Load balanced diffusive capture process on homophilic scale-free networks

András Telcs[a], Márton Csernai[b,*], András Gulyás[b]

[a] *Dept. of Computer Science and Information Theory, Budapest University of Technology and Economics, 1117 Budapest, Hungary; Dept. of Quantitative Methods, Faculty of Economics, University of Pannonia, 8200 Veszprém, Hungary*
[b] *Dept. of Telecommunications and Media Informatics, Budapest University of Technology and Economics, 1117 Budapest, Hungary*

Abstract

Diffusive capture processes are known to be an effective method for information search on complex networks. The biased N lions-lamb model provides quick search time by attracting random walkers to high degree nodes, where most capture events take place. The price of the efficiency is extreme traffic concentration on top hubs. We propose traffic load balancing provided by type specific biased random walks. For that we introduce a multi-type scale-free graph generation model, which embeds homophily structure into the network by utilizing type dependent random walks. We show analytically and with simulations that by augmenting the biased random walk method with a simple type homophily rule, we can alleviate the traffic concentration on high degree nodes by spreading the load proportionally between hubs with different types of our generated multi-type scale-free topologies.

Keywords:
random walk, diffusive capture process, scale-free network, community structure, information search, load balancing

1. Introduction

First-passage processes have been actively researched due to their prevalence in modeling dynamical natural phenomena [1]. A particular branch of research mainly focuses on a special case of this process, called diffusive capture process (or the moving trap model), where two different particles diffuse in some media until they meet [2]. The diffusing particles are intuitively called lions and lambs, and the process is also referred to as the diffusing predator-prey model. With the imminent proliferation of network science, the dynamical properties of this

*Corresponding Author. Tel: +36-1-463-2187, Fax: +36-1-463-3107
Email addresses: telcs@szit.bme.hu (András Telcs), csernai@tmit.bme.hu (Márton Csernai), gulyas@tmit.bme.hu (András Gulyás)

process on complex networks has been widely researched recently and was linked to many physical and chemical phenomena [3, 4, 5].

From a technological point of view, it is of special interest, how this process could be employed for information search in todays communication networks. The appropriate discovery and localization of resources is a key issue in any kind of networked application. During the last decade resource localization has been addressed in diverse systems with different purposes. The proliferation of distributed algorithms is imminent in P2P networks [6, 7]. On the other hand, we know from many empirical studies that complex social structures are very efficient in resource discovery. Inspired by the related physics literature, the diffusive capture process was proposed for information search in P2P networks. In [8] it has been investigated, how the N lions and lamb (NLL) model can be adapted to information search in a peer-to-peer network setting. NLL is a special case of the diffusive capture process, where N lions (information markup packets) try to catch one lamb (query packets), while both diffuse on the underlying network.

The dynamical properties of this process is closely related to the first passage times of the coupled random walks. Among firsts, [9] showed an analytical formula for the mean first passage time of a random walk between two arbitrary nodes of a scale-free network, and the analysis was generalized to actual first passage times in [10, 11]. Lee et al. in [12] thoroughly analyzed the capture probability of a lamb by several lions performing independent random walks on a scale-free network with tree structure and also on networks with loops. They found that on general scale-free networks which usually contain loops, the capture time is much smaller than the mean first passage time. This effect is due to the phenomenon that both walkers approach the stationary distribution, which is biased towards high degree nodes, hence they meet shortly after they arrive to the subset or close proximity of high degree nodes. Furthermore, it is shown that the capture probability is dependent on the degree exponent of the scale free network.

It is shown in [13] that the search process can be improved in terms of the search time by biasing the random walk to prefer high degree nodes. This makes the diffusing particles to quickly move to the core of the network consisting of the high degree nodes where the capture events take place. The average capture time $\langle T \rangle$ of the RWrs in the the biased NLL (BNLL) model is analyzed in [3, 13] in the specific case of one lion and one lamb. From [13] we know that the capture time $\langle T \rangle$ on looped scale free networks grows with N^{β}, where $\beta = 0.5$ on scale-free networks with $\gamma = 2.4$. In [3] the formula for the probability that two walkers are on the same node is shown to be proportional to $k^{2(\alpha+1)}$, and it it shown that there exists a crossover value $\alpha_c = (\gamma - 3)/2$, where γ is the exponent of the power-law degree distribution of the graph. For $\alpha < \alpha_c$ the lamb survives with a finite probability, and for $\alpha \geq \alpha_c$ the lion surely captures the lamb. Although the biasing modification speeds up the capture process, it puts a high resource allocation burden on the hubs in the network. To the best of our knowledge, no one has addressed this open problem.

We propose a method to improve the BNLL model in terms of balancing the traffic load on the inner core of the network by constructing a scale-free overlay network with embedded community structure. We exploit this clustered topology to balance the load of the core. It is a well-known fact, that many natural complex networks have some kind of community structure, and to identify these clusters is a widely researched area [14, 15, 16]. Furthermore, many graph generation models have been proposed to reproduce the naturally emerging community structure in synthetically built graphs [17, 18, 19, 20, 21, 22, 23, 24]. However, these models try to mimic various specific aspects of the phenomenon and so the graph generation methods are often quite complex. For our purposes, we show a simple overlay graph generation model, where we embed basic homophily structure into the graph. We label the nodes with different *types*. We employ random walks to find nodes where new nodes are connected during the graph generation process. The link generation process is then carefully adjusted to create links between nodes with the same type, and also between nodes with different types. The ratio of *intra-type links* and *inter-type links* is set as an input parameter of our model. By adjusting this parameter we can control the cliquishness of the community structure of the graph, which in turn will have an effect on the search process.

