Biased Random Search in Complex Networks

Preprint · June 2020

3 authors:

Lasko Basnarkov
Ss. Cyril and Methodius University in Skopje
56 PUBLICATIONS **318** CITATIONS
SEE PROFILE

Miroslav Mirchev
Ss. Cyril and Methodius University in Skopje
37 PUBLICATIONS **172** CITATIONS
SEE PROFILE

Ljupco Kocarev
Macedonian Academy of Sciences and Arts
411 PUBLICATIONS **14,646** CITATIONS
SEE PROFILE

Biased Random Search in 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
and
[2]*Macedonian Academy of Sciences and Arts, P.O. Box 428, 1000 Skopje, Macedonia*
(Dated: June 16, 2020)

We study two types of biased random walk over complex networks, which are based on local information. In the first approach, the transitions towards neighboring nodes with smaller degrees are favored. We show analytically that for well connected networks, biasing the random walk based on inverse of nodes' degrees leads to a uniform distribution of the visiting frequency, which arguably helps in speeding up the search. The second approach explores a random walk with a one-step memory with two-hop paths arrival balancing. We introduce a framework based on absorbing Markov chains for theoretical calculation of the mean first passage time in random walk with memory and apply it in the second approach. Numerical simulations indicate that both approaches can reduce the mean searching time of the target. The one-step memory based method proved to be better for undirected networks, while the inverse-degree biasing leads to faster search in directed networks.

I. INTRODUCTION

The pursuit for appropriate models of the nontrivial interconnections between the units of real systems has led to the emergence of the complex networks 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 nice review of the topic and long list of references. Large portion of dynamical processes on complex networks like the PageRank algorithm [12], various types of searching [13, 14], or community detection [15] are based on or related to the random walk. 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 then 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 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 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. It was shown that for certain type of small world networks, 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 attempts for modification of the URW aimed for 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 the path integrals, the Maximal Entropy Random Walk was in-

[*] lasko.basnarkov@finki.ukim.mk

troduced as a modification of URW 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]. Combination of the local diffusion and knowledge of the topology has recently been applied for study of routing of neural signals [21]. Biasing of the random walk has been shown to be useful in sampling of complex networks as well [22].

In this work we study two simple algorithms built upon the intuition that searching in a network would be faster if the nodes have nearly equal chance to be visited by the random walker. They are based on local information and short memory and result in reduction of the search time as compared to that of the URW. In the first approach the probability of choosing a node is inversely proportional to its degree, while in the second algorithm the aim is to make the probabilities of visiting of the second neighbors nearly equal. Thus, they could be placed somewhere between the memory and computationally intensive optimal algorithms from one side and less demanding, but slow ones at the opposite. The potential of searching improvement of both algorithms is studied with numerical simulations. We provide some arguments that by the first algorithm, which is degree-based biased RW, the searching is approaching to the optimal one, when each node has many neighbors. Numerical simulations show that when the last condition is not satisfied, biasing does not improve the searching. This approach is successful even for directed networks when biasing is based on reciprocals of the indegrees. To the best of our knowledge, a study about searching improvement in directed networks does not exist. Furthermore, we provide theoretical basis for studying searching with random walk with memory of certain length. It is applied in the second algorithm in this work to derive analytical expression for the MFPT. The potential application of these strategies would be in searching in networks with a dynamic or not completely known structure, such as wireless sensor networks [23] and unstructured peer-to-peer networks [24], or in some other very large systems where a full knowledge of the network structure can not be obtained to allow a shortest path algorithms application.

The remainder of the text is organized as follows. In Section II we present theoretical analysis of the first approach of biasing of the random walk that is based on inverse degree. In the following Section III we introduce the framework for studying search with random walk with memory and apply it on the second algorithm. The numerical results are provided in Section IV. The paper finishes with the conclusions.

II. MEMORYLESS SEARCH IN COMPLEX NETWORKS

Random walk on complex networks where the transition probability does not depend on the past represents a Markov chain. The studies of Markov chains have longer history than those of complex networks and one can conveniently use results known for the former while studying the properties of the latter. Here we have opted first to present a derivation of an exact expression of the MFPT between a pair of nodes in a network, based on generating functions [7], which was successfully applied for random walk on lattice. Then, it will be appended with explanation how MFPTs between all pairs of nodes within a network can be calculated simultaneously [25], by method borrowed from the Markov chains theory.

Consider general (not necessarily uniform) random walk on strongly connected directed or connected undirected network, which means that each node can be reached by any other node. Denote by $p_{i,j}^{(n)}$ the probability that random walker which started from node i is located at node j after n steps. The random walk is initialised with the starting probabilities $p_{i,i}^{(0)} = 1$ and $p_{i,j}^{(0)} = 0$ for $i \neq j$. Assuming the same starting node i, let the probability that the random walker will be at location j for the first time after n steps be $f_{i,j}^{(n)}$. These two probabilities are related with the following relationship

$$p_{i,j}^{(n)} = \sum_{m=0}^{n} f_{i,j}^{(m)} p_{j,j}^{(n-m)}, \tag{1}$$

which simply considers all possible scenarios for reaching the node j for the first time at some earlier moment m and then returning to the same node within the remaining $n-m$ steps. To account for the situation when $i = j$ one can augment the last relationship and obtain

$$p_{i,j}^{(n)} = \delta_{i,j}\delta_{n,0} + \sum_{m=0}^{n} f_{i,j}^{(m)} p_{j,j}^{(n-m)}, \tag{2}$$

by using the Kronecker symbol $\delta_{i,j}$.

