

Random walks, interacting particles, dynamic networks: Randomness can be helpful

Colin Cooper *

1 Introduction

The aim of this article is to discuss some applications of random processes in searching and reaching consensus on finite graphs. The topics covered are: Why random walks?, Speeding up random walks, Random and deterministic walks, Interacting particles and voting, Searching changing graphs.

As an introductory example consider the self-stabilizing mutual exclusion algorithm of Israeli and Jalfon [35], based on the random walk of tokens on a graph G . Initially each vertex emits a token which makes a random walk on G . On meeting at a vertex, tokens coalesce. Provided the graph is connected, and not bipartite, eventually only one token will remain, and the vertex with the token has exclusive access to some resource. The token makes a random walk on G , so in the long run it will visit all vertices of G . Typical questions are: how long before only one token remains (coalescence time), how long before every vertex has been visited by the token (cover time), what proportion of the time does each vertex have the token in the long run (stationary distribution of the random walk), how long before the walk approaches the stationary distribution (mixing time), how long before the number of visits to each vertex approximates the frequency given by the stationary distribution (blanket time). If these quantities can be understood and manipulated, then we can tune the random walk to perform efficiently on a given network.

For example, in *Fair Circulation of a Token* [33], the transition probability of the walk is modified, so that each vertex has the same probability of holding the token in the long run, i.e. the stationary distribution is uniform. The questions asked in [33] are, how can this modification be achieved, and what

*Department of Informatics, King's College, London, U.K.

is its effect on the quantities above. The walk proposed in [34] is reversible, so the relationship between the the blanket and cover times holds (see (1), below). Thus after a time of the order of the coalescence time plus cover time, all vertices get an acceptable share of the token. The aim is to choose transition probabilities for the walk which minimize coalescence and cover time subject to the fairness condition.

2 Terminology

There is a substantial literature dealing with discrete random walks. For an overview see e.g. [2, 40]. We review some definitions and results.

Given a graph $G = (V, E)$, let $|V| = n$, $|E| = m$, and let $d(v) = d_G(v)$ denote the degree of vertex v for all $v \in V$. A *simple* random walk $\mathcal{W}_v = (\mathcal{W}_v(t), t = 0, 1, \dots)$ is defined as follows: $\mathcal{W}_v(0) = v$ and given $x = \mathcal{W}_v(t)$, $\mathcal{W}_v(t + 1)$ is a randomly chosen neighbour of x . A simple random walk on a graph G defines a Markov chain on the vertices V . If G is a finite, connected and non-bipartite graph, then this chain has a stationary distribution π given by $\pi(v) = d_G(v)/(2|E|)$. Thus if $P_v^{(t)}(w) = \mathbf{Pr}(\mathcal{W}_v(t) = w)$, then $\lim_{t \rightarrow \infty} P_v^{(t)}(w) = \pi(w)$, independent of the starting vertex v .

A weighted random walk, assumes each *undirected* edge $e = \{u, v\}$ has a weight (or conductance) $w_e = w_{v,u} = w_{u,v}$. Thus $w_e = 1/r_e$, where r_e is resistance of the edge. The vertex weight $w_v = \sum_{u \in N(v)} w_{v,u}$, and the transition probability of the random walk at v is $p(v, u) = w_{v,u}/w_v = w_e/w_v$. The total weight of the network is $w = \sum_{edges\ e} w_e$, each edge being counted at each vertex. The stationary distribution of the walk, for vertices is $\pi(v) = w_v/w$, and for directed transitions $e = (u, v)$, $\pi(e) = w_e/w$. For simple walks, $w_{u,v} = r_{u,v} = 1$, $w_v = d(v)$ the degree of vertex v , and $p(u, v) = 1/d(v)$; the total weight $w = 2m$, and so $\pi(v) = d(v)/2m$ and $\pi(e) = 1/2m$.

A Markov chain is reversible if $\pi(u)p(u, v) = \pi(v)p(v, u)$. For a weighted walk and edge $e = \{u, v\}$, as $\pi(u)p(u, v) = w_e/w = \pi(v)p(v, u)$, weighted walks are always reversible. Its fair to say that reversible random walks are usually easier to analyze than non-reversible walks. Examples of non-reversible walks are walks on digraphs and Google page rank computation.