Type dependent random walks are also used in the search process, where the types are assigned also to the random walkers. Since the lions carry the information packets, their type is determined by the type of their origin. On the other hand, the lamb carries the search term, and does a random walk defined by the type associated to the query. The biased random walk transition rule is modified to use the degree based power bias only on the neighboring nodes with the same type as the random walker. This forces the random walkers to prefer nodes with the same type as themselves as their next hops. It is shown that the random walkers diffuse between the subgraphs of the different types of nodes evenly. The load is spread from the top degree nodes to smaller hubs as well, while still maintains the advantageous properties of the model, such as the high capture probability and low capture time.

In our analysis we show that the walks spend almost all their time on nodes with their own type, consequently the search is limited to a subnetwork of size N/t, where N is the number of nodes and t is the number of types in the graph. The gain on the speed is dependent of the ratio of intra-type and inter-type links in the generated multi-type scale-free graphs. With simulations it is found that, for appropriate parameter settings, the average capture time is lower in case of many types than in case of one type, which is the equivalent of the BNLL model with one lion and one lamb. The metric we used to assess our model is the amount of traffic going through the network core consisting of the top hubs. The traffic on the core consisting of a set number of hubs is inversely proportional to the number of types built into the network. We support this result with matching simulation results. One should take into consideration that there is a processing delay on the nodes, because all visiting lamb-lion pairs should be matched. If the traffic is decreased by a factor of t, the processing time is linearly decreasing by t as well.

The rest of this paper is structured as follows. In Section 2 we describe the multi-type scale-free graph generation model and prove that both the typed subgraphs, and the whole graph has power-law distribution to aid the capture process. Section 3 presents the modified biased NLL capture process with the analysis and simulation of capture times. In Section 4 we show the load balancing capabilities of our capture process on the generated multi-type scale-free graphs both analytically and by simulations. Finally, section 5 summarizes our results.

2. Multi-type scale-free graph generation

In this section the method for generating multi-type scale-free graphs (MTSF) is described. We create graphs with inherent homophily structure by assigning different types to nodes and control the edge creation process between nodes with the same type and also between the nodes with different types. Later in this section it is proved, that the degree distribution of both the resulting subgraphs consisting of nodes with same types and also the whole graph follow the power-law. The importance of this structure is that it provides low average capture times for the biased search process. Based on the types we introduce to the graphs, we make the biased capture process to concentrate the random walkers to their corresponding type subgraphs, and this creates natural load balancing in the capture process (detailed in Sec. 4).

We demonstrate the multi-type graph generation process, when there are two types of nodes in the network. Naturally, the process can be easily generalized for an arbitrary number of types. Let $G = (\Gamma, E)$ be an undirected graph. The set of nodes with the two types are referred to as A and B, $\Gamma = A \cup B$. Let $|A|$ denote the number of nodes with type A and $N = |\Gamma|$, the number of all nodes in the network. The degree of a node $x \in A$ is divided into two parts. The first part, labeled as k_x^A, consist of the links from x to neighbor nodes with type A. The other part is the links from x to neighbor nodes with type B denoted as $k_x^{A,B}$. The total degree of x is then $k_x = k_x^A + k_x^{A,B}$. Assume that we insert new nodes with type A and B into the graph with a predefined ratio r, $|A| = rN$, $|B| = (1-r)N$, $(0 < r \leq 1)$. For the sake of simplicity, our multi-type graph is started from a full graph. The size of this initial full graph equals two times the number of types we build into the network, so every type is represented by two nodes at start. This enables the first homophilic walk of the initial nodes to find a target node with the same type. This is a simple architectural choice in the model. The initial conditions of the model depending on the number of types can be further investigated in the future.

The multi-type graph generation is executed by iteratively adding new nodes to the graph. We create a new node y with a random type according to the above defined parameter r. We start RWs on G from a randomly chosen node s which has the same type as y. We run two different kinds of RWs on the network to determine where to connect the new nodes and choose between the two kinds of walks probabilistically. The *homophilic* RW uses links that connect nodes with same type, we choose this one with probability p. The *heterophilic* RW uses links that run between nodes with different types, this one is chosen with probability $1-p$. The new node is connected to the node where the RW is at after a defined step length m. This process is repeated until the new node has a predefined number of links.

One should note that the walk is not restarted again and again from the same node s but from a uniformly randomly selected new one. In this way the local nature of the algorithm is not violated. In [26] the effect of choice of the number of steps (denoted by l there) is discussed in details, investigating its effect on clustering. They claim that for $m >> 1$, the original BA model with low clustering is attained. Since our main concern is the proper generation of the graph, low m values are not effecting our main goals. Note that for low m, the network generation has memory, which causes degree correlation and clustering, but an easy alternation of the algorithm may increase the memory loss. Instead of choosing uniformly a new node each round, after finishing the insertion of edges for a node, we do M extra steps with the walk and the endpoint will be the initial node for the next round ($M >> m$). This method provides very good mixing between node insertion.