It is sometimes more convenient to handle a sequence by using generating function

$$F(s) = \sum_{n=0}^{\infty} f^{(n)} s^n, \tag{3}$$

which is power series with the terms of the sequence $f^{(n)}$ taken as coefficients. In order to ensure convergence for sequences with finite terms, the variable s is restricted to $s < 1$. The respective generating functions of the visiting, or node occupation probability $p_{i,j}^{(n)}$ and the probability of first arrival $f_{i,j}^{(n)}$ are

$$P_{i,j}(s) = \sum_{n=0}^{\infty} p_{i,j}^{(n)} s^n,$$
$$F_{i,j}(s) = \sum_{n=0}^{\infty} f_{i,j}^{(n)} s^n. \tag{4}$$

One can multiply the equation (2) by s^n and then sum up over all moments n. Then at left hand side of such sum will appear the generating function of the occupation probability, while at right product of it and that of the first arrival probability and a Kronecker delta

$$P_{i,j}(s) = \delta_{i,j} + P_{j,j}(s) F_{i,j}(s). \tag{5}$$

One can see that the generating function of the first arrival probability can be calculated as

$$F_{i,j}(s) = \frac{P_{i,j}(s) - \delta_{i,j}}{P_{j,j}(s)}. \tag{6}$$

By using the probabilities of first arrival at moments n one can express the MFPT between the nodes i and j as

$$m_{i,j} = \sum_{n=0}^{\infty} n f_{i,j}^{(n)}. \tag{7}$$

From the definition of the generating function of the first arrival probability (4), one can notice that the MFPT can be determined from it as well by using the relationship

$$m_{i,j} = \lim_{s \to 1^-} \frac{dF_{i,j}(s)}{ds}. \tag{8}$$

We note that the usage of limit is needed, because for some sequences which do not converge to zero, like $p_{i,j}^{(n)}$, the generating function is not defined for $s = 1$.

For strongly connected directed networks, as well as connected undirected networks, when the number of steps goes to infinity, $n \to \infty$, the node visiting probability converges to the stationary one, regardless of the starting node i [11]

$$\lim_{n \to \infty} p_{i,j}^{(n)} = w_j. \tag{9}$$

Then one can introduce the sequence of relaxation of occupation probability towards the stationary value

$$r_{i,j}^{(n)} = p_{i,j}^{(n)} - w_j. \tag{10}$$

Using the generating function of the stationary probability

$$\sum_{n=0}^{\infty} w_j s^n = \frac{w_j}{1-s}, \tag{11}$$

one can express the generating function of the occupation probability as

$$P_{i,j}(s) = R_{i,j}(s) + \frac{w_j}{1-s}, \tag{12}$$

where the $R_{i,j}(s)$ is the generating function of the relaxation of visiting probability

$$R_{i,j}(s) = \sum_{n=0}^{\infty} \left(p_{i,j}^{(n)} - w_j \right) s^n. \tag{13}$$

Finally, one can substitute the generating functions $P_{i,j}(s)$ and $P_{j,j}(s)$ by using the relationship (12) and obtain

$$F_{i,j}(s) = \frac{R_{i,j}(s) + \frac{w_j}{1-s} - \delta_{i,j}}{R_{j,j}(s) + \frac{w_j}{1-s}}. \tag{14}$$

If one calculates the first derivative of the last expression with respect to s, and finds its value at the limit $s \to 1^-$, the MFPT will be given as

$$m_{i,j} = \frac{R_{j,j} - R_{i,j}}{w_j}, \tag{15}$$

where we have used the sums

$$R_{i,j} = \sum_{n=0}^{\infty} \left(p_{i,j}^{(n)} - w_j \right). \tag{16}$$

The derivation of the same expression (15) for lattices, with the generating functions formalism can be found in [7], while for complex networks it was obtained previously in [26, 27] by using Laplace transform. The mean return time can be obtained by taking $i = j$ in equation (6), and then repeating the procedure of taking derivative by s and then the limit $s \to 1^-$, which will result in

$$m_{i,i} = \frac{1}{w_j}. \tag{17}$$

The last result is easy to understand since the stationary probability corresponds to the visiting frequency of perpetual random walk, which in turn is inversely proportional to the mean period of return. These results are well known for the MFPT in random walks on complex networks for more than a decade, and for latices even more.

Direct calculation of the MFPT from (15) needs a method for calculation of the infinite sums (16) for each source-target pair. We will present a more convenient method for calculation of all MFPTs in a network, which can be found for example in [25]. To start with, observe first that powers of the transition matrix of the random walk on the network $\mathbf{P}^n$ contain information for node occupation probabilities. More precisely, the probability of the walker to be at state j when it started from i after n steps, $p_{i,j}^{(n)}$, is the respective term of the n-th power $\mathbf{P}^n$. The stationary probabilities w_j constitute the stationary row vector which is left eigenvector of the transition matrix $\mathbf{w} = \mathbf{wP}$, that corresponds to the largest eigenvalue. One can construct a matrix $\mathbf{W}$ with all rows identical to this vector, which possesses two properties summarized as follows

$$(\mathbf{WP})^n = \mathbf{W}^n = \mathbf{W}. \tag{18}$$

The first property $(\mathbf{WP})^n = \mathbf{W}^n$ is due to the definition of the rows of the matrix $\mathbf{W}$ as eigenvectors of $\mathbf{P}$. The second property means that the matrix of the stationary distribution is idempotent, $\mathbf{W}^n = \mathbf{W}$. To realize that, one should see first that in the product $\mathbf{W}^2$, the i,j-th entry is the scalar product of $\mathbf{w}$ with column vector with all elements equal to w_j. This scalar product, due to the normalization $\sum w_i = 1$ has value w_j. This reasoning holds for all pairs i and j, which proves that $\mathbf{W}$ is idempotent. One can notice that the n-th term in the sum $R_{i,j}$ (16) is the respective element of the difference of matrices $\mathbf{P}^n - \mathbf{W}$. In order to obtain a way to account for all possible terms n, one could use the following relationship [25]

$$\begin{aligned}(\mathbf{P} - \mathbf{W})^n &= \sum_{i=0}^{n} (-1)^i \binom{n}{i} \mathbf{P}^{n-i}\mathbf{W}^i \\ &= \mathbf{P}^n + \left[\sum_{i=0}^{n} (-1)^i \binom{n}{i}\right] \mathbf{W}, \end{aligned} \tag{19}$$

which can be shown by using the properties (18). If one further makes binomial expansion of $(1-1)^n$, it will be obtained that the sum in the square brackets in the last expression equals -1. Then one has

$$\mathbf{P}^n - \mathbf{W} = (\mathbf{P} - \mathbf{W})^n. \tag{20}$$

Now, the sum of relaxation of node occupation probabilities between the nodes i as source and j as target (16) can be obtained as the respective term from the infinite sum of matrices which represents the fundamental matrix

$$\mathbf{Z} = \mathbf{I} + (\mathbf{P} - \mathbf{W}) + (\mathbf{P} - \mathbf{W})^2 + \ldots \tag{21}$$

