determined, such a random walk will be completely defined once the starting step is specified. One particular initialization of the walk which starts from some node i is to choose randomly one of its neighbors and then continue with the memory-based algorithm specified with the matrix $\mathbf{P}$. Finding some target j in the network corresponds to reaching any of the states denoted with sj in the Markov chain, where s is any neighbor of the node j.[2] Then, the MFPT from node i to j could be related to the mean time to absorption (MTA) of the random walk initialized at any state ir in an absorbing Markov chain corresponding to the node j as target. In this absorbing chain, all states sj are absorbing, while the remaining ones rt, where $t \neq j$, and $r \neq j$ are transient states.[3] Before deriving the relationship between MTA and MFPT, we first present some well-known results about the MTA in absorbing Markov chains, which can be found, for example, in Ref. [25]. For this purpose, one should first determine the transition matrix of the absorbing Markov chain, which depends on the target j, and is thus conveniently denoted with $\mathbf{P}_{(j)}$. The construction of $\mathbf{P}_{(j)}$ relies on two observations. By the first one the random walk stops at any absorbing state, which implies that the probability of leaving any such state is zero. From the second, the transition probabilities between the transient states and from the transient to the absorbing states in the absorbing chain are the same as the respective ones in the original chain. The transition matrix of the absorbing Markov chains is conveniently represented in the canonical form, which for a memory-based random walk targeting the node j reads

$$\mathbf{P}_{(j)} = \begin{vmatrix} \mathbf{Q}_{(j)} & \mathbf{R}_{(j)} \\ \mathbf{0} & \mathbf{I} \end{vmatrix}. \tag{1}$$

In the last equation, $\mathbf{Q}_{(j)}$ is a matrix consisting of transition probabilities between the transient states, the submatrix $\mathbf{R}_{(j)}$ corresponds to the probabilities of the transitions from the transient to the absorbing states, while the appropriately sized zero matrix $\mathbf{0}$ and identity matrix $\mathbf{I}$ mean that the random walker does not continue further from any of the absorbing states. The MTA in an absorbing Markov chain determined by the target node j from all possible starting states ir is the respective element of the column vector

$$\mu_{(j)} = \mathbf{Y}_{(j)}\mathbf{c}, \tag{2}$$

where $\mathbf{c}$ is a column vector with all elements equal to one, while $\mathbf{Y}_{(j)}$ is the fundamental matrix of the absorbing Markov chain [25], which is calculated with

$$\mathbf{Y}_{(j)} = (\mathbf{I} - \mathbf{Q}_{(j)})^{-1}. \tag{3}$$

[1] For directed networks, the neighbor t must be chosen from among those toward which s points.

[2] For a directed network, nodes s are only those which point to j.

[3] One should note that, in the Markov chain that models the random walk with memory, there can be states denoted js. They can be included in the absorbing chain only if one needs to calculate the mean recurrence time, or the average time needed for the walker starting from j to return at j again. When the starting node differs from the target, such states are omitted in order to reduce the size of the matrices involved.

The MTA vector $\mu_{(j)}$ contains sufficient information for calculation of the MFPTs from all starting nodes to the given target j, as shown below.

Since in a random walk on complex networks at a single time step, exactly one hop is made, each random first passage time equals the number of steps needed for reaching the target for the first time, which is the length of the respective random walk. Thus, by definition, the MFPT between the starting node i and the target j is the weighted sum of the lengths l of all walks $\mathcal{W}_{i,j}$, which visit j only at the last step:

$$m_{i,j} = \sum_{\mathcal{W}_{i,j}} l(\mathcal{W}_{i,j})p(\mathcal{W}_{i,j}), \tag{4}$$

where $p(\mathcal{W}_{i,j})$ is the probability of occurrence of the walk $\mathcal{W}_{i,j}$. Let us first consider the case when the target is not a neighbor of the source. Then, the sum in the last expression can be organized by summing over all walks with memory that visit the neighbor s of i at the first step, and then summing over the whole neighborhood $\mathcal{N}_i$ of i:

$$m_{i,j} = \sum_{s\in\mathcal{N}_i} p_{i,s} \sum_{\mathcal{W}_{s,j}} [1 + l(\mathcal{W}_{s,j})]p(\mathcal{W}_{s,j}), \tag{5}$$

where $p_{i,s}$ denotes the probability to hop from i to s in the first step. Since every random walk in an absorbing Markov chain is absorbed with probability one [25], the measure of the memory-based random walks in the complex network that miss the target j indefinitely is zero. This implies that the normalization condition of the probabilities of the memory-based walks that pass through each neighbor s of the initial node i and reach the target j, is given as

$$\sum_{\mathcal{W}_{s,j}} p(\mathcal{W}_{s,j}) = 1, \tag{6}$$

where the summation is made for each neighbor s separately. One can also note that the MFTP from the neighbor s of the starting node i to the target j by pursuing the memory-based random walk equals the MTA from the starting state is in the absorbing Markov chain determined with the same target. This MTA is the respective term $\mu_{(j),is}$ of the MTA vector $\mu_{(j)}$ and is given with the following sum

$$\mu_{(j),is} = \sum_{\mathcal{W}_{s,j}} l(\mathcal{W}_{s,j})p(\mathcal{W}_{s,j}). \tag{7}$$