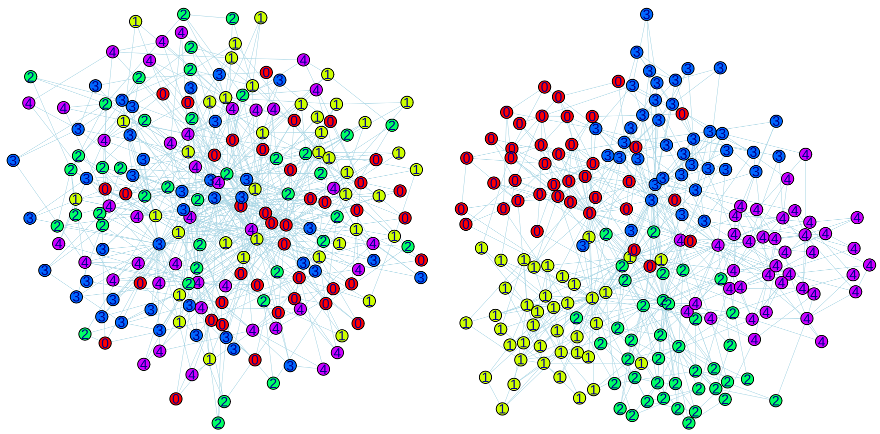
Let us denote the number of links between nodes with type-A as $|E_A|$. Also, $|E_B|$ is the number of links between nodes with type-B. We denote the number of inter-type edges between type-A and type-B nodes that are created by type-A nodes with $|E_{A,B}|$. Similarly, $|E_{B,A}|$ denotes the number of edges between type-A and type-B nodes that are created by type-B nodes. Note that, we can create new nodes with different types according to the parameter r, so $|E_{A,B}|$ and $|E_{B,A}|$ might not always be equal. Using these notations, the total number of inter-type links equals $|E_{A,B}| + |E_{B,A}|$. Also, $|E_A|$ and $|E_B|$ will be the number of type A and B intra-type links respectively. With this method we can generate topologies, where the ratio of intra-type links (between same types of nodes) and inter-type links (between different types of nodes) can be set arbitrarily with the parameter p independently of the parameter r, which only makes our model more general.

The results of this process can be seen on Figure 1, where we plotted two small MTSF with 200 nodes with $p = 0.4$ and $p = 0.8$. Our results are independent of the number of new links. In our simulations this number is set to 3. Also we use $r = 1/t$, where t is equal to the number of types in the network. The length of the walks is set to $m = 5$.

This process, as it is shown in the following, gives similar results in terms of the degree distribution as the Barabasi-Albert graph generation model, but this method does not require global knowledge of the topology.

Lemma 1. *In the whole graph all the created links follows the preferential attachment rule giving the graph power law degree distribution. In the subgraph A (or other type) all the created links follows the preferential attachment rule giving the subgraph power law degree distribution.*

Proof. Our proof follows the line of thoughts in [26]. Our contribution is that we extend it to hold for the typed subgraphs as well. We want to show that if we connect new nodes to the endpoints of the random walks, it yields preferential



(a) $p = 0.4$ (b) $p = 0.8$

Figure 1: The figure shows a MTSF graphs with five types, $N = 200$, plotted with the Fructerman-Reingold spring-based graph drawing algorithm [25]. The type of the nodes are written inside the nodes. In case of $p = 0.8$ the intra-type links, that are running between nodes with the same type, construct clearly separable type groups. This effect is weaker in case of $p = 0.4$. The intra-type links are connecting the type subgraphs together forming a global connected graph. (Color online)

attachment in the form

$$P\left(x\right) = \frac{k_x}{\sum_i k_i}. \tag{1}$$

Let $a_1, a_2 \in A$. The *homophilic* transition probability is given as $P_{A,A}\left(a_2|a_1\right) = 1/k^A_{a_1}$. This equals the probability that the RW will go to node a_2 given that it was in a_1, and a_2 is a neighbor of a_1. Similarly, $P_{A,A}\left(a_1|a_2\right) = 1/k^A_{a_2}$. The *heterophilic* transition probability for $b \in B$ is given as $P_{A,B}\left(b|a_1\right) = 1/k^{A,B}_{a_1}$.

Assume that a new node y is created and it is of type A, and that the initial site of the random walk is chosen uniformly from type A nodes in the network, $P_A\left(s\right) = 1/|A|$. Next, let us calculate the probability that the *homophilic* random walk finds itself at node $d_A \in A$ after one step by employing the Bayes-rule:

$$P_{A,A}\left(d_A\right) = \frac{P_{A,A}\left(d_A|s\right) P_A\left(s\right)}{P_{A,A}\left(s|d_A\right)} = \frac{k^A_{d_A}}{k^A_s}\frac{1}{|A|}.$$

Since node s was chosen randomly, its type A degree k^A_s equals, the average type A degree of the network $\left\langle k^A \right\rangle$, and since $|A|k^A_s = |A|\left\langle k^A \right\rangle = 2|E_A|$,

$$P_{A,A}\left(d_A\right) = \frac{k^A_{d_A}}{2|E_A|}.$$

Similarly, the probability that the *heterophilic* RW is at node $d_B \in B$ is

$$P_{A,B}\left(d_B\right) = \frac{k^{A,B}_{d_B}}{2|E_{A,B}|}.$$

The probabilities $P_{A,A}$ and $P_{A,B}$ invariant of the length of the walk, so we can use them to calculate the probability $P(d)$ in the stationary sense[1] that the RW is at node d as $P(d) = p\, r\, P_{A,A}(d) + p(1-r) P_{B,B}(d) + (p-1)\; r\; P_{A,B}(d) + (p-1)(1-r) P_{B,A}(d)$. If we iterate the graph generating process for long enough, we can approximate $|E_A| \approx p\; r\; |E|$, $|E_B| \approx p(1-r)|E|$, $|E_{A,B}| \approx (1-p)\, r|E|$ and $|E_{B,A}| \approx (1-p)(1-r)|E|$, and we get

$$P(d) = \frac{1}{2|E|}\left(k_d^A + k_d^B + k_d^{A,B} + k_d^{B,A}\right). \tag{2}$$

As a special case, when we are interested in the type A subgraph, $d_A \in A$,

$$P(d_A) = P_{A,A}(d_A) = \frac{k_{d_A}^A}{2|E|}. \tag{3}$$

From these equations it can be seen that the subgraphs of a single type, and also, the whole graph follows the preferential attachment rule.