The fundamental matrix $\mathbf{Z}$ as power series could be expressed in more compact form as

$$\mathbf{Z} = (\mathbf{I} - \mathbf{P} + \mathbf{W})^{-1}. \tag{22}$$

The last relationship is result of the fact that $\lim_{n\to\infty} \mathbf{P}^n = \mathbf{W}$. Thus, the fundamental matrix can be used to calculate the MFPTs between all pairs of nodes. The complexity involved is due to the calculation of matrix inverse. To summarize, the MFPT from initial node i, to the target j through the elements of the fundamental matrix $z_{i,j}$ is

$$m_{i,j} = \frac{z_{j,j} - z_{i,j}}{w_j}. \tag{23}$$

As it is defined the MFPT is a property of the network parameterized by two nodes – the starting one i and the final j. A related property of one node only is obtained by averaging all MFPTs starting from all other nodes and targeting it

$$g_i = \frac{1}{N} \sum_{j=1}^{N} m_{j,i}. \tag{24}$$

In the literature it was called Global Mean First Passage Time – GMFPT [28]. 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 Graph MFPT (GrMFPT) [29].

A. Inverse degree biasing in undirected networks

Although the MFPT between any two nodes (23) in a network is associated with the elements of the fundamental matrix (22) through simple relationship, the elements of the latter depend on the whole network structure in a nontrivial way. Thus, it cannot be easily deduced how any intervention in local navigation rules can improve the searching time. One can notice however, that the denominator in (23) represents the stationary distribution of visits by the walker, on which it is easier to influence as we will see. In many natural and artificial networks the majority of nodes have few neighbors, while small number of nodes are much better connected. Thus, increasing the visiting frequency of the less connected nodes would decrease the MFPT towards them and consequently decrease the average GrMFPT. Clearly, any bias of the random walk aiming at increasing the denominator, could modify the numerator as well, but one could hope that the improvement by increasing the visiting frequency has dominant effect on the MFPT. By using ensemble average of networks, in an earlier work [27], it was obtained that biasing random walk by taking transition probability toward a neighbor to be proportional to the inverse if its degree, results in uniform visiting frequency. Also, extensive numerical studies of a more general setting, when the transition probability to a node is proportional to the power of its degree k_j^α, the best performance for searching in uncorrelated networks was obtained for $\alpha = -1$ [29]. Inspired by these results, we aim to study the conditions when the stationary distribution can be made uniform by considering single network. Moreover, as will be seen, the same approach can be applied for directed networks as well. Also, such distribution of visiting frequency is accompanied with improved searching as compared to the URW.

Inverse degree biasing of the random walk which we consider is given by taking transition probability of a node i to its neighbor j to be inversely proportional to its degree $p_{ij} \sim 1/k_j$. The normalization condition $\sum_{j\in\mathcal{N}_i} p_{ij} = 1$ where $\mathcal{N}_i$ is the set of neighbors of node i, will result in transition probability

$$p_{ij} = \frac{1/k_j}{\sum_{l\in\mathcal{N}_i} 1/k_l}. \tag{25}$$

Define node-centric, local average of the reciprocal of degrees of the neighbors as

$$\langle 1/k \rangle_i = \frac{1}{k_i} \sum_{l \in \mathcal{N}_i} 1/k_l, \tag{26}$$

where the subscript i in the average denotes that it is calculated only over the set of the neighbors of the node i. Then the normalization sum in Eq. (25) can be expressed through the local average as

$$\sum_{l \in \mathcal{N}_i} 1/k_l = k_i \langle 1/k \rangle_i . \tag{27}$$

Now, consider networks where most of the nodes are well connected, which technically means $k_i \gg 1$. Then, for uncorrelated networks, or those where the degree of any node is independent on the degrees of its neighbors, the local average can be approximated with the network average

$$\langle 1/k \rangle_i \approx \langle 1/k \rangle = \frac{1}{N} \sum_{j=1}^{N} 1/k_j \tag{28}$$

Then, the normalization sum appearing in the denominator in (25) can be conveniently expressed through the network average as

$$\sum_{l \in \mathcal{N}_i} 1/k_l \approx k_i \langle 1/k \rangle . \tag{29}$$

With this approximation the stationary distribution satisfies the set of equations

$$w_j = \sum_{i \in \mathcal{N}_j} \frac{1/k_j}{k_i \langle 1/k \rangle} w_i = \frac{1/k_j}{\langle 1/k \rangle} \sum_{i \in \mathcal{N}_j} \frac{w_i}{k_i}. \tag{30}$$

By using the approximation (29) one can verify that $1/N$ is the solution of the implicit equations appearing in (30).

Thus, we have shown that inverse degree biasing of the RW, for networks with good connectivity leads to approximately constant invariant density of visiting frequency. As we will verify numerically later on, this results in searching in complex networks that is nearly optimal with GrMFPT approaching to N – the number of nodes in the network. We note that, inverse degree biasing is preferable choice for fastest random search for every complex network where all nodes have with big enough degree. Although in narrower context, it extends the previous findings [27], where ensemble averaging leads to the same result. Also, as will be shown later, for less connected networks the stationary distribution significantly deviates from the uniform, which furthermore supports our analysis. In such situation, this observation is accompanied with worsening of the search as compared to the uniform case.

It seems that the limit of searching propensity of any undirected connected network is $G \geq N$. However, there is deterministic strategy that is twice faster and which holds for graphs that have a Hamiltonian cycle. It is a path 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 [30]. 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 node, one can easily show that GMFPT and GrMFPT will be $N/2$.