When the neighbor in the first step is chosen uniformly, one has $p_{i,s} = 1/k_i$. Then, by using Eqs. (6) and (7) in Eq. (5), one can express the MFPT from the node i to j through the MTAs obtained by the Markov model for the memory-based random walk as

$$m_{i,j} = 1 + \frac{1}{k_i} \sum_{s\in\mathcal{N}_i} \mu_{(j),is}. \tag{8}$$

When the target j is a neighbor of i, it can be either reached through the direct one-step route with probability $p_{i,j} = 1/k_i$, or through longer walks for which one can apply the same reasoning as above. We note that, in considering the walks through the other neighbors of the initial node i, in the sum running over the neighborhood of i, the target j should be omitted. Then, by using the normalization condition (6), one can obtain that

$$\sum_{s\,\in\,\mathcal{N}_i\ \ s\,\neq\, j} p_{i,s} \sum_{\mathcal{W}_{s,j}} [1 + l(\mathcal{W}_{s,j})]p(\mathcal{W}_{s,j}) = \frac{k_i - 1}{k_i} + \frac{1}{k_i} \sum_{s\,\in\,\mathcal{N}_i\ \ s\,\neq\, j} \mu_{(j),is}. \tag{9}$$

By adding the contribution of the direct walk to the last expression one will obtain similar result as (8):

$$m_{i,j} = 1 + \frac{1}{k_i} \sum_{s\,\in\,\mathcal{N}_i\ \ s\,\neq\, j} \mu_{(j),is}. \tag{10}$$

If the trivial value $\mu_{(j),ij} = 0$ is used, one can see that the same expression (8) can be used for any target, regardless of whether it is neighbor of the starting node.

The analysis above can be applied to the simpler case as well—the random walk without memory. The Markov transition matrix in this situation consists of the transition probabilities between the nodes. Then, with each target node is associated only one absorbing state—the target itself. However, it is more convenient to have one fundamental matrix $\mathbf{P}$ for the whole network, instead of using a different one for each node separately. Without going into details, which can be found, for example, in Ref. [25], we just state that the MFPT from the starting node i to the target j is given by

$$m_{i,j} = \frac{z_{j,j} - z_{i,j}}{w_j}, \tag{11}$$

where $z_{i,j}$ are the elements of the respective fundamental matrix $\mathbf{Z}$, while w_j is the stationary occupation probability of node j. The fundamental matrix $\mathbf{Z}$ is obtained from

$$\mathbf{Z} = (\mathbf{I} - \mathbf{P} + \mathbf{W})^{-1}, \tag{12}$$

where the matrix $\mathbf{W}$ consists of rows identical to the stationary probability distribution $\mathbf{w}$ of $\mathbf{P}$. We use the expression (11) to calculate the MFPT between nodes in the memoryless random walk with which our model is compared. The reader interested in a more detailed and intuitive derivation of the same expression (11) with the generating functions formalism, but for lattices only, can refer to Ref. [7], while for complex networks based on the Laplace transform, deeper explanation can be found in Refs. [23,26].

One should note that the MFPT is a property of the network parametrized by two nodes—the starting one i and the final j and is thus sensitive to the choice of this pair. A related property of one node only is obtained by averaging all MFPTs starting from all other nodes and targeting it

$$g_i = \frac{1}{N-1} \sum_{j\,=\,1 j\,\neq\, i}^{N} m_{j,i}. \tag{13}$$

In the literature it was called global mean first passage time—GMFPT [27]. This property can be also seen as a kind of centrality measure of nodes in a complex network. By going one step further, one can average across GMFPTs for all nodes and get a property of the whole network which was introduced as the Graph MFPT (GrMFPT) [24]:

$$G = \frac{1}{N}\sum_{i=1}^{N} g_i. \tag{14}$$

We use this variable for comparison of the searching by different random walks in complex networks.

III. SEARCHING ALGORITHM BASED ON RANDOM WALK WITH MEMORY OF ONE STEP

The results for the MFPT obtained in the previous section are general and hold for every random walk with jumping probabilities depending on the current and previously visited node. They are given in a form that does not provide much intuition about which navigation rules provide a better search for the target. Even from the expression for the MFPT of the memoryless walk, one is not sure how the transition probabilities should be defined in order to obtain a faster search. We stress that an interesting contribution was the finding that, if the probability to jump to a neighbor is the inverse of that neighbor's degree, then the search in an undirected network is faster compared with the URW, and in that scenario the stationary occupation probability approaches the uniform one $\mathbf{w}_j \approx 1/N$ [23]. This suggests that searching improvement could be expected from biasing, which increases the probability for visiting poorly connected nodes, as the inverse degree algorithm does. As shown in the Appendix, under certain circumstances, inverse-degree biasing can result in a nearly constant distribution of visiting frequencies even for memoryless random walks on directed networks as well. This flattening of the invariant density happens in well-connected networks, in which each node has many neighbors. As we will see below, our numerical simulations indicated that inverse-degree biasing does not improve searching for networks with small average degree. In that case, the distribution of visiting frequency deviates more significantly from the uniform one as well. Thus, navigation rules which favor jumps towards less connected nodes and result in a nearly uniform distribution of visiting frequency could be candidates for a good searching algorithm.