In Figure 2 the degree distributions of MTSF are shown in case of two types for $N = 10^5$, (a) $p = 0.4$ and (b) $p = 0.8$. The plots show that both the whole graph and the typed subgraphs follow power-law distribution. It can be seen that the feature is invariant of the parameter values p. However, for $p = 0.8$ the typed subgraphs contain more intra-type links than in case of $p = 0.4$. The distribution of inter-type links is not counted in the diagrams, hence the bigger gap between the distribution of typed subgraphs and the whole graph in case of $p = 0.4$. In the following sections we analyze how the ratio of intra-type and inter-type links effect the performance of our type dependent biased capture process in terms of the average time of the captures and the distribution of the load on nodes.

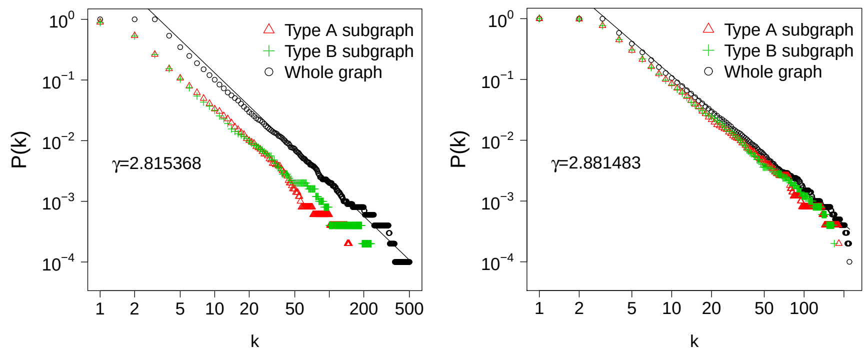
3. Type dependent biased capture process

In this section we show how a small modification of the BNLL model defined in [13] allows us to leverage the homophilic structure of the graphs generated by the multi-type scale-free graph model presented in Sec. 2 to distribute the traffic load.

3.1. Type dependent biased random walks

In the graphs generated by our model, different types are assigned to the vertices. In our type based BNLL process the random walkers (RWrs), i.e. the lamb and the lion, are also assigned one of these types. This enables us to construct the transition probabilities for the RWrs, such that they favor links, which will take them to nodes with the same type as themselves. Our formalism

[1] This can be easily deduced when we substitute the derived results into a longer walk, the terms corresponding to the steps in the middle fall out.



(a) $p = 0.4$

(b) $p = 0.8$

Figure 2: Cumulative degree distribution of MTSF in case of two types, $N = 10^5$. The plots show the degree distribution for both the whole graph and the subgraphs of the different types as well. The black line shows the slope of the degree distribution exponent γ. (Color online)

of the type dependent random walks adopts the mathematical treatment of random walks detailed in [27]. Let us assume that a random walker (RWr) has type $A \in \mathbb{T}$. We define node attractiveness for a node x as

$$a^A(x) = a^A(type(x), k_x) = \begin{cases} k_x^\alpha & \text{if} \quad type(x) = A \\ 1 & \text{if} \quad type(x) \neq A \end{cases} \tag{4}$$

We define the type dependent random walk on the graph giving type dependent weights to the edges employing the node attractiveness. We define $\mu^A_{x,y} = \mu^A_{y,x}$ $(x, y \in \Gamma)$ as a symmetric weight-function on $\Gamma \times \Gamma$,

$$\mu^A_{x,y} = a^A(x)\, a^A(y). \tag{5}$$

It is easy to see, that if there is only one type in the graph, Eq. (5) is equivalent with the symmetric weight defined in [3] (Eq. (5)). The one-step transition probability is then defined for the type$-A$ random walk by

$$\mathbb{P}^A(x, y) = \frac{\mu^A_{x,y}}{\mu^A(x)}. \tag{6}$$

where $\mu^A(x) = \sum_{y\sim x} \mu^A_{x,y}$. This way we can define type dependent transition probabilities for every type of random walkers on the graph.

3.2. Time spent by walkers on types

We know that the time the RWrs spend on x is proportional to the stationary distribution, which is proportional to $\mu(x) = k^\alpha \sum_{y\sim x} a(y)$, so $\tau(x) \sim k^{\alpha+1}\langle a(y)\rangle$. We used the fact that the degrees of the nodes are uncorrelated in the graph, so we can substitute the $a(y)$ values corresponding to the neighbors of x with the average attractiveness value. We calculate $\langle a(y)\rangle$

from the type dependent attractiveness definition and the distribution of inter-type and intra-type links in the MTSF graph: $\langle a(y)\rangle = (1/N)\sum_y a(y) = (1/N)\sum_{k=1}^{N-1} Nk^{-\gamma}\left[pk^{\alpha} + (1-p)\right]$. This gives us for the average attractiveness