Cover Time. For $v \in V$ let C_v be the expected time taken for a random walk W on G starting at v , to visit every vertex of G . The *vertex cover time* $C(G)$ of G is defined as $C(G) = \max_{v \in V} C_v$.

Cover Time of a simple random walk. The vertex cover time of simple random walks on connected graphs has been extensively studied. It

is a classic result of Aleliunas, Karp, Lipton, Lovász and Rackoff [4] that $C(G) \leq 2m(n-1)$. It was shown by Feige [27], [28], that for any connected graph G , the cover time satisfies $(1 - o(1))n \log n \leq C(G) \leq (1 + o(1))\frac{4}{27}n^3$. As an example of a graph achieving the lower bound, the complete graph K_n has cover time determined by the Coupon Collector problem. The *lollipop* graph consisting of a path of length $n/3$ joined to a clique of size $2n/3$ gives the asymptotic upper bound for the cover time.

Blanket Time. Let $N_v(t)$ be the number of times a random walk visits vertex v in t steps. Then for a walk in stationarity, $\mathbf{E}N_v(t) = t\pi_v$. The blanket time $\tau_B(\delta)$ is the first $t \geq 1$ such that for all vertices $u, v \in V$

$$\frac{N_u(t)/\pi(u)}{N_v(t)/\pi(v)} \geq \delta.$$

Let $t_B(G, \delta) = \max_{v \in V} \mathbf{E}_v \tau_B(\delta)$, then Ding, Lee and Peres [21] prove that for any reversible random walk and $0 < \delta < 1$

$$t_B(G, \delta) = A(\delta)C(G) \tag{1}$$

for some constant $A(\delta) \geq 1$. In other words the blanket time $t_B(G, \delta)$ is a constant multiple of $C(G)$.

Mixing Time. For $\epsilon > 0$ let

$$T_G(\epsilon) = \max_v \min \{t : \|P_v^{(t)} - \pi\|_{TV} \leq \epsilon\},$$

where

$$\|P_v^{(t)} - \pi\|_{TV} = \frac{1}{2} \sum_w |P_v^{(t)}(w) - \pi(w)|$$

is the Total Variation distance between $P_v^{(t)}$ and π .

We say that a random walk on G is *rapidly mixing* if $T_G(1/4)$ is *poly*($\log |V|$). The choice of $1/4$ is somewhat arbitrary, any constant strictly less than $1/2$ will suffice. Rapidly mixing Markov chains are extremely useful, as the process soon converges to stationary behavior. Many random graphs are expanders, and walks on expanders are rapidly mixing. This expansion property also leads to logarithmic diameter, and good connectivity, explaining the attractiveness of random graphs as network models.

3 Speeding up Random Walks

As previously mentioned, for a simple random walk, the cover time $C(G)$ of a connected n vertex graph G satisfies the bounds $n \log n \leq C(G) \leq (4/27)n^3$. There are several strategies that can be used to speed-up the cover time of a random walk on an undirected connected graph. In this section, the approach we consider, is to modify the behavior of the walk. The price of the speed up is some extra work performed locally by the walk, or by the vertices of the graph. Another approach, considered in Section , is to use multiple random walks.

Biased transitions. Use weighted transition probabilities derived from a knowledge of the local structure of the graph. The general theory of reversible weighted random walks is given in [2]. Ikeda, Kubo, Okumoto, and Yamashita [34] studied the speed up in the worst case cover time of any connected graph obtained by using transition probabilities which are a function of the degree of neighbour vertices (β -walks). For example, they found that if edge $e = \{u, v\}$ is given a weight $w_{u,v} = 1/\sqrt{d(u)d(v)}$, then this gives a $C(G) = O(n^2 \log n)$ upper bound on cover time for *any* connected n vertex graph G , (as opposed to $\Theta(n^3)$ bound for simple random walks).