Memory-based algorithms are obviously more complex than their memoryless counterparts and their implementation could be justified if they provide improved searching. Guided by the reasoning above, one can pursue a strategy which should result in decreased differences between the probabilities for reaching the second neighbors, which hopefully would bring uniform stationary occupation probability and faster searching. One intuitive way to make such navigation rules is as follows: Assume that, at the previous step, the walker was at node r, from where it jumped to the node s, and in the next step it would visit some node t from the set of neighbors of s. Denote by b_{rt} the number of all two-hop walks from node r to t. The matrix $\mathbf{B}$ with elements b_{rt} is the square of the adjacency matrix $\mathbf{A}$, $\mathbf{B} = \mathbf{A}^2$. Then, the probability to visit node t after being at nodes r and s in the previous two steps corresponds to the transition probability from state rs to st in the related Markov chain. In analogy to inverse-degree biasing, one could favor visiting the less accessible second neighbors by choosing the following jumping probability:

$$p_{rs,st} = \frac{\frac{1}{b_{rt}}}{\sum_{u\in\mathcal{N}_s}\frac{1}{b_{ru}}}, \tag{15}$$

where the sum in the denominator is used for normalization of the probabilities and it runs in the neighborhood of the node s, $\mathcal{N}_s$. This formula assigns a larger weight to nodes t which have less alternative paths to be reached from node r, i.e., those with smaller b_{rt}. In this way, the probability to visit a node of that kind from r in two steps will be increased and become closer to that of nodes which are accessible from r in two steps through more alternative ways. We note that, for undirected networks, every node is a second neighbor to itself, and there is a chance to return to the same node r. However, $b_{rr} = k_r$ and the probability $p_{rs,sr}$ is the lowest within all $p_{rs,st}$, hence, immediate returning is disfavored. In this way, the appearance of short loops is suppressed.

The related Markov model of a random walk with memory could be successfully applied for an analytical calculation of the stationary occupation probability as well, which could be used to check whether its flattening is accompanied by a searching improvement. To find the stationary occupation probability, one should first calculate the invariant distribution of the states of the related Markov chain $\mathbf{v}$, which is obtained from the stationarity condition $\mathbf{vP} = \mathbf{v}$ of the full transition matrix $\mathbf{P}$ of the Markov chain. Its terms are the stationary probabilities of states v_{rs} that correspond to all pairs of neighbors rs. Then, the stationary distribution of frequency of visits of the node s, by a random walk with memory of one step, would be either of the sums $\sum_r v_{rs}$ or $\sum_t v_{st}$ running within the neighborhood of node s.

IV. NUMERICAL RESULTS

In this section we provide the results obtained by using analytical expressions and numerical simulations with memory-based random walks and compare them with the uniform and inverse-degree-biased random walk. The search effectiveness was studied by calculating the GrMFPT of each considered network. The stationary occupation probability was also calculated to check whether its flattening accompanies efficient searching. First, we conduct a thorough analysis using generic network models, such as random, scale-free, and small-world networks. Then, we apply the approaches on two real networks: the Internet at autonomous systems level (undirected), and a reduced set of Wikipedia pages (directed).

The calculations of theoretical expressions involve inverse matrix operations, and the latter presents the major constraint in our analysis. For the random walk with memory, the number of states in the related Markov chain equals the number of links, which limits the size of networks that we could study. Therefore, we opted to perform the analyses of the MFPT and the invariant density for networks with $N = 100$ nodes. We varied the average node degree by changing the native model parameters to see how the connectivity affects the search. For both the analytical and the numerical results, we averaged over ten network instances for every parameter setting for each network type. Moreover, in the numerical simulations we performed 100 repetitions of the search among all node pairs, for each scenario.

We studied purely random graphs, scale-free and small-world networks as the most typical kinds of networks. For generating such graphs we used algorithms from the NetworkX library in PYTHON, which allow construction of the three graph types with given parameter values [28]. The random graphs are complex networks created according to the Erdős-Rényi model where every pair of nodes i and j is connected with some predefined probability p, which appears as parameter of the graph together with the number of nodes N [1]. If the probability p is large enough then the obtained graph would very likely be connected—there will be a path between each pair of nodes. The small-world networks were built following the Watts-Strogatz model [2]. It starts with a regular ring lattice network with N nodes each connected with n neighbors, and then randomly rewires the links with some probability p. The scale-free networks were generated using the Barabási-Albert model, which sequentially builds the network by adding nodes one by one [3]. The network builds upon a seed of m_0 nodes without edges, and every newly added node forms m links with the existing network.[4] Preferential attachment is employed as the probability to connect to existing nodes is taken to be proportional to its degree.