B. Inverse degree biasing in directed networks

The same approximations for the invariant density can be applied for directed networks as well, but with additional condition. First, note that by following the intuition that improvement of GrMFPT can be obtained by equalizing the chances of visiting the nodes, one should try by assigning higher probabilities to nodes with smaller indegree. Thus, the transition probability of the biased random walker would be taken as

$$p_{ij} = \frac{1/k_j^{\text{in}}}{\sum_{l \in \mathcal{N}_i^{\text{out}}} 1/k_l^{\text{in}}}, \tag{31}$$

where $\mathcal{N}_i^{\text{out}}$ denotes the set of neighbors of the node i toward which it points to. Furthermore, for networks with good connectivity, one can make similar approximation

$$\sum_{l \in \mathcal{N}_i^{\text{out}}} 1/k_l^{\text{in}} \approx k_i^{\text{out}} \langle 1/k \rangle^{\text{in}} , \tag{32}$$

where $\langle 1/k \rangle^{\text{in}}$ denotes network average of the reciprocal of indegrees

$$\langle 1/k \rangle^{\text{in}} = \frac{1}{N} \sum_{j=1}^{N} 1/k_j^{\text{in}}. \tag{33}$$

The last approximation is obtained with the same reasoning as the one for the undirected networks [refer to Eqs. (28] and (29)). The stationary distribution satisfies equation similar to that for the undirected networks

$$w_j = \sum_{i \in \mathcal{N}_j^{\text{in}}} \frac{1/k_j^{\text{in}}}{k_i^{\text{out}} \langle 1/k \rangle^{\text{in}}} w_i = \frac{1/k_j^{\text{in}}}{\langle 1/k \rangle^{\text{in}}} \sum_{i \in \mathcal{N}_j^{\text{in}}} \frac{w_i}{k_i^{\text{out}}}. \tag{34}$$

If one assumes that the invariant density is constant $w_i = 1/N$, then from Eq. (34) one would have

$$\frac{1}{N} \approx \frac{1/k_j^{\text{in}}}{N \langle 1/k \rangle^{\text{in}}} \sum_{i \in \mathcal{N}_j^{\text{in}}} \frac{1}{k_i^{\text{out}}}. \tag{35}$$

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

$$\langle 1/k \rangle^{\text{in}} \approx \langle 1/k \rangle^{\text{out}} . \tag{36}$$

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

$$\sum_{i\in\mathcal{N}_j^{\mathrm{in}}} \frac{1}{k_i^{\mathrm{out}}} \approx k_j^{\mathrm{in}} \left\langle 1/k \right\rangle^{\mathrm{out}} \approx k_j^{\mathrm{in}} \left\langle 1/k \right\rangle^{\mathrm{in}}. \quad (37)$$

Plugging the last approximation in the stationary density equation (35), one will see that it is identity. We should mention that although network averages of the reciprocals of in- and outdegrees are nearly equal, the biasing inverse to the outdegrees does not result in a stationary distribution approaching to uniform one. The reason for that is the fact that the sum of inverse of degrees (37) is always proportional to the indegree of the node j because it accounts for neighbors pointing to the node j. By repeating the analysis in this section, and using biasing with inverse of outdegrees, one can verify that the stationary density condition like (35) is not satisfied.

III. SEARCH IN COMPLEX NETWORKS WITH MEMORY OF ONE STEP

When the transition probabilities towards the neighbors are conditioned on the nodes visited in the past, then the random walk is not Markov chain. However, if the random process has a memory of finite length, or finite history length, it can be conveniently expressed by appropriate Markov chain. We will present here the case when the memory has length one, but it can be generalized to longer periods in a similar manner. Let the walker at certain step has moved from node r to its neighbor s. When the random walk has a memory of one step, the probability of proceeding towards some neighbor t from s, depends only on the previously visited node r, but not on the preceding ones

$$p(t|s, r, u_1, u_2, \dots) = p(t|s, r), \quad (38)$$

where $u_1, u_2, \dots$ is the sequence of nodes visited before r. Then one can make a Markov chain consisting of states which are pairs of neighboring nodes. To make the connection between the random walk and the related Markov chain more intuitive, denote with rs the state in the Markov chain when the random walker has visited node r immediately before s. The transition probabilities in the chain from state rs to st are thus $p_{rs,st} = p(t|s, r)$, which can be organized in the respective transition probability matrix $\mathbf{P}$. Generalization of the procedure for making Markov chain for random walk with longer memory is straightforward, but one should keep in mind that the size of the corresponding matrix will rise exponentially.

The starting step of random walk with memory is not specified by this kind of walk. 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. Finding the 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 [31]. Thus, the MFPT from node i to j would be related to the mean times to absorption (MTA) of random walk in absorbing Markov chain that started in any state ir, where the absorbing states are sj. Before deriving that relationship, we will first present some well known results about the MTA in absorbing Markov chains. [25]. For such purpose, one should first determine the transition matrix of the absorbing Markov chain. It depends on the target j, and thus we will denote it with $\mathbf{P}_j$ Since the random walk should stop at any absorbing state, the probability of leaving it is zero. Also, the transitions between other states are the same as in the ordinary chain. Thus the absorbing chain matrix $\mathbf{P}_j$ differs from the general matrix $\mathbf{P}$ only in the rows corresponding to the first index sj, which in the absorbing matrix have values $p_{sj,rt} = \delta_{sj,rt}$. It is convenient to write the transition matrix of the absorbing Markov chain related to the target node j in the form

$$\mathbf{P}_j = \begin{vmatrix} \mathbf{Q}_j & \mathbf{R}_j \\ \mathbf{0}_j & \mathbf{I}_j \end{vmatrix}, \quad (39)$$

In the last equation $\mathbf{Q}_j$ is a matrix corresponding to the transient states which correspond to all pairs of neighboring nodes rs in the network, where neither r nor s is the target node j. The submatrix $\mathbf{R}_j$ corresponds to the rows with transient states rs as before while the columns sj consist of all absorbing states, which lead to the target j. We remind that the transition probabilities in these two matrices are the same as the respective ones in the original chain. Lastly, the zero matrix $\mathbf{0}_j$ and the identity matrix $\mathbf{I}_j$ denote that from the absorbing states sj the random walk can not move further and remains in the same state. To simplify the notation, we will omit the index j, and use Greek letters α and β to identify states instead of using pairs of neighboring network nodes.