The following proof that the cover time of any reversible random walk is $\Omega(n \log n)$, is due to T. Radzik. The expected first return time $\mathbf{E}T_u^+$ to u is $\mathbf{E}T_u^+ = 1/\pi(u)$. Also, $\mathbf{E}T_u^+$ is at most the commute time $K(u, v)$ between u and v ($K(u, v) \geq \mathbf{E}T_u^+$). Why? We either visit v on way back to u or we do not. For at least half the vertices $\pi(u) \leq 2/n$. Why? $\sum_{u \in V} \pi(u) = 1$. Let S be this set of vertices, all with $K(u, v) \geq \mathbf{E}T_u^+ \geq n/2$. Let $K_S = \min_{i,j \in S} K(i, j)$ then [36]

$$C(G) \geq (\max_{S \subseteq V} K_S \log |S|)/2.$$

Thus $C(G) \geq (n/4) \log(n/2)$.

Local exploration. At each step the walk uses look-ahead probing to a fixed distance, or marks an unmarked neighbour. A look-ahead- k walk can see all vertices at edge distance less than or equal to k from its current position. A simple random walk is look-ahead-0. Look-ahead walks were studied by [29]. For graphs of large minimum degree, using look-ahead can substantially improve cover time. For example, for finite k , look-ahead- k random walks on the hypercube reduce the cover time to $O(n/(\log n)^{k-1})$.

However for regular graphs of constant degree, it is shown in [17] that look-ahead- k does not reduce cover time below $\Theta(n \log n)$.

In [1], Adler, Halperin, Karp, and Vazirani introduce a sampling process based on coupon collecting on a network. At each step the process chooses a random vertex v . If v is unseen then it is marked as seen. If v is seen but has unseen neighbours, pick an unseen neighbour u.a.r. and mark it. The authors show that, e.g. for the n vertex hypercube H_n , the time to mark all vertices is $O(n)$. The random walk version of this marking process of [1] was studied in [12]. Depending on the degree and the expansion of the graph, the authors prove several upper bounds similar to [1]. In particular, when G is the hypercube or a random graph of minimum degree $\Omega(\log n)$, the process marks all vertices in time $O(n)$, improving the $\Theta(n \log n)$ cover time of standard random walks.

Previous history. Modify the walk transitions using previous history of the walk to avoid repetitions.

Non-backtracking walks are fast $O(n)$ on a n -cycle, but seem to lead to little improvement on expanders. Alon et al. [6] considered non-backtracking random walks on r -regular expanders. A non-backtracking walk $X(t+1)$ does not return over the edge $(X(t-1), X(t))$ unless no other move is possible. They establish that these walks are rapidly mixing. However, this mixing result can be used to show that this process has a cover time of $\Omega(n \log n)$ for r -regular expanders.

Avin and Krishnamachari [9] considered *Random Walks with Choice*. Instead of moving to a random neighbour at each step, the walk selects d neighbours uniformly at random and then chooses to move to the least visited vertex among them.

Edge processes. In [13] the authors investigate a random process which prefers unvisited edges. An *edge-process* (E -process) acts as follows: If there are *unexplored edges* incident with the current vertex pick one according to a rule \mathcal{A} and make a transition along this edge. If there are no unexplored edges incident with the current vertex, move to a random neighbour using a simple random walk. Thus the walk uses unvisited edges whenever possible, and makes a random walk otherwise. The rule \mathcal{A} could be determined online by an adversary; alternatively it could be a random choice over unvisited edges incident with the current walk position.

We use the expression *with high probability* (**whp**) to mean with probability tending to one asymptotically, as the size n of the vertex set tends to

infinity.

For random regular graphs of even degree, **whp** the $E(\mathcal{A})$ -process explores the graph in expected time linear in the size of the vertex set, *irrespective of the choices made by rule \mathcal{A}* .

Theorem 1 [13] *Let $\mathcal{G}_r(n)$ be the set of n vertex r -regular graphs of constant even degree, $r \geq 4$. Let G be chosen uniformly at random from \mathcal{G}_r . Let \mathcal{A} be an arbitrary rule for choosing unvisited edges, and let $C_E(G)$ denote the cover time of the $E(\mathcal{A})$ -process on G . Then **whp** $C_E(G) = O(n)$, irrespective of the choice of rule \mathcal{A} .*

The **whp** term in the theorem above depends on the u.a.r. choice of graph G , not on the expected performance of the E -process, and is asymptotic in the size n of the vertex set. The cover time of a random regular graph of degree $r \geq 3$ is $(r - 1)/(r - 2) n \log n$ **whp** [17]. Thus Theorem 1 offers an $\Theta(\log n)$ speed up.