In Fig. 1(a) we compare the obtained GrMFPTs for the URW, inverse-degree-biased random walk, and the memory-based random walk over scale-free networks. The horizontal axis represents the average degree $\langle k\rangle$, which is approximately $2m$, where $m \in [2, 10]$. The seed network is composed of $m_0 = m$ nodes without edges. First, one can observe that the numerical (N) and the theoretical (T) results are very close, which confirms the correctness of the analytical expressions. The memory-based random walk always outperforms the uniform one. The inverse-degree-biased random walk is also better than the URW, when the average node degree is not very small. One can notice that all curves decrease asymptotically towards the value corresponding to the number of nodes N. As we will see from the other numerical results, N seems to be the minimal possible value for the GrMFPT. Thus, as an optimal random search could be considered the one for which GrMFPT equals the number of nodes, $G = N$. Although for networks with very large average degree the GrMFPT seems to approach to N for different kinds of random walk, the effectiveness of a biasing procedure becomes apparent for less connected networks.

We note that there is deterministic strategy that is twice as fast and which holds for graphs that have a Hamiltonian cycle. It is a walk passing though all nodes and visiting each node only once. We emphasize here that determination whether a graph has a Hamiltonian cycle is not a trivial task and was proven to be an NP-complete problem [29]. In that case the MFPT from the source to the target will equal the number of nodes in between them along the cycle, and for uniformly chosen starting and target nodes, one can easily show that GrMFPT will be $N/2$.

[4]The parameters m_0 and m here are denoted as the authors Barabási and Albert originally did and are different from the elements of the MFPT matrix $m_{i,j}$.

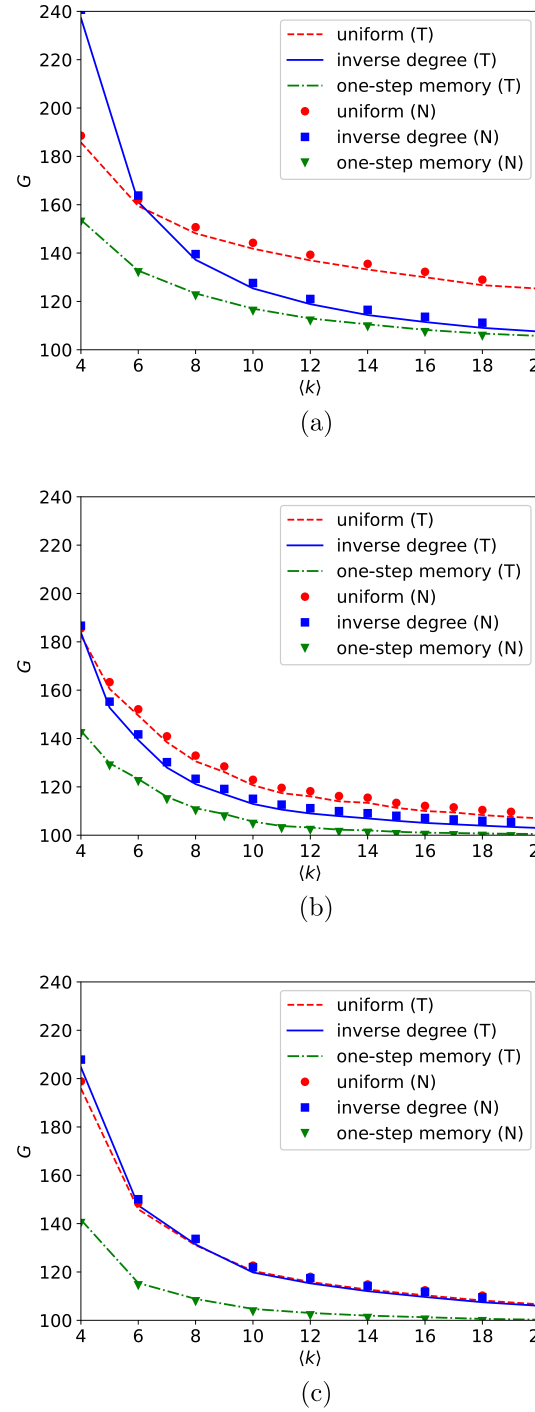


FIG. 1. GrMFPT in (a) BA, (b) ER, and (c) WS networks of $N = 100$ nodes with different average node degree $\langle k\rangle$ for the three cases: uniform (red line and circles), inverse degree (blue line and squares) and one-step memory (green line and triangles). The lines are theoretical values (T) and the markers numerical estimates (N).

The two biasing procedures bring search improvement for the purely random Erdős-Rényi graphs also, as shown in Fig. 1(b). We generated ten network instances with $N = 100$ nodes for different average node degree $\langle k\rangle$ by varying the link existence probability $p \in [0.04, 0.2]$. As can be seen, the inverse-degree biasing gives lower GrMFPT than the

URW, except for $\langle k\rangle = 4$ where they are about the same, while the one-step memory outperforms them both. Again the numerical results are in accordance with the theoretical results.