The MTA equals the average number of steps while the walker is in the transient states. Similarly to the analysis of the MFPT of memoryless walk, the powers of the transient submatrix $\mathbf{Q}^n$ contain the probabilities $q_{\alpha\beta}^{(n)}$ of the walker which started at state α to be at β after n steps. We introduce a binary random indicator variable $I_\beta^{(n)}$, which has value 1, if the walker is present at state β, and 0 if it is absent. Its expected value $E(I_\beta^{(n)})$ equals the probability $q_{\alpha\beta}^{(n)}$. The expected number of steps when the walker is present at β in the first n iterations is

$$E(I_\beta^{(0)} + I_\beta^{(1)} + \dots + I_\beta^{(n)}) = q_{\alpha\beta}^{(0)} + q_{\alpha\beta}^{(1)} + \dots + q_{\alpha\beta}^{(n)}. \quad (40)$$

The expected number of visits of the state β, for an infinite walk is obtained by simply letting $n \to \infty$. One can introduce a fundamental matrix for this absorbing Markov chain as the infinite sum

$$\mathbf{Y} = \mathbf{I} + \mathbf{Q} + \mathbf{Q}^2 + \cdots. \quad (41)$$

It contains the expected number of steps of perpetual random walk starting from state α (the row) which is present at node β (the column). The mean time to absorption of random walk which started at state α equals the mean number of steps in which the walker is in *any* transient state β, or the sum

$$\mu_{\alpha,\beta} = \sum_{\beta} y_{\alpha,\beta}, \tag{42}$$

where $y_{\alpha,\beta}$ are the elements of the fundamental matrix of the absorbing chain $\mathbf{Y}$. More compact expression can be obtained by using the vector consisting of all mean times to absorption μ from any starting state α, by using the matrix equation

$$\mu = \mathbf{Y}\mathbf{c}, \tag{43}$$

where $\mathbf{c}$ is a column vector with all elements equal to one. Because any random walk finishes in the absorbing state with a probability one, the powers of the matrix $\mathbf{Q}^{(n)}$ become vanishingly small as $n \to \infty$. Accordingly, the infinite sum of matrices converges and the fundamental matrix can be represented as

$$\mathbf{Y} = \left(\mathbf{I} - \mathbf{Q}\right)^{-1}. \tag{44}$$

Similarly to calculation of MFPTs in random walk without memory, the MTA can be obtained from the transition matrix by using matrix inverse only. We remind that because the whole analysis was performed by choosing particular node j as target, we can explicitly use the respective index in the equation for determination of the mean time to absorption

$$\mu_j = \mathbf{Y}_j \mathbf{c}_j. \tag{45}$$

As we will see below, μ_j contains sufficient information for calculation of MFPTs from all nodes to a particular target j, which are not its neighbors.

The MFPT between two nodes is by definition weighted sum of the lengths l of all paths $\mathcal{P}_{i,j}$ that start at the initial node i and finish at the target j and which visit j only at the last step

$$m_{i,j} = \sum_{\mathcal{P}_{i,j}} l\left(\mathcal{P}_{i,j}\right) p\left(\mathcal{P}_{i,j}\right), \tag{46}$$

where $p(\mathcal{P}_{i,j})$ is the probability of occurrence of the path $\mathcal{P}_{i,j}$. Let us first consider the case when the target is not neighbor of the source. The sum in the last expression can be organized by summing over all paths that have the neighbor s of the starting node after the first step, and then summing over all such neighbors s

$$m_{i,j} = \sum_{s \in \mathcal{N}_i} p_{is} \sum_{\mathcal{P}_{s,j}} \left[1 + l(\mathcal{P}_{s,j})\right] p(\mathcal{P}_{s,j}), \tag{47}$$

where p_{is} denotes the probability to jump from i to s in the first step. Since all the paths that we consider eventually reach the target j, the normalization condition of the probability reads

$$\sum_{\mathcal{P}_{s,j}} p(\mathcal{P}_{s,j}) = 1. \tag{48}$$

One can also note that the MFTP from the neighbor s to the target j by pursuing the memory-based random walk equals the MTA from the starting state is in the Markov chain, which is the respective term from the MTA vector μ_j

$$\mu_{is,j} = \sum_{\mathcal{P}_{s,j}} l(\mathcal{P}_{s,j}) p(\mathcal{P}_{s,j}). \tag{49}$$

When the neighbor in the first step is chosen uniformly one has $p_{is} = 1/d_i$. Then 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

$$m_{i,j} = 1 + \frac{1}{d_i} \sum_{s \in \mathcal{N}_i} \mu_{s,j}. \tag{50}$$

We can now consider the case when the target j is neighbor of i. This target could be reached in one step with probability $p_{ij} = 1/d_i$, by pursuing the shortest path, while for the mean number of steps through all other paths one can apply the same reasoning as above. We note that in the sum running in the neighborhood of the initial node i, the target j should be omitted. Then by using the normalization condition (48), one can obtain that

$$\sum_{\substack{s \in \mathcal{N}_i \\ s \neq j}} p_{is} \sum_{\mathcal{P}_{s,j}} \left[1 + l(\mathcal{P}_{s,j})\right] p(\mathcal{P}_{s,j}) = \frac{d_i - 1}{d_i} + \frac{1}{d_i} \sum_{\substack{s \in \mathcal{N}_i \\ s \neq j}} \mu_{s,j}. \tag{51}$$

Adding the contribution of the direct path to the last expression one will obtain similar result as (50)

$$m_{i,j} = 1 + \frac{1}{d_i} \sum_{\substack{s \in \mathcal{N}_i \\ s \neq j}} \mu_{s,j} \tag{52}$$

By using the trivial value $\mu_{j,j} = 0$, one can see that the same expression (50) can be used for any target, regardless whether it is neighbor to the starting node, or not.

In the previous section, the biasing of RW by taking transition probabilities proportional to the inverse of node degrees aims at increasing the visiting chances to the less connected nodes. This approach is based only on the nearest neighborhood properties – the degrees of the neighbors which are one hop away. Another strategy could be created by trying to make the probabilities to reach the second neighbors as similar as possible.

One intuitive way to make such preference is as follows. Assume that at the previous step the walker was at node r, from where it has jumped to the node s, and in the next step it should visit some node t from the set of neighbors of s. Denote the number of all two-hop paths from node r to t with b_{rt}. 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. One choice for favoring the less accessible second neighbors is the following

$$p_{rs,st} = \frac{\frac{1}{b_{rt}}}{\sum_{u \in \mathcal{N}_s} \frac{1}{b_{ru}}}, \tag{53}$$

where the sum in the denominator is used for normalization of the probabilities. 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 paths. Since for undirected networks every node is a second neighbor to itself, 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 $p_{rs,st}$ and the immediate returning is disfavored.