However, as usual, there is a caveat. For odd degree regular graphs, the experiments below show that performance is same order as simple random walk. Thus, as in the case of the β -walk of [34], improvement in performance depends on exploiting the topology of the graph.

The experimental plot (Figure 1) (from [13]) gives the *normalised* cover time of the E -process, i.e. the actual cover time divided by n . The curves drawn behind the experimental data in the figure are of the form $cn \log n$, where c was determined by inspection. In the experiments, unvisited edges are chosen u.a.r. by the E -process. It would appear the plots for even degrees 4 and 6 are constant, i.e. the cover time is $O(n)$. The experimental evidence suggests e.g. that the normalised cover time of 3-regular graphs is $0.93n \log n$.

Using Randomness in Deterministic Walks

In the context of robotics the exploration of graphs is often done using the rotor-router model, or Propp machine. The process works as follows. Each undirected edge $\{u, v\}$ is replaced by a pair of directed edges $(u, v), (v, u)$. All edges adjacent to a vertex are assigned labels in some order. In the beginning a pointer points to the first edge in this order which is used to determine the next edge to be traversed if v is visited. The walk starts at an arbitrary vertex. It moves over the edge to which the pointer points. After this edge is traversed, the pointer moves on to the edge with the next label, in a cyclic way. Since the number of configurations (direction of the rotors and the position of the walk) is bounded, the walk must be *locked-in* a loop

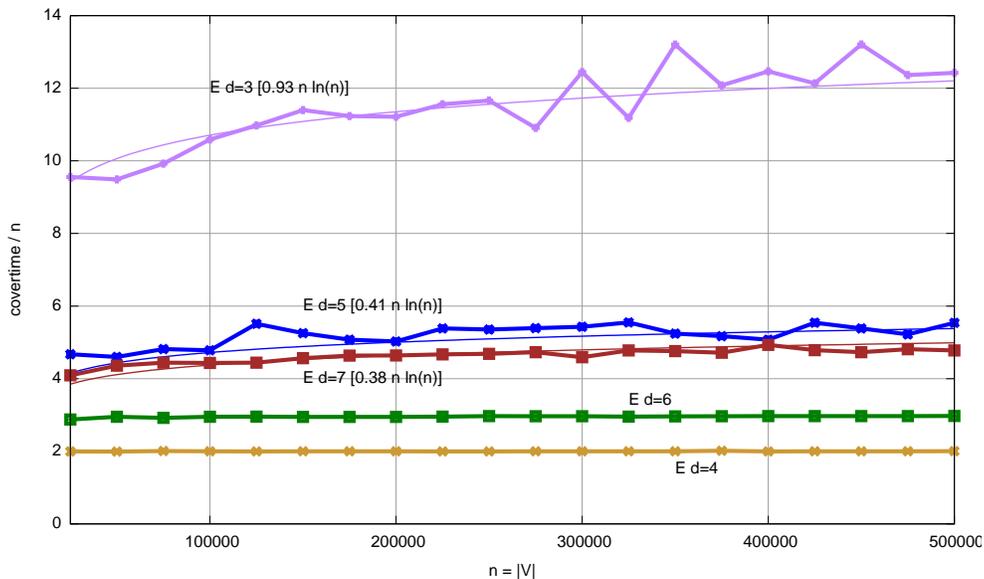


Figure 1: Normalised cover time of E -process as function of size and degree d

eventually. The authors of [43] proved that the walk gets locked-in an Euler tour. In [11, 44] the authors proved that the lock-in time is bounded by $O(|V| \cdot |E|)$. This bound was further improved in [45] to $2|E| \cdot D$, where D is the diameter of G .

In [30] the authors consider a deterministic walk, the *basic walk*. The idea of the walk is based on an observation that one can cover the symmetric digraph counterpart of a graph by a collection of directed cycles. The cycles are formed according to a simple rule. At any node v with the degree d , the incoming edge incident via a port i becomes the predecessor of the outgoing arc incident via port $(i + 1) \bmod d$. A certain arrangement of the edge of the nodes of the graph leads to a cycle that visits every node. In [20] the authors show that there exists an edge order that leads to a cycle of length $4.33n$ that visits every node.