In Fig. 1(c) we show how the biasing affects the random walk in Watts-Strogatz networks, where the rewiring probability is $p = 0.2$. Unlike for the other network types under study, the inverse-degree biasing does not improve the GrMFPT. This is probably due to the smaller degree variability in this kind of network. On the other hand, the one-step memory approach still reduces the GrMFPT, as was the case for the other network types. The theoretical expressions are once again confirmed by the numerical simulations.

We also made numerical experiments to see whether a mechanism behind the search improvement is nearly uniform stationary occupation probability. The extent of flattening of the stationary occupation probability was studied with the Kullback-Leibler (KL) divergence [30]. KL divergence estimates the deviation of one distribution from another. In the case when one has two discrete distributions $P(i)$ and $Q(i)$, it is defined as

$$D_{\mathrm{KL}}(P||Q) = \sum_i P(i)\log\frac{P(i)}{Q(i)}. \tag{16}$$

One can notice from the definition that this is an asymmetric quantity, $D_{\mathrm{KL}}(P||Q) \neq D_{\mathrm{KL}}(Q||P)$ and, within the definition provided above, P has the role of the prior, or the distribution with which we compare. In our case it is the constant $P(i) = 1/N$. This divergence vanishes when the two distributions coincide. In Fig. 2(a) is shown the KL divergence between the constant density and those for the uniform, inverse degree, and one-step memory random walks in BA networks. As can be noticed, both biasing procedures result in invariant density that is closer to the flat one than the uniform approach does. Also, the larger the average degree is, the approximation of the invariant density with the uniform one is more correct, as the theoretical analysis in the Appendix suggests. However, even though for networks with smaller average degree the biasing makes the distribution closer to the uniform, searching is slower than for the URW. This clearly indicates that the leveling of the visiting frequencies is not always sufficient for optimizing the search.

Similarly, Fig. 2(b) shows the KL divergence between the uniform density and the distribution of the visiting frequency for the URW and the two other random walks in ER networks. Once again, the biasing yields a density that is closer to the constant one than the URW, which is probably the reason for the lower GrMFPT obtained in Fig. 1(b). On the other hand, in WS networks [see Fig. 2(c)] the inverse-degree biasing gives a density which is closer to a constant than the URW, while the one-step memory approach does not, even though it proved fastest in such scenarios as is evinced in Fig. 1(c).

We also numerically compared the searching performance of the three kinds of random walks in directed networks. In Fig. 3(a) are shown the respective GrMFPTs. We can see that the one-step memory provides better results than the URW, but the inverse indegree approach outperforms them both significantly, which was not the case in the undirected networks.

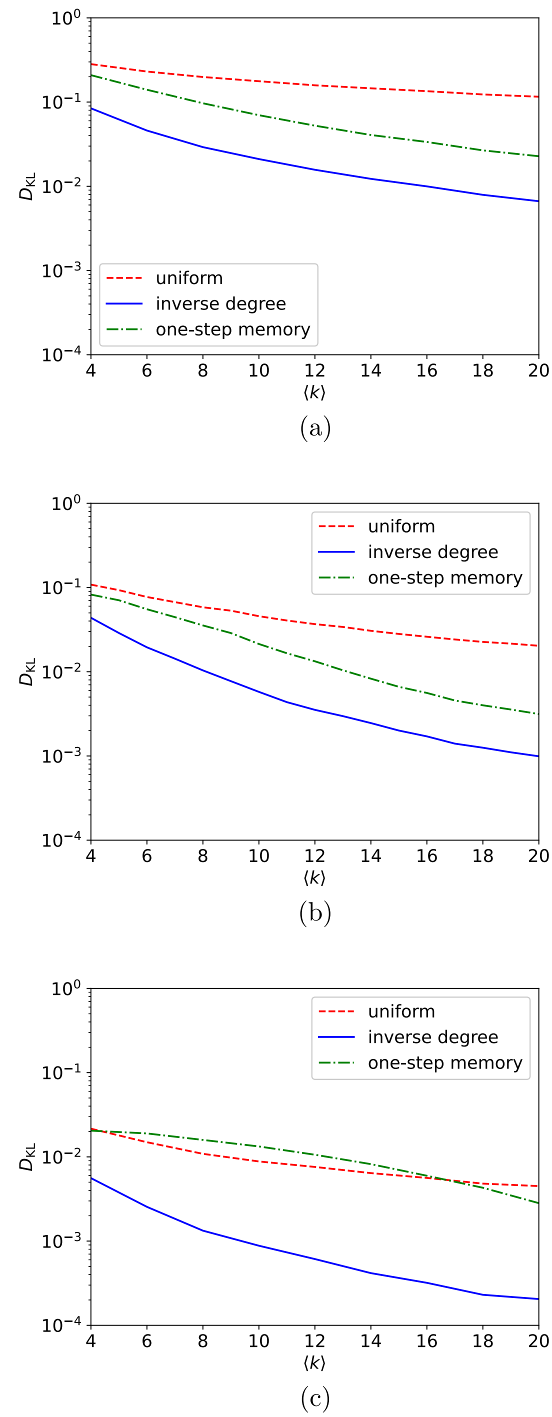
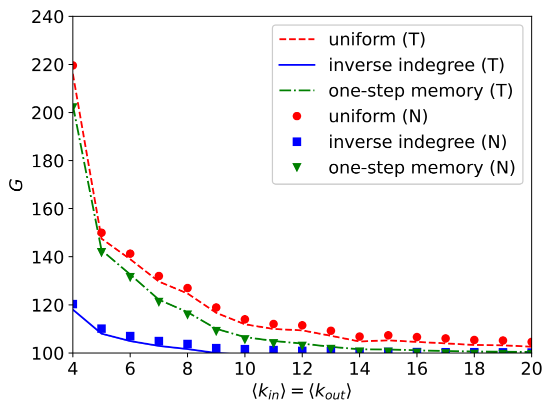