$$\langle a(y)\rangle = \sum_{k=1}^{N-1} pk^{\alpha-\gamma} + (1-p)\,. \tag{7}$$

We calculate the ratio between the time that a RWr spends on its own type and on all other types. This intuitively shows that a RWr prefers to stay on the subgraph consisting of the nodes with the same type as itself. For the sake of simplicity we do the calculations for two types, but our formalization can be generalized to more types. Let us recall, that in a MTSF graph with two types, the number of nodes with type A and B are $|A| = rN, |B| = (1-r)N$ respectively. We calculate the time spent on the set of nodes with type A similarly to the calculation of time on individual nodes. First we calculate the $\mu(A)$ measure of the type-A node set as $\mu(A) = \sum_{x\in A}\mu(x) = |A|\sum_{k=1}^{N-1} ck^{-\gamma}k^{\alpha}k\langle a(y)\rangle$. The time spent on other type of nodes is proportional to $\mu(B) = \sum_{z\in B}\mu(z) = |B|\sum_{k=1}^{N-1} ck^{-\gamma}k\langle a(y)\rangle$. The ratio of the time spent on same and other type of nodes by the walk is then given by

$$\frac{\tau(A)}{\tau(B)} \sim \frac{\mu(A)}{\mu(B)} \sim \frac{\sum_{k=1}^{N-1} k^{\alpha-\gamma+1}}{\sum_{k=1}^{N-1} k^{1-\gamma}}. \tag{8}$$

If one chooses $\gamma > 2$, which is typical for social networks and $\alpha > \gamma - 2$

$$\frac{\tau(A)}{\tau(B)} \sim \frac{N^{\alpha+2-\gamma}}{c - c'N^{2-\gamma}} \sim N^{\alpha+2-\gamma}. \tag{9}$$

The picture is still satisfactory if $\alpha = \gamma - 2$, because then $\frac{\tau(A)}{\tau(B)} \sim c\log N$. If $\alpha < \gamma - 2$ then $\frac{\tau(A)}{\tau(B)} \sim c$, which means we cannot guarantee that the walker will spend most of its time among its type. For $\gamma \leq 2$, the ratio $\frac{\tau(A)}{\tau(B)}$ goes to infinity if $\alpha \geq 0$. In this case the walker stays mostly on its type as well.

3.3. Average capture time of type dependent biased capture process

We run simulations to determine the parameter dependence of our model with respect to parameter p on the average capture time $\langle T\rangle$. We run the simulation of 10^5 type dependent capture process on graphs of size $N = 10^4$, and averaged over 10 graph realizations, setting the parameter p of the MTSF graph generation model. We simulated one iteration with one lion and one lamb, and set the type of the RWrs randomly among the iterations. The results can be seen on Figure 3 for $\alpha = 1, 2, 3$ and 4. The plot shows that on graphs generated with high values of p the average capture times $\langle T\rangle$ degrades. This can be explained by the low number of inter-type links in case of high p, which would allow the RWrs to quickly find their way into their own typed subgraph.

This phenomena is independent of α. However, for high values of $\alpha \geq 4$ the performance of $\langle T\rangle$ decays for low values of p as well. This anomaly can be explained as the effect of the typed subgraphs not being connected in case of low values of p. In this case, setting α to a high value prevents the RWrs to escape a subgraph island where they might get stuck for a while.

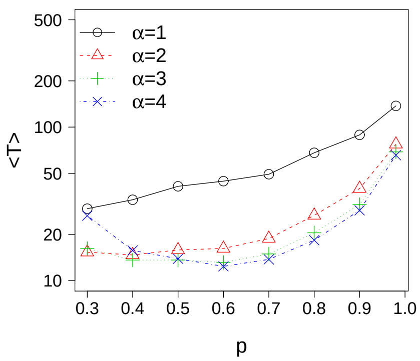


Figure 3: The average capture time for MTSF graphs with 10 types and different p values. (Color online)

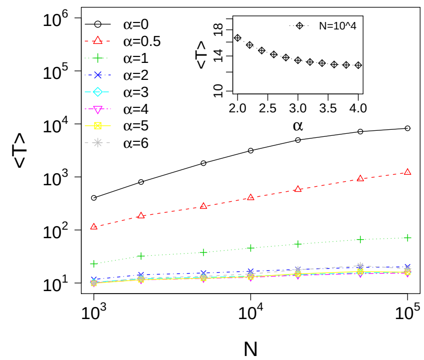
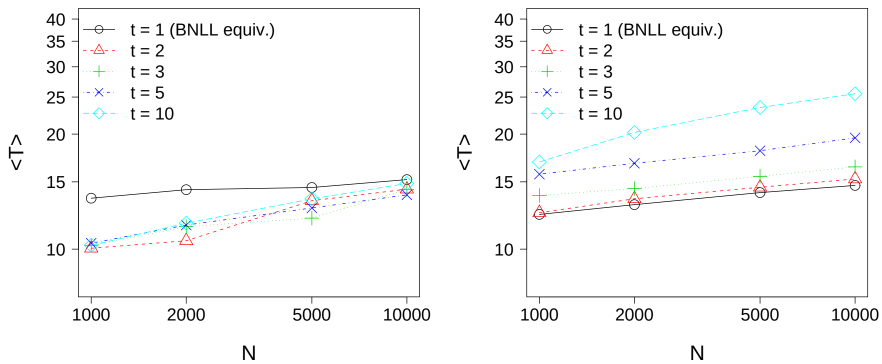


Figure 4: The scaling of the average capture time $\langle T\rangle$ with network growth for different α values in case of 10 types, $p = 0.6$. The inset shows $\langle T\rangle$ for $N = 10^4$ for a finer resolution of α. (Color online)

The scaling behavior of the average capture time $\langle T\rangle$ is also measured in case of different network sizes. We generated graphs with $p = 0.6$ in the size range $N = 10^3..10^5$. The capture process was run on each graph five times the size of the graph, and results are averaged over 10 different graph realizations. Figure 4 shows that the scaling of $\langle T\rangle$ of our type based BNLL model is in accordance with the results in [13]. It is shown on Figure 4 that $\langle T\rangle \approx 10..15$ for $\alpha \geq 2$ when there are 10 types in the MTSF graph. The inset shows that for $\alpha > 4$, $\langle T\rangle$ converges to a optimal value.