In [13], a variant of this basic walk was used to overcome the problems encountered by the edge-process (E -process) on odd-degree regular graphs. The \vec{E} -process was defined as follows. Firstly, replace the graph G with a symmetric r -regular digraph, $D(G)$. Thus each edge $\{u, v\}$ of G is replaced by a pair of directed edges (u, v) , (v, u) . Edges are initially distinguished

as unvisited in each direction. As with the E -process, the \vec{E} -process starts from some arbitrary vertex v , and makes a transition along an unvisited out-edge chosen according to some rule \mathcal{A} . On returning to v , another unvisited out-edge is chosen, until all out-edges incident with v have been inspected. The walk then moves at random until a vertex u with unvisited out-edges is encountered, and u becomes the new start vertex. Transitions along unvisited edges to vertices x other than the current start vertex v , are handled as follows. Let a_0, \dots, a_{r-1} be the neighbours of x in some fixed order. On entering a vertex x along unvisited directed edge (a_i, x) , we exit along (x, a_{i+1}) (addition modulo r). The purpose of this is to stop the walk turning back on itself at any vertex, and is the same idea as the basic walk of [30], except it is applied only to unvisited edges.

Theorem 2 *Let $\mathcal{G}_r(n)$ be the set of n vertex r -regular graphs of constant degree $r \geq 3$. Let G be chosen u.a.r. from \mathcal{G}_r . Let $\vec{C}_E(G)$ denote the cover time of the $\vec{E}(\mathcal{A})$ -process on G . Then **whp** $\vec{C}_E(G) = O(n)$.*

This $O(n)$ cover time is to be compared to an $O(|E|D) = O(n \log n)$ upper bound for Propp machines on random regular graphs.

The following experiments for the \vec{E} -process are from [13]. The results for the undirected E -process for degree-3, and degree-5 graphs are taken from Figure 1, and compared with the \vec{E} -process in Figure 2. The directed processes appear linear, whereas the undirected processes appear to have $\omega(n)$ growth. The experiments of [13] indicated that although the time spent on the random walk in the \vec{E} -process is a small constant proportion of the total (less than 1/20%), it was still necessary for $O(n)$ cover time. The experiments thus raise many interesting questions; and in particular the role of randomness as a catalyst to assist rapid completion of otherwise deterministic search processes.

4 Multiple and Interacting Particles

We distinguish three main types of process, namely particles which walk independently but coalesce on meeting, and those which walk independently but remain distinct, and either interact (e.g. exchange information) on meeting, or have no interaction.

4.1 Coalescing particle systems

In a coalescing random walk, a set of particles make independent random walks in an undirected connected graph. Whenever one or more particles