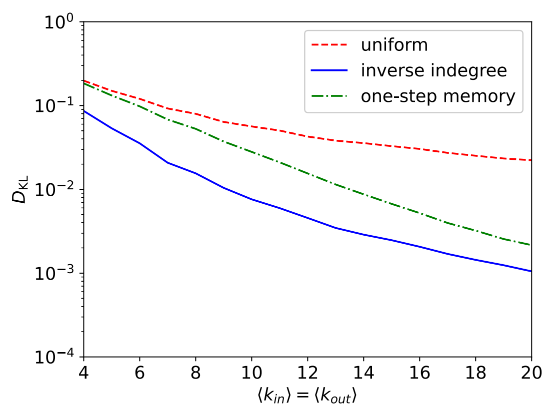
FIG. 2. Kullback-Leibler divergence of the stationary occupation probability of uniform (red), inverse degree (blue), and one-step memory (green) random walks from the uniform density in (a) BA, (b) ER, and (c) WS networks with $N = 100$ nodes for different average node degrees.

Biasing based on inverse outdegree performs slower than the URW (results are not shown), as expected.

The flattening of the invariant density is an ingredient which helps in search improvement in directed networks as well. We have numerically verified that, as expected, for well-connected networks when biasing of random walk is based



(a)



(b)

FIG. 3. Random walks in directed ER networks with different average degree $\langle k\rangle$: (a) Comparison of the GrMFPT for uniform (red circles), inverse indegree (blue squares), and one-stop memory (green triangles). (b) Kullback-Leibler divergence of the invariant density from a uniform density for three approaches: uniform (red circles), inverse-indegree biased (blue squares) and one-step memory (green triangles).

on inverse of indegrees, the invariant density is closer to the constant than that of a URW. In Fig. 3(b) are shown the KL divergence of the URW on directed ER networks with the two biasing alternatives: one based on inverse of indegrees, and another on the one-step memory. The results are in concordance with the theoretical analysis.

We have finally tried the searching performance of the three approaches in two real-world networks. The first network is a snapshot of the Internet topology at autonomous systems level obtained from BGP logs on January 2, 2000, which is an undirected graph consisting of 6474 nodes and 13 233 links [31]. Its average node degree is $\langle k\rangle \approx 4$. The second network is an extracted set of Wikipedia pages [32,33]. The graph is directed and consists of 4592 nodes and 119 882 links, from which we take the largest strongly connected component that has 4051 nodes and 119 000 links. The average indegree and outdegree of the largest component are $\langle k_{\text{in}}\rangle = \langle k_{\text{out}}\rangle \approx 29$. These networks are larger and it is much more difficult to calculate the GrMFTP theoretically, so in Table I we provide only the results obtained by numerical simulations. The results for the Internet network are obtained by averaging over 10^6 randomly selected source-target pairs out of 6474×6473 possible couples. For the extract of the Wikipedia network the averaging is performed with 1.5×10^5 pairs, out of 4051×4050 possible, because the simulations take much longer due to the larger number of steps required to reach the targets. One can note that, for the undirected case, the inverse-degree biasing worsens the search of the URW because the majority of nodes are not well connected as the theory asks, while it shows great reduction of the MFPT for the directed network. The memory-based strategy performs well in both scenarios. These results confirm our previous findings for paradigmatic network models that the inverse indegree biasing is better for directed networks, while the memory-based approach outperforms the others for undirected ones.

TABLE I. GrMFPT for two real networks with uniform, inverse degree and one step-memory random walks.

Network	Uniform	Inverse degree	One-step memory
Internet (AS)	1.93×10^4	1.78×10^5	1.80×10^4
Wikipedia (extr.)	3.01×10^7	1.09×10^4	8.15×10^5

V. CONCLUSIONS

In this work we studied random walks on complex networks with transition probabilities that depend on the nodes visited in the recent past. We have shown that such walks can be analyzed with the appropriate Markov chain, and for the case of memory of one step we derived an exact expression for the MFPT between pairs of nodes. One particular navigation algorithm was proposed that avoids the hubs by accounting for the two-hop-paths between the nodes. The searching ability of this algorithm was compared with that of the URW, and of another hubs-avoiding biased random walk with jumping probabilities inversely proportional to the node degrees. The proposed one-step memory approach has shown better searching performance than the URW and the inverse-degree-biased random walk for undirected networks, particularly when the majority of nodes have a small degree. We have furthermore demonstrated that the inverse-degree biasing based on indegree leads to improved random search in directed networks, which is even better than the memory-based one. The introduced technique with absorbing Markov chain could be also applied in theoretical analysis of other scenarios. One example is random searching of targets when each node knows the identity of its neighbors. In this case the absorbing states would be all neighbors of the target.