We note that the related Markov model of random walk with memory could be successfully applied for determination of the stationary distribution of the visiting frequency as well. First one needs to calculate the stationary distribution of the states of the Markov chain $\mathbf{v}$, which is obtained from the stationarity condition $\mathbf{vP} = \mathbf{v}$. It contains the probabilities v_{rs} of the random walk with memory being at state rs. Then the stationary distribution of frequency of visits of the node s, by random walk with memory of one step, would be the sum of all v_{rs}, or v_{st}, within the neighborhood of the node s.

IV. NUMERICAL RESULTS

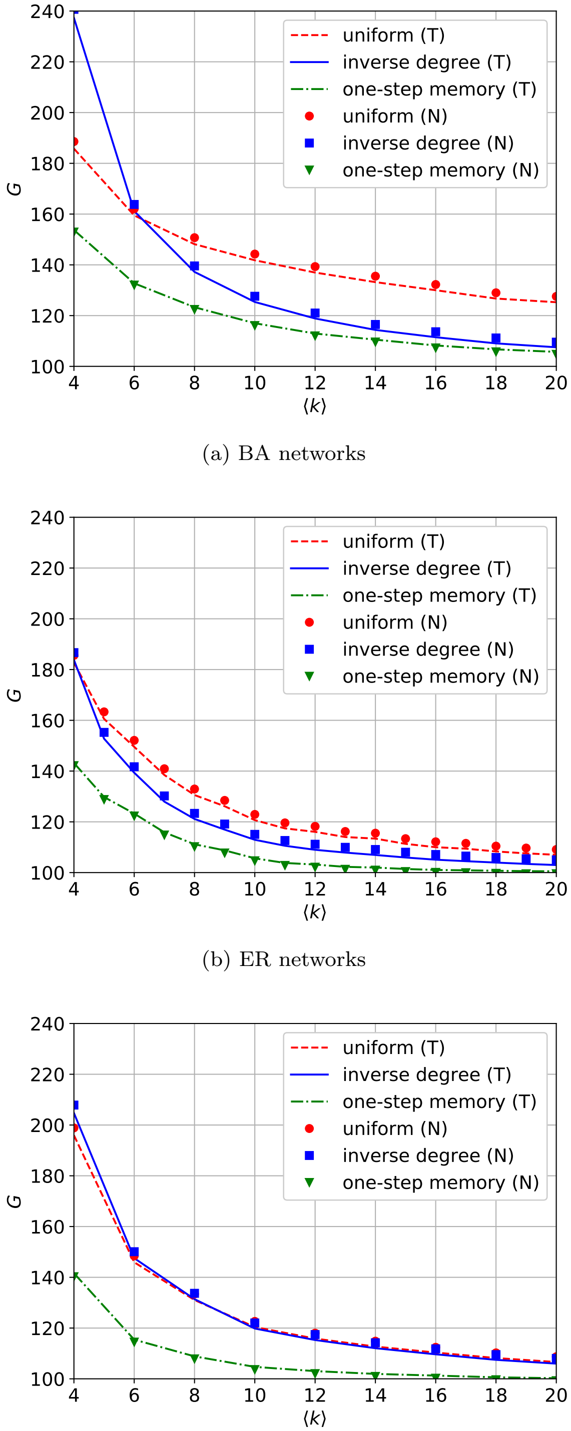
In this section we provide some results obtained with the analytical expressions and numerical simulations with the two biased random walks and compare them with the uniform random walk. First, we conduct a thorough analysis using generic network models, such as random, scale-free and small world networks. 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 effectiveness of the biasing procedures were studied by calculations of the Graph MFPT and the invariant density from the transition matrix, by using the theoretical expressions. Furthermore, numerical simulations of the random walks were performed to verify the results. Since the theoretical expressions involve an inverse matrix operation, the latter presents the major constraint in the calculations. 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 have opted to perform the analyses of the MFPT and the invariant density for networks with $N = 100$ nodes. We have 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 10 network instances for every parameter setting for each network type. Moreover, in the numerical simulations we have 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 [32]. 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 a 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 [33]. Preferential attachment is employed as the probability to connect to an existing node is taken to be proportional to its degree.

In Figure 1a we compare the obtained GrMFPTs for the uniform random walk, the biased random walk with a transition probability proportional to the inverse degree and the memory based one 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. As we previously mentioned, the results were obtained by averaging of 10 networks with certain m. First, we can observe that the numerical (N) and the theoretical (T) results are very close, which confirms the accuracy of the analytical expressions. It can be noticed that when the average node degree is large enough, the two biasing alternatives provide smaller GrMFPT than the uniform random walk, which also approaches the optimal value N. When the majority of the nodes have few neighbors (four or six in this case), the inverse degree biasing worsens the searching in the network. We think that this is probably due to the much higher preference of the weakly connected nodes and consequently decreasing the chances of exploring new areas in the network by avoiding the hubs. The one-step memory approach is better than the other two since it involves memory and biasing which intents to equalize the chances of visiting the second neighbors of a given node. One can notice that all curves decrease asymptotically towards the value corresponding to the number of nodes, which arguably is the minimal possible value for the GrMFPT.

The biasing brings search improvement for the purely random Erdős-Rényi graphs also, as it is shown in Fig-



(a) BA networks

(b) ER networks

(c) WS networks

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/circle), inverse degree (blue line/square) and one-step memory (green line/triangle). The lines are theoretical values (T) and the markers numerical estimates (N).

ure 1b. We generated 10 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 it can be seen the inverse degree biasing gives lower average GrMFPT than the uniform random walk, 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 ones.