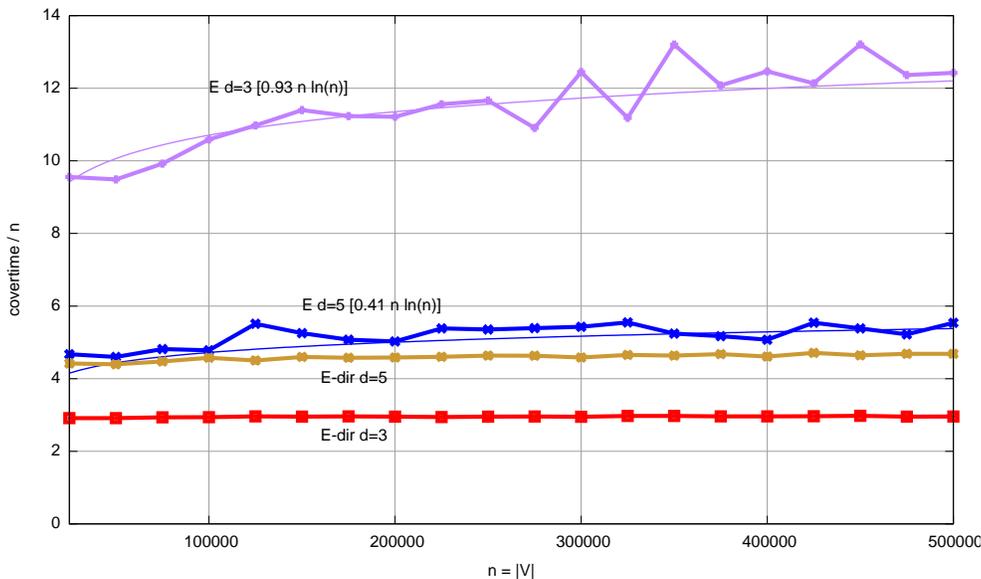


Figure 2: Directed \vec{E} -process *vs* E -process for $d = 3$ and $d = 5$

meet at a vertex, then they become a single particle which continues with the random walk.

The expected time for all initial particles to coalesce to a single particle depends on the starting positions of the walks. For a connected graph G , let $C_k(i_1, \dots, i_k)$, $2 \leq k \leq n$, be the coalescence time when there are initially k particles starting from distinct vertices i_1, \dots, i_k . The worst case expected coalescence time is $C(k) = \max_{i_1, \dots, i_k} \mathbf{E}(C_k(i_1, \dots, i_k))$.

A system of coalescing particles where initially one particle is located at each vertex, corresponds to the *voter model*, which is defined as follows. Initially each vertex has a distinct opinion, and at each step each vertex changes its opinion to that of a random neighbour. Let $\mathbf{E}(C_V)$ be the expected time for voting to be completed, i.e. for a unique opinion to emerge. ($\mathbf{E}(C_V)$ is also called the *voting time*, *trapping time* or the *consensus time*.) It is known that the expected time for a unique opinion to emerge, is the same as the expected time $C(n)$ for all the particles to coalesce (see [2]). Thus, by establishing the expected coalescence time $C(n)$, as $\mathbf{E}(C_V) = C(n)$, we also obtain the expected time for voting to be completed. If the graph G is bipartite, then for coalescence to complete, it is assumed that the walk pauses with some fixed probability at each step. Equivalently, for voting, that vertices

may choose their own opinion with this probability.

We summarize some of what is known about these problems for finite graphs. Cox [19] considered coalescence time of random walks and the consensus time of the voter model for d -dimensional tori. In a variant of the voter model, *the two-party model*, initially there are only two opinions A and B . The two-party model was considered by Donnelly and Welsh [24]. Hassin and Peleg [31] and Nakata *et al.* [41] also consider the two-party model, and discuss its application to agreement problems in distributed systems. These papers focus on analysing the probability that all vertices will eventually adopt the opinion which is initially held by a given group of vertices. The central result is that the probability that opinion A wins is $d(A)/(2m)$, where $d(A)$ is the sum of the degrees of vertices initially holding opinion A , and m is the number of edges in G . The case where there are more than two opinions, can be reduced to the two opinion case by forming two groups A and *Not A*. The time to complete voting in the two-party model depends on the way the opinions are initially distributed in the graph. For the class of expanders we study, our result that $C(n) = O(n)$ implies that voting completes in $O(n)$ expected time irrespective of the number of opinions.

Let $H_{u,v}$ denote the *hitting time* of vertex v starting from vertex u , that is, the random variable which gives the time taken by a random walk starting from vertex u to reach vertex v ; and let $h_{\max} = \max_{u,v} \mathbf{E}(H_{u,v})$. Aldous [3] showed that $C(2) = O(h_{\max})$, which implies that $C(n) = O(h_{\max} \log n)$ (since the number of particles halves in $O(h_{\max})$ steps), and conjectured that $C(n)$ is actually $O(h_{\max})$. Cox's results [19] imply that the conjecture $C(n) = O(h_{\max})$ is true for constant dimension tori and grids.

In the same paper [3], Aldous also states a lower bound for $C(2)$. For graphs, this bound can be simplified to $C(2) = \Omega(m/\Delta)$, where Δ is the maximum degree of a vertex in G . For the class of expanders we study in this paper, this gives $C(2) = \Theta(n)$. However, the bounds $C(2) = \Omega(m/\Delta)$ and $C(2) = O(h_{\max})$ can be far apart. For example, for a star graph (with loops), $C(2) = \Theta(1)$ whereas the bounds give $\Omega(1) \leq C(2) = O(n)$.