The numerical experiments on generic network models besides verifying the correctness of the theoretical expressions, have shown that when the nodes have enough neighbors, the GrMFPT approaches the number of nodes from above, for the three kinds of random walks considered. However, the usefulness of the biasing alternatives is that they allow nearly optimal performance to be achieved for less connected networks than the URW does. Also, both biasing approaches show better flattening towards the constant of the stationary occupation probability than the URW. The inverse-degree biasing results in stationary occupation probability that is always closer to the uniform than the two other kinds of random walk. This is not sufficient for best searching because it was obtained that the memory-based random walk performs better on undirected networks. However, the obtained results suggest that leveling of the stationary occupation probability can at least serve as an indicator for a possibly good searching algorithm, particularly when the respective KL divergence has very small value.

ACKNOWLEDGMENTS

This research was partially supported by the Faculty of Computer Science and Engineering at the Ss. Cyril and Methodius University in Skopje, Macedonia.

APPENDIX: CONDITIONS FOR NEARLY UNIFORM DISTRIBUTION OF THE VISITING FREQUENCY

The analysis in this section will be performed for a random walk on directed complex networks, although the same reasoning applies for undirected networks as well with minor modifications. Consider a random walk on a directed network, with the transition probability toward certain node j to be inversely proportional to its indegree k_j^{in}. Due to the normalization, the jumping probability from node i to its neighbor j would then be

$$p_{ij} = \frac{1/k_j^{\text{in}}}{\sum_{l\in\mathcal{N}_i^{\text{out}}} 1/k_l^{\text{in}}}, \tag{A1}$$

where $\mathcal{N}_i^{\text{out}}$ denotes the set of neighbors of the node i toward which it points to. Define node-centric, local average of the reciprocal of indegrees of the neighbors as

$$\langle 1/k\rangle_i^{\text{in}} = \frac{1}{k_i^{\text{out}}} \sum_{l\in\mathcal{N}_i^{\text{out}}} 1/k_l^{\text{in}}, \tag{A2}$$

where the subscript i in the average denotes that it is calculated only over the set $\mathcal{N}_i^{\text{out}}$. Then the normalization sum in Eq. (A1) can be expressed through the local average as

$$\sum_{l\in\mathcal{N}_i^{\text{out}}} 1/k_l^{\text{in}} = k_i^{\text{out}} \langle 1/k\rangle_i^{\text{in}}. \tag{A3}$$

Now, consider well-connected uncorrelated networks. Such networks are those where the degree of any node is independent of the degrees of its neighbors and where, for the majority of the nodes, $k_i^{\text{in}} \gg 1$ and $k_i^{\text{out}} \gg 1$ hold. Then the local average can be approximated with the network average of the reciprocal of indegrees:

$$\langle 1/k\rangle_i^{\text{in}} \approx \langle 1/k\rangle^{\text{in}} = \frac{1}{N}\sum_{j=1}^{N} 1/k_j^{\text{in}}. \tag{A4}$$

In such a case the normalization sum appearing in the denominator in Eq. (A1) can be conveniently expressed through the network average as

$$\sum_{l\in\mathcal{N}_i^{\text{out}}} 1/k_l^{\text{in}} \approx k_i^{\text{out}} \langle 1/k\rangle^{\text{in}}. \tag{A5}$$

The stationary distribution of the visiting frequency satisfies the following set of self-consistent equations:

$$w_j = \sum_{i\in\mathcal{N}_j} p_{i,j} w_i, \tag{A6}$$

for each node j. This means that the following holds:

$$w_j = \sum_{i\in\mathcal{N}_j^{\text{in}}} \frac{1/k_j^{\text{in}}}{k_i^{\text{out}}\langle 1/k\rangle^{\text{in}}} w_i = \frac{1/k_j^{\text{in}}}{\langle 1/k\rangle^{\text{in}}} \sum_{i\in\mathcal{N}_j^{\text{in}}} \frac{w_i}{k_i^{\text{out}}}. \tag{A7}$$

If one assumes that the invariant density is constant, $w_i = 1/N$, then from Eq. (A7) one would have

$$\frac{1}{N} \approx \frac{1/k_j^{\text{in}}}{N\langle 1/k\rangle^{\text{in}}} \sum_{i\in\mathcal{N}_j^{\text{in}}} \frac{1}{k_i^{\text{out}}}. \tag{A8}$$

Now, for networks where the direction of the links is independent of the degree of nodes, the averages of reciprocals of indegrees and outdegrees would be nearly the same

$$\langle 1/k\rangle^{\text{in}} \approx \langle 1/k\rangle^{\text{out}}. \tag{A9}$$

For networks where most of the nodes have many incoming and outgoing links, one can make the following approximation:

$$\sum_{i\in\mathcal{N}_j^{\text{in}}} \frac{1}{k_i^{\text{out}}} \approx k_j^{\text{in}}\langle 1/k\rangle^{\text{out}} \approx k_j^{\text{in}}\langle 1/k\rangle^{\text{in}}. \tag{A10}$$

Plugging the last approximation into the stationary density equation (A8), one will see that it is an identity.