Figure 5 shows how the number of types in the MTSF graph effects the average capture time. Our calculations suggest that by introducing more than one type into the graph, the average capture time decreases, because the type based bias forces the walkers to stay on the subgraph of their own type, which is a smaller graph than the whole graph. The single-typed graph is a special case of the multi-type graph model, which represents the original BNLL model with one lion and one lamb. In general, the $\langle T\rangle$ values on graphs with many types are similar to the results on single-typed graphs. On Figure 5 (a) the graphs were generated with $p = 0.4$, which yields more inter-type links. This makes it easier for the RWrs to find their way into the subgraph corresponding to their own type which results in lower $\langle T\rangle$ values. Figure 5 (b) shows the average capture times on graphs with $p = 0.8$. On this figure it can be seen that the high number of intra-type links slightly degrades the average capture times. With more number of types the effect is stronger, as it gets harder for the RWr to find its way into its own type. The simulations were run with $\alpha = 2$.



(a) $p = 0.4$ (b) $p = 0.8$

Figure 5: Figures show the effect of parameter p on the scaling of $\langle T \rangle$ in case of 1, 2, 3, 5 and 10 types. (Color online)

4. Load balancing properties of capture process

The original BNLL search strategy proved to be successful except with respect to the load of high degree nodes. In this section we demonstrate that the use of multi-type graphs and the type dependent search algorithm provides load balancing between high degree nodes with different types. We define the load $\tau(x)$ of a node x as the time that the RWrs spend on x.

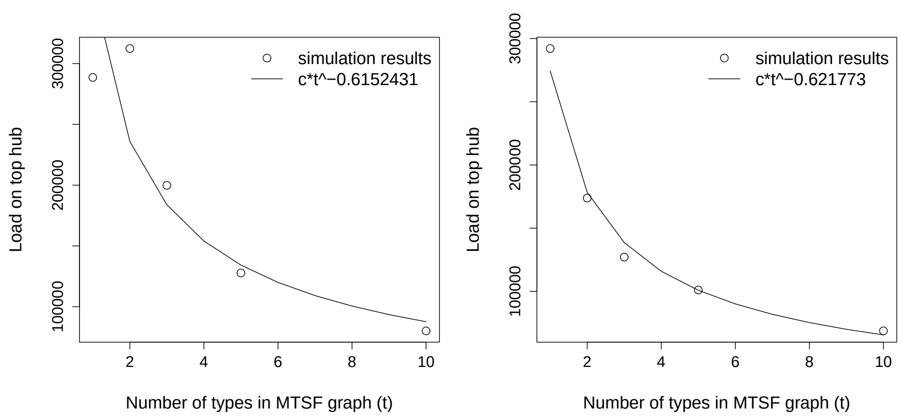
4.1. Load on hubs

It is well-known that the degree of the top degree node on a scale free network (with fat tail exponent γ and with N nodes) scales with $N^{\frac{1}{\gamma-1}}$ (c.f. in [28] below (135)). That determines the traffic load on the top node as well.

A random walker is either of the same type as the top degree node with probability $\frac{1}{t}$ or other with probability $1 - \frac{1}{t}$. In the former case the load is $N^{\frac{\alpha+1}{\gamma-1}} \langle a(y) \rangle$ while in the latter $N^{\frac{1}{\gamma-1}} \langle a(y) \rangle$. The traffic load is than $\tau(x_{top}) \sim \frac{1}{t} N^{\frac{\alpha+1}{\gamma-1}} \langle a(y) \rangle + \left(1 - \frac{1}{t}\right) N^{\frac{1}{\gamma-1}} \langle a(y) \rangle$:

$$\tau(x_{top}) \sim \frac{1}{t} N^{\frac{\alpha+1}{\gamma-1}} \langle a(y) \rangle \qquad (10)$$

We checked the analytical expectations of the load balancing feature of the proposed process on the MTSF graphs. Simulations of 10^5 type dependent capture processes were run on MTSF graphs of size $N = 10^4$, generated with different values of p, and averaged over 10 graph realizations. During these simulations we set $\alpha = 3$. Figure 6 shows the simulation results for the load on the top hubs in our generated MTSF graphs. The simulation results seemingly contradict the analytic results, because $\tau(x_{top}) \sim c\ t^{-\eta}$ with $\eta \approx -0.6$ fits the measured data better than $a + b\ t^{-1}$. However, this is an artifact of the least square method due to the heteroscedasticity and finite size effects. Also, on Figure 6 (a) we see that the simulation data deviates from our analytical expectation in case of $t = 2$. This abnormality looks coming from the difference of the network structure for $p = 0.4$ and 0.8, where the network for $p = 0.4$ has more inter-type links than the network in case of $p = 0.8$. In the network with low number of inter-type links (large p) and $t > 1$, RWrs are mainly walking around their own type hubs and they are locked into the cluster of own type nodes for long time. As a result, the load is distributed on high degree hubs of different types, as it is shown on Figure 6 (b). However, with more inter-type links and $t > 1$, RWrs can walk in and out of different type of clusters more easily. In particular, if there are two types of nodes ($t = 2$), the top hub has one of two types and the higher ratio of inter-type links helps to visit it. As a result, the load is not decreased as much as it is expected, and as it does in case of smaller number of links between types. This anomaly can be seen in case of $t = 2$ on Figure 6 (a).