Aldous and Fill [2] showed that for regular graphs $C(n) \leq e(\log n + 2)h_{\max}$, for d -regular s -edge connected graphs $C(n) \leq \frac{dn^2}{4s}$, and for complete graphs $C(n) \sim n$ (where $f(n) \sim g(n)$ means that $f(n) = (1 \pm o(1))g(n)$).

Cooper *et al.* [18] showed that the conjecture $C(n) = O(h_{\max})$ is true for the family of random regular graphs. They proved that for r -regular random graphs, $C(n) = \mathbf{E}(C_V) \sim 2((r-1)/(r-2))n$, **whp**. Such graphs are classic expanders in the sense of [32], and rapidly mixing (mixing time

$T = O(\log n)$.

As noted above, voting processes can be viewed as consensus or aggregation. There is a large amount of research focusing on distributed selection and aggregation in different scenarios and various settings (see e.g. [37, 39] or [7] for a survey). If two or more opinions are canvassed at each step then the time to complete voting can reduce from $O(n)$ to $O(\log n)$. As an example, we mention the result of [22]. At the beginning each vertex of a complete graph has an own opinion. Then, in each step every vertex contacts two neighbours uniformly at random, and changes its opinion to the median of the opinions of these two vertices and its own opinion. It is shown that in time $O(\log n)$ all nodes will have the same opinion, **whp**.

Cover time of multiple random walks

The simplest application of multiple particle walks, is to the speed-up of cover time of a graph G . Using k independent random walks to improve s - t connectivity testing was initially studied by Broder et al. [14]. They proved that for k random walks starting from (positions sampled from) the stationary distribution, the cover time of an m edge graph is $O((m^2 \log^3 n)/k^2)$. In the case of r -regular graphs, Aldous and Fill [2] give an upper bound on the cover time of $C_k \leq (25 + o(1))n^2 \log^2 n/k^2$, which holds for $k \geq 6 \log n$.

Subsequent to this, the value of $C_k(G)$ was studied by Alon et al. [5] for general classes of graphs. They found that for expanders the speed-up was $\Omega(k)$ for $k \leq n$ particles. They also give an example, the barbell graph (two cliques joined by a long path), and a starting position, for which a speed-up exponential in k is obtained, provided $k \geq 20 \log n$.

In the case of random r -regular graphs, [18] establish the k -particle cover time. $C(G)(k) \sim C(G)/k$ independently of any arbitrary choice of k starting positions; i.e., the speedup is exactly linear, as is the case for the complete graph.

If we consider k particles as starting from the same vertex, the speed-up is defined as the ratio of the cover time of a single random walk, to the cover time of the k random walks, i.e. from the worst case starting position. General results for seed-up in this model, were obtained by [26] and [25]. For example [26] present a lower bound on speed-up that depends on the mixing time, and give a $\Omega(k)$ speed-up for many graphs, even when k is as large as n . They prove that the speed-up is $O(k \log n)$, [26] on any graph, or $O(k \log n, k^2)$ [25]. For a large class of graphs [26] improve this bound to $O(k)$, matching a conjecture of Alon et al. [5]

Multiple random walks with information passing

We consider the problem of passing messages between particles moving randomly on a graph. We assume that when particles meet at a vertex, they exchange all messages which they know. We refer to such particles as *agents* to distinguish them from non-communicating particles. If initially one agent has a message it wants to pass to all the others, we refer to this process as *broadcasting* (among the agents). Formally, there are two sets $I(t), U(t)$ of informed and uninformed vertices, respectively. Initially $I = \{\rho_1\}$, where ρ_1 is the agent with the message, and $U = \{\rho_2, \dots, \rho_k\}$. If a member of I meets a member ρ of U , then ρ becomes informed and is moved from U to I . The broadcast time is the step at which $U = \emptyset$.

Dimitriou et al. [23] obtained a general bound of $O(M \log k)$ for broadcasting among k particles where $M = \max(M_{i,j})$ is the expected meeting time of two random walks starting from worst case positions i, j . Pettarin et al. [42] studied problem for the n -vertex toroidal grid, They prove that with k agents, broadcasting and gossiping complete in $\tilde{\Theta}(n/\sqrt{k})$, where the $\tilde{\Theta}(\cdot)$ notation allows $\text{poly}(\log n)$ error terms.