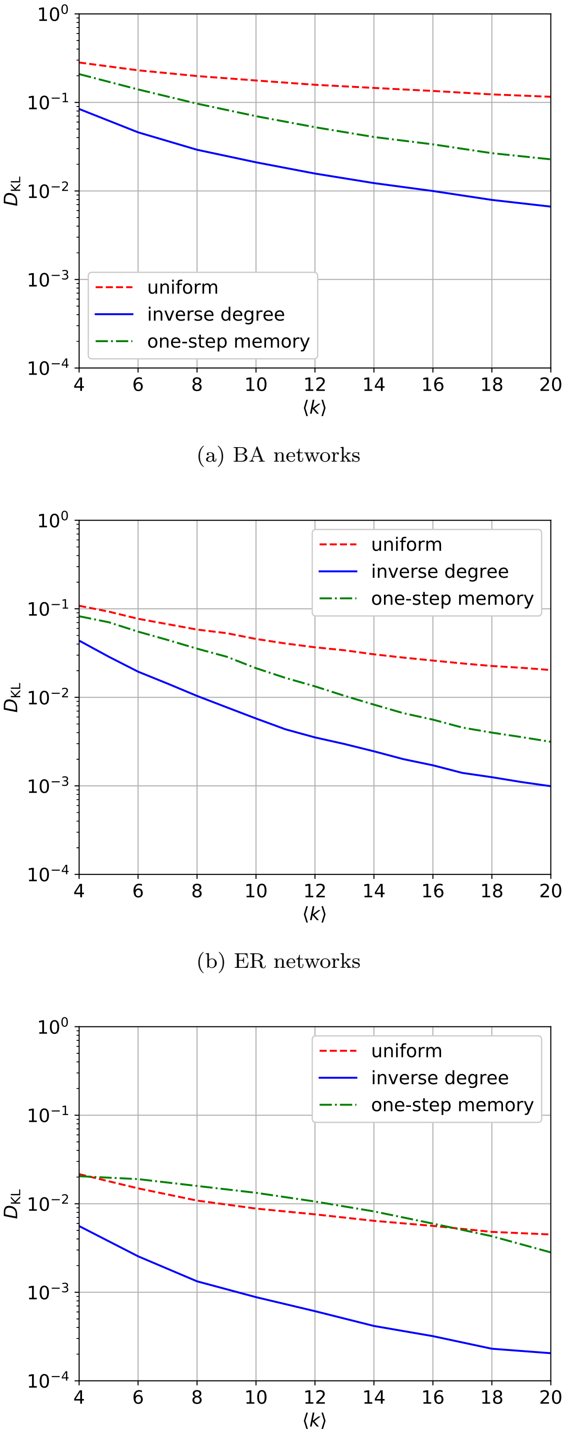
In Figure 1c we show how the biasing affects the random walk in Watts-Strogatz networks, where the rewiring probability is $p = 0.2$. Unlike for the other network types under study, the inverse degree biasing does not improve the GrMFPT. This is probably due to the smaller degree variability in this kind of networks. On the other hand, the one-step memory approach still reduces the GrMFPT, as it 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 the reason behind the search improvement is the flattening of the stationary distribution of the visiting frequency by a perpetual random walker. A convenient quantity for estimating the deviation of one distribution from another is the Kullback-Leibler (KL) divergence [34]. In the case when one has two discrete distributions $P(i)$ and $Q(i)$, it is defined as

$$D_{\mathrm{KL}}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}. \tag{54}$$

One can notice from the definition that this is 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 Figure 2a 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 previous sections 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 shows that the leveling of the visiting frequency is not sufficient for optimizing the search.

Similarly, Figure 2b shows the KL divergence between the flat density and the uniform and both biasing approaches in ER networks. Once again, the biasing yields a density that is closer to the constant one than the uniform random walk, which is probably the reason for the lower GrMFPT obtained in Figure 1b. On the other hand, in WS networks (see Figure 2c) the inverse degree biasing gives a density which is closer to a constant than the uniform random walk, while the one-step memory approach does not, even though it proved fastest in Figure 1c.

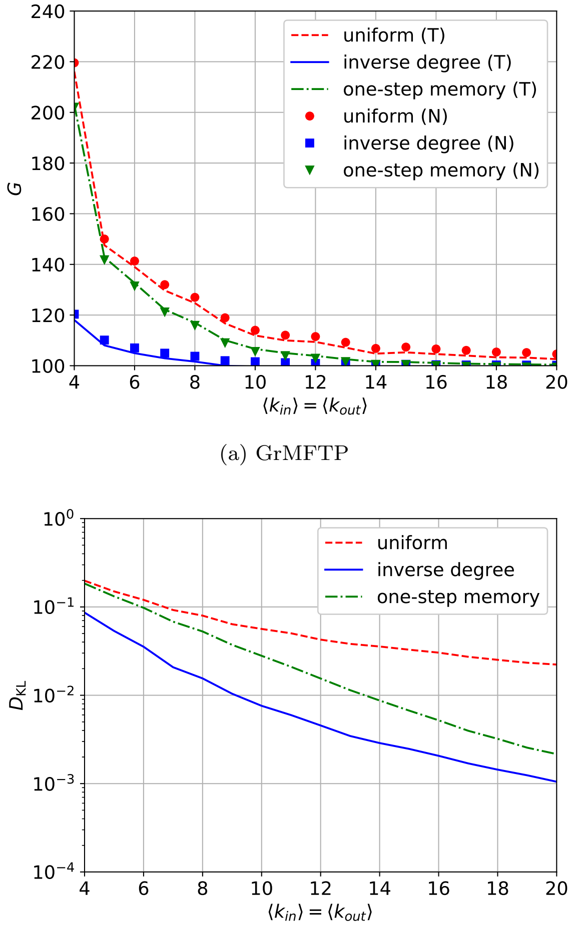


(a) BA networks

(b) ER networks

(c) WS networks

FIG. 2: Kullback-Leibler divergence of the invariant density 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.



(a) GrMFTP

(b) Kullback-Leibler divergence

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 degree (blue squares) and one-stop memory (green triangles); and (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).

We also tested the searching improvement in directed networks and in Figure 3a are shown the GrMFPTs for directed ER networks by uniform, inverse indegree and one-step memory random walks. We can see that the one-step memory provides better results than the uniform random walk, but the inverse indegree approach outperforms them both significantly, which was not the case in undirected networks. Biasing based on inverse outdegree performs slower than the uniform random walk (results are not shown).

The flattening of the invariant density is an ingredient which helps in search improvement in directed networks as well. We have verified that, as expected, for well connected networks when biasing of random walk is based on inverse of indegrees, the invariant density is closer to the constant, than that of a URW. In Figure 3b 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 presented in Section II B.