(a) $p = 0.4$ (b) $p = 0.8$

Figure 6: Figures show the simulation results for the load on the top hub (circles) and the analytical trend line fitted to the data (line) in case (a) $p = 0.4$ and (b) $p = 0.8$

High traffic is effected not only by the top hub but other high degree nodes. A possible choice of the identification of this part of the graph is to consider the Hirsch-core introduced in [29]. A network has a h-index of h, if not more than h of its nodes have a degree of not less than h. The Hirsch-core $H \subset V$ then is the set of the top h nodes, all has degree not less then h. The cumulated traffic on the core can be calculated easily as $\tau(H) \sim \mu(H) = \sum_{x \in H} \mu(x)$:

$$\tau(H) \sim \frac{1}{t} N^{\frac{\alpha+2}{\gamma-1}} \langle a(y) \rangle \tag{11}$$

Figure 7 (a) and (b) shows the comparison of the load on the Hirsch-core in case of 1, 3 and 10 types embedded into the MTSF graph. On the x-axis we plotted the degrees k of the member nodes of the Hirsch-core in ascending order. The y-axis shows the load, which correspond to the node with degree k. We plotted the the average of simulation results of 10 graph realizations on each plot. On Figure 7 (a) and (b) it can be seen that the load on the top degree node is about 10 times lower in case of $t = 10$ than in case of $t = 1$, which coincides with our analytical expectations. The figures clearly show that when there are 10 types in the MTSF graph, the very high load of the top hubs is shared out among the lower degree hubs in the core. As it can be seen from the figures, the load balancing feature is virtually independent of the value p, i.e. the ratio of the inter-type and intra-type links in the MTSF graph.

The distribution of the number of captures is important aspect of a search system built on the diffusive capture process. The node, where a capture event takes place, must inform the party with the requested resource to send it to the user who is looking for that resource. On Figure 7 (c) and (d) it is shown that in case of many types, the number of capture events is distributed similarly well to the load among the top hubs.

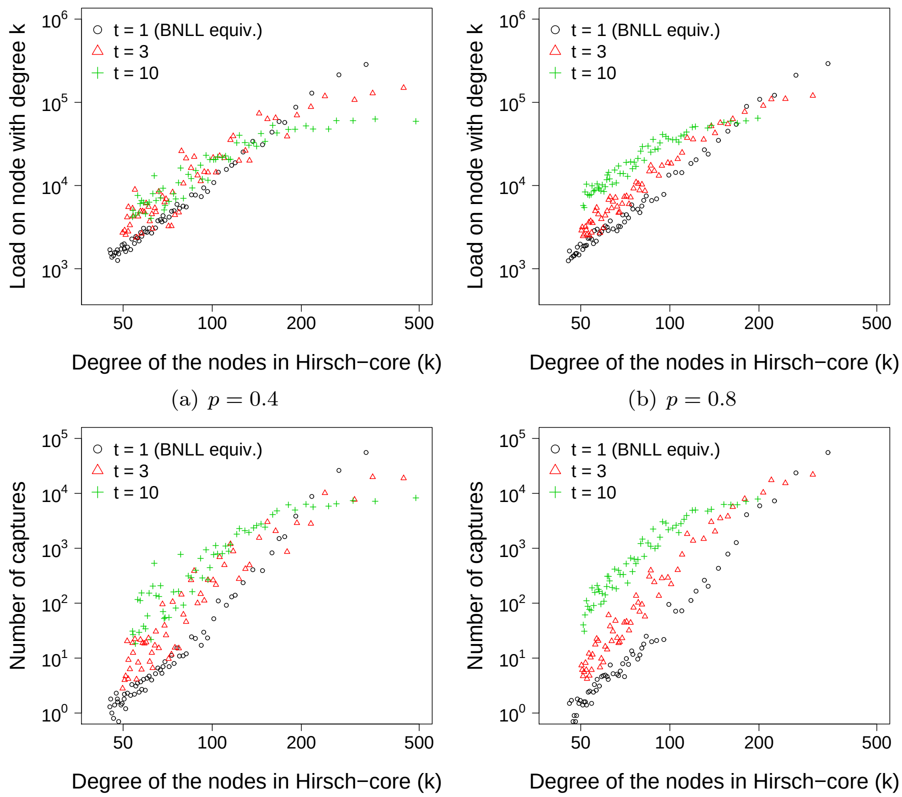
5. Conclusions

In our paper an improved version of the biased diffusive capture process was presented, which distributes the traffic load of the high degree nodes in the network. A multi-type scale-free graph generation model was proposed to embed homophily structure into the overlay graph with the use of type dependent random walks. It was shown that both the whole graph and also the typed subgraphs have power law degree distribution, which speeds up the capture process. The biased capture process has been improved to employ type dependent transition probabilities, which efficiently spreads the load among the top hubs with different types. The paper demonstrated through analysis and simulations that the modified capture process achieves similar or better average capture times, as the original BNLL model in case of one lion and one lamb. Also, it was shown that the traffic load incurred by the capture process on the high degree nodes are spread evenly among the typed subgraphs. The proposed system lends itself to applications in which the available resources can be effectively categorized into a small number of categories. As a future improvement the types can be embedded into a metric space, which could allow more sophisticated means of categorization. In this current paper we showed that with such a simple model detailed here, the advantageous features of balancing the load on the high degree nodes are already available.