We should mention that, although network averages of the reciprocals of in- and outdegrees are nearly equal, the biasing inverse of the outdegrees does not result in a stationary distribution approaching a uniform one. The reason for that is the fact that the sum of inverse of degrees [Eq. (A10)] is always proportional to the indegree of the node j because it accounts for neighbors pointing to node j. Repeating the analysis above by using biasing with the inverse of outdegrees, one can verify that the stationary density condition like Eq. (A8) is not satisfied.

[1] P. Erdos and A. Rényi, Publ. Math. Inst. Hung. Acad. Sci **5**, 17 (1960).
[2] D. J. Watts and S. H. Strogatz, Nature (London) **393**, 440 (1998).
[3] A.-L. Barabási and R. Albert, Science **286**, 509 (1999).
[4] A. Arenas, A. Díaz-Guilera, J. Kurths, Y. Moreno, and C. Zhou, Phys. Rep. **469**, 93 (2008).
[5] R. Olfati-Saber, J. A. Fax, and R. M. Murray, Proc. IEEE **95**, 215 (2007).
[6] M. E. J. Newman, Phys. Rev. E **66**, 016128 (2002).

[7] B. D. Hughes, *Random Walks and Random Environments, Volume 1: Random Walks* (Clarendon, Oxford, 1995).
[8] G. M. Viswanathan, S. V. Buldyrev, S. Havlin, M. Da Luz, E. Raposo, and H. E. Stanley, Nature (London) **401**, 911 (1999).
[9] L. Bachelier, Annales Scientifiques de l'École Normale Supérieure, **17**, 21 (1900).
[10] T. Jia, D. Wang, and B. K. Szymanski, Nat. Hum. Behav. **1**, 0078 (2017).
[11] N. Masuda, M. A. Porter, and R. Lambiotte, Phys. Rep. **716**, 1 (2017).
[12] S. Brin and L. Page, Comput. Networks **30**, 107 (1998).
[13] S. Carmi, R. Cohen, and D. Dolev, Europhys. Lett. **74**, 1102 (2006).
[14] M. Boguna, D. Krioukov, and K. C. Claffy, Nat. Phys. **5**, 74 (2009).
[15] M. Rosvall and C. T. Bergstrom, Proc. Natl. Acad. Sci. USA **105**, 1118 (2008).
[16] J. Kleinberg, in *STOC '00: Proceedings of the Thirty Second Annual ACM Symposium on Theory of Computing* (ACM, New York, NY, USA, 1999), pp. 163–170.
[17] J. Kleinberg, in *International Congress of Mathematicians Vol. III* (Eur. Math. Soc., Zürich, 2006), pp. 1019–1044.
[18] C. Borgs, M. Brautbar, J. Chayes, S. Khanna, and B. Lucier, *Internet and Network Economics* (Springer, 2012), pp. 406–419.
[19] Z. Burda, J. Duda, J.-M. Luck, and B. Waclaw, Phys. Rev. Lett. **102**, 160602 (2009).
[20] A. P. Riascos and J. L. Mateos, Phys. Rev. E **86**, 056110 (2012).
[21] A. Avena-Koenigsberger, X. Yan, A. Kolchinsky, P. Hagmann, O. Sporns *et al.*, PLoS Comput. Biol. **15**, e1006833 (2019).
[22] S. Shioda, ACM SIGMETRICS Perform. Eval. Rev. **42**, 21 (2014).
[23] A. Fronczak and P. Fronczak, Phys. Rev. E **80**, 016107 (2009).
[24] M. Bonaventura, V. Nicosia, and V. Latora, Phys. Rev. E **89**, 012803 (2014).
[25] C. M. Grinstead and J. L. Snell, *Introduction to Probability* (American Mathematical Society, Providence, 1997).
[26] J. D. Noh and H. Rieger, Phys. Rev. Lett. **92**, 118701 (2004).
[27] V. Tejedor, O. Bénichou, and R. Voituriez, Phys. Rev. E **80**, 065104(R) (2009).
[28] Networkx: Software package for complex networks in python language, https://networkx.github.io/.
[29] R. M. Karp, in *Complexity of Computer Computations* (Springer, 1972), pp. 85–103.
[30] D. J. MacKay, *Information Theory, Inference and Learning Algorithms* (Cambridge University Press, 2003).
[31] J. Leskovec, J. Kleinberg, and C. Faloutsos, in *KDD '05: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining* (ACM, New York, NY, USA, 2005), pp. 177–187.
[32] R. West, J. Pineau, and D. Precup, in *Proceedings of the 21st International Joint Conference on Artificial Intelligence* (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2009), pp. 1598–1603.
[33] R. West and J. Leskovec, in *Proceedings of the 21st Annual Conference on World Wide Web, WWW'12* (ACM, 2012), pp. 619–628.