We have also tried the searching performance of the two 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 2.1.2000, which is an undirected graph consisting of 6474 nodes and 13233 links [35]. Its average node degree is $\langle k \rangle \approx 4$. The second network is an extracted set of Wikipedia pages [36, 37]. The graph is directed and consists of 4592 nodes and 119882 links, from which we take the largest strongly connected component that has 4051 nodes and 119000 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 pairs, while for the extract of the Wikipedia network the averaging is performed with $1.5 \cdot 10^5$ node pairs, out of 4051×4050 possible pairs as 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 worsens the search of the URW, because 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 degree 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)	$1.93 \cdot 10^4$	$1.78 \cdot 10^5$	$1.80 \cdot 10^4$
Wikipedia(extr.)	$3.01 \cdot 10^7$	$1.09 \cdot 10^4$	$8.15 \cdot 10^5$

V. CONCLUSIONS

In this work we studied the potential for improvement of searching in complex networks by applying biasing of the random walk. We have examined two approaches that avoid the hubs by making the transition probabilities proportional to the inverse of the degree of the nodes, or by accounting for the two-hop-paths between the nodes. It was obtained that for very large and well connected networks, the inverse-degree biasing procedure is approaching the optimal searching. This is characterized by average number of steps needed to find the target node that is slightly larger than the number of nodes in the network. The optimal achievement is due to the flattening towards the constant of the invariant density of visits of the nodes by an infinite random walker. The one-stop memory approach has shown searching improvement as well, and although being more complex, it is promising since it shows better performance than the uniform random walk even for networks where the majority of nodes have a small degree.

For theoretical analysis of the random walk with memory of one step was introduced an appropriate absorbing Markov chain model. This model enables obtaining closed form expressions of the MFPT between nodes in the complex network. This approach could be straightforwardly applied for random walk with longer memory as well. The technique with absorbing Markov chain can be also applied for random searching of target when each node knows the identity of its neighbors. In this case the absorbing states would be all neighbors of the target. Finally, we have shown that the inverse degree biasing based on indegree, leads to nearly optimal random search in directed networks as well, which is even better than the memory-based one.

ACKNOWLEDGEMENT

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

[1] P. Erdos and A. Rényi, Publ. Math. Inst. Hung. Acad. Sci **5**, 17 (1960).

[2] D. J. Watts and S. H. Strogatz, Nature **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, Physics reports **469**, 93 (2008).

[5] R. Olfati-Saber, J. A. Fax, and R. M. Murray, Proceedings of the IEEE **95**, 215 (2007).

[6] M. E. Newman, Physical review E **66**, 016128 (2002).

[7] B. Hughes, Oxford **2**, 1995 (1995).

[8] G. M. Viswanathan, S. V. Buldyrev, S. Havlin, M. Da Luz, E. Raposo, and H. E. Stanley, nature **401**, 911 (1999).

[9] L. Bachelier, in *Annales scientifiques de l'École normale supérieure*, Vol. 17 (1900) pp. 21–86.

[10] T. Jia, D. Wang, and B. K. Szymanski, Nature Human Behaviour **1**, 0078 (2017).

[11] N. Masuda, M. A. Porter, and R. Lambiotte, Physics reports **716**, 1 (2017).

[12] S. Brin and L. Page, Computer networks and ISDN systems **30**, 107 (1998).

[13] S. Carmi, R. Cohen, and D. Dolev, EPL (Europhysics Letters) **74**, 1102 (2006).

[14] M. Boguna, D. Krioukov, and K. C. Claffy, Nature Physics **5**, 74 (2009).

[15] M. Rosvall and C. T. Bergstrom, Proceedings of the National Academy of Sciences **105**, 1118 (2008).

[16] J. Kleinberg, *The small-world phenomenon: An algorithmic perspective*, Tech. Rep. (Cornell University, 1999).

[17] J. Kleinberg, in *Proceedings of the International Congress of Mathematicians (ICM)*, Vol. 3 (2006) pp. 1019–1044.

[18] C. Borgs, M. Brautbar, J. Chayes, S. Khanna, and B. Lucier, in *International Workshop on Internet and Network Economics* (Springer, 2012) pp. 406–419.

[19] Z. Burda, J. Duda, J.-M. Luck, and B. Waclaw, Physical review letters **102**, 160602 (2009).

[20] A. Riascos and J. L. Mateos, Physical Review E **86**, 056110 (2012).

[21] A. Avena-Koenigsberger, X. Yan, A. Kolchinsky, P. Hagmann, O. Sporns, *et al.*, PLoS computational biology **15**, e1006833 (2019).

[22] S. Shioda, ACM SIGMETRICS Performance Evaluation Review **42**, 21 (2014).

[23] S. D. Servetto and G. Barrenechea, in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications* (ACM, 2002) pp. 12–21.

[24] C. Gkantsidis, M. Mihail, and A. Saberi, in *IEEE INFOCOM*, Vol. 1 (Citeseer, 2004) pp. 120–130.

[25] C. M. Grinstead and J. L. Snell, *Introduction to probability* (American Mathematical Soc., 2012).

[26] J. D. Noh and H. Rieger, Physical review letters **92**, 118701 (2004).

[27] A. Fronczak and P. Fronczak, Physical Review E **80**, 016107 (2009).

[28] V. Tejedor, O. Bénichou, and R. Voituriez, Physical Review E **80**, 065104 (2009).

[29] M. Bonaventura, V. Nicosia, and V. Latora, Physical Review E **89**, 012803 (2014).

[30] R. M. Karp, in *Complexity of computer computations* (Springer, 1972) pp. 85–103.

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

[32] "Networkx: Software package for complex networks in python language," `https://networkx.github.io/`.

[33] 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_{ij}.

[34] D. J. MacKay, *Information theory, inference and learning algorithms* (Cambridge university press, 2003).

[35] J. Leskovec, J. Kleinberg, and C. Faloutsos, in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining* (2005) pp. 177–187.

[36] R. West, J. Pineau, and D. Precup, in *Twenty-First International Joint Conference on Artificial Intelligence* (2009).

[37] R. West and J. Leskovec, in *Proceedings of the 21st international conference on World Wide Web* (2012) pp. 619–628.