Acknowledgements

The authors would like to thank the anonymous referees for their insightful and helpful comments which greatly helped to improve the paper.

[1] S. Redner. *A guide to first-passage processes*. Cambridge Univ Pr, 2001.



(a) $p = 0.4$ (b) $p = 0.8$

(c) $p = 0.4$ (d) $p = 0.8$

Figure 7: Figures (a-b) show the traffic load distribution on the nodes of the Hirsch-core in case of 1 and 10 types for different values of p. Figures (c-d) show the number of captures on the Hirsch-core for the same simulations as above. The x-axis shows the degree of the core nodes in ascending order. (Color online)

[2] P.L. Krapivsky and S. Redner. Kinetics of a diffusive capture process: lamb besieged by a pride of lions. *Journal of Physics A: Mathematical and General*, 29:5347–5357, 1996.

[3] S. Kwon, S. Yoon, and Y. Kim. Condensation phenomena of a conserved-mass aggregation model on weighted complex networks. *Phys. Rev. E*, 77:066105, Jun 2008.

[4] D. Ben-Avraham. Computer simulation methods for diffusion-controlled reactions. *The Journal of Chemical Physics*, 88(2):941–948, 1988.

[5] Shlomo Havlin, Raoul Kopelman, Rodney Schoonover, and George H. Weiss. Diffusive motion in a fractal medium in the presence of a trap. *Phys. Rev. A*, 43:5228–5232, May 1991.

[6] C. Wang and B. Li. Peer-to-peer overlay networks: A survey. *Department of Computer Science, The Hong Kong University of Science and Technology*, 2003.

[7] J. Risson and T. Moors. Survey of research towards robust peer-to-peer networks: Search methods. *Computer Networks*, 50(17):3485–3521, 2006.

[8] S. Lee, S.-H. Yook, and Y. Kim. Diffusive capture processes for information search. *Physica A: Statistical Mechanics and its Applications*, 385:743–749, 2007.

[9] J.D. Noh and H. Rieger. Random walks on complex networks. *Phys. Rev. Lett.*, 92(11):118701, 2004.

[10] S. Condamin, O. Bénichou, V. Tejedor, R. Voituriez, and J. Klafter. First-passage times in complex scale-invariant media. *Nature*, 450(7166):77–80, 2007.

[11] V. Tejedor, O. Bénichou, and R. Voituriez. Global mean first-passage times of random walks on complex networks. *Physical Review E*, 80(6):065104, 2009.

[12] S. Lee, S.-H. Yook, and Y. Kim. Diffusive capture process on complex networks. *Phys. Rev. E*, 74:046118, Oct 2006.

[13] S. Lee, S.-H. Yook, and Y. Kim. Searching method through biased random walks on complex networks. *Phys. Rev. E*, 80:017102, Jul 2009.

[14] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3–5):75–174, 2010.

[15] Karsten Steinhaeuser and Nitesh V. Chawla. Identifying and evaluating community structure in complex networks. *Pattern Recognition Letters*, 31(5):413–421, 2010.

[16] Aykut Firat, Sangit Chatterjee, and Mustafa Yilmaz. Genetic clustering of social networks using random walks. *Computational Statistics and Data Analysis*, 51(12):6285–6294, 2007.

[17] Josep M. Pujol, Andreas Flache, Jordi Delgado, and Ramon Sangüesa. How can social networks ever become complex? modelling the emergence of complex networks from local social exchanges. *Journal of Artificial Societies and Social Simulation*, 8(4):12, 2005.

[18] M. Tsvetovat and K.M. Carley. Generation of realistic social network datasets for testing of analysis and simulation tools. Technical report, DTIC Document, 2005.

[19] Steve Gregory. Ordered community structure in networks. *Physica A: Statistical Mechanics and its Applications*, 391(8):2752–2763, 2012.

[20] A. Lancichinetti, S. Fortunato, and F. Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78(4):046110, 2008.

[21] Lynne Hamill and Nigel Gilbert. Social circles: A simple structure for agent-based social network models. *Journal of Artificial Societies and Social Simulation*, 12(2):3, 2009.

[22] A. Vázquez. Growing network with local rules: Preferential attachment, clustering hierarchy, and degree correlations. *PHYSICAL REVIEW E Phys Rev E*, 67:056104, 2003.

[23] P.L. Krapivsky and S. Redner. Organization of growing random networks. *Physical Review E*, 63(6):066123, 2001.

[24] P. Pollner, G. Palla, and T. Vicsek. Preferential attachment of communities: The same principle, but a higher level. *EPL (Europhysics Letters)*, 73:478, 2006.

[25] T.M.J. Fruchterman and E.M. Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991.

[26] J. Saramaki and K. Kaski. Scale-free networks generated by random walkers. *Physica A: Statistical Mechanics and its Applications*, 341:80–86, 2004.

[27] A. Telcs. *Art of Random Walks, Lecture notes in mathematics (Book 1885).* Springer, 2006.

[28] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74:47–97, Jan 2002.

[29] A. Schubert, A. Korn, and A. Telcs. Hirsch-type indices for characterizing networks. *Scientometrics*, 78(2):375–382, 2009.