The problem was studied in detail for random regular graphs in [18], where the following **whp** results were obtained for $k \leq n^\epsilon$ agents (for a sufficiently small positive constant ϵ) starting from *general position*; i.e. if there is a pairwise separation $d(v_i, v_j) \geq \Omega(\ln \ln n + \ln k)$ between the starting positions.

Let B_k be the time taken for a given agent to broadcast to all other agents. Then $\mathbf{E}(B_k) \sim 2(r-1)/(r-2) H_{k-1} \cdot n/k$, where H_k is the k -th harmonic number. An alternative and less efficient way to pass on a message is for the originating agent to tell it directly to all the others. In this case message passing completes in time $D(k)$ where $\mathbf{E}(D_k) \sim (r-1)/(r-2) H_{k-1} \cdot n$. Compared to this, broadcasting improves the expected time for everybody to receive the message by a multiplicative factor of $k/2$. Finally, suppose each agent has a message it wants to pass to all other agents, a process of *gossiping*. Let $k \rightarrow \infty$; then **whp** gossiping among the agents can be completed in time $O(n(\log k)^2/k)$.

5 Searching dynamic graphs

In this section we consider random walks on dynamic graphs. There are two cases, either the number $n(t)$ of vertices in the graph varies over time t , or n is fixed, but the structure alters. In both cases there are many possible

models.

In the model of Avin et al. [8], an evolving graph $G(t)$ is a graph sampled at each step from a given space of graphs, using an agreed probability distribution. For any d -regular connected non-bipartite evolving graph G the cover time of the simple random walk on G is $O(d^2 n^3 \log^2 n)$. The general case is $C(G(t)) = O(\Delta^2 n^3 \log^2 n)$.

A more restrictive model $G(t)$ is one in which non-edges are inserted with probability p , and existing edges removed with probability q , at any step. The starting graph $G(0)$ is an Erdős-Renyi graph $G_{n,\hat{p}}$, with $\hat{p} = p/(p+q)$. With these conditions, $G(t)$ is in $G_{n,\hat{p}}$ for all t . Indeed $\Pr(e(t)) = (1-\hat{p})p + \hat{p}(1-q) = \hat{p}$, for any edge $e(t)$. Clementi et al. [15] investigated flooding in this model, and Baumann et al [10] refined this analysis for fixed depth flooding.

Koba et al. [38] studied two types of random walks on a related process, for random subgraphs of arbitrary connected graphs H (i.e. $H = K_n$ for $G_{n,\hat{p}}$). Each existing edge is retained with probability p at each step. In the case of \hat{p} constant, they consider the following strategies, *CBC: choose destination before checking*, and *CAC: choose destination after checking*. For *CBC* they proved $C(H(t)) = C(H)/p$ among other results. Let $q = 1 - p$ then for *CAC*, they prove that $C(H)/(1 - q^\Delta) \leq C(H(t)) \leq C(H)/(1 - q^\delta)$, where $C(H)$ is cover time of H and δ, Δ are minimum and maximum degrees of H respectively. Thus, when H is d -regular, $C(H)/(1 - q^d) = C(H(t))$.

If we consider a random graph process $(G(t), t = 0, 1, \dots)$ in which the graph evolves at each step by the addition of new vertices and edges then the random walk is searching a growing graph, so we cannot hope to visit all vertices of the graph. For example, consider a simple model of search, on e.g. the WWW, in which a particle (which we call a spider) makes a random walk on the nodes of an undirected graph process. As the spider is walking the graph is growing, and the spider makes a random transition to whatever neighbours are available at the time. For simplicity, we assume that the growth rate of the process and the transition rate of the random walk are similar, so that the spider has at least a chance of crawling a constant proportion of the process. Although the edges of the WWW graph are directed, the idea of evaluating models of search on an undirected process has many attractions, not least its simplicity.

In [16] a study was made of the success of the spider's search on comparable graph processes of two distinct types: a random graph process and a web graph process. At each step a new vertex is added which directs m edges towards existing vertices, either choosing vertices randomly (giving a

random graph process) or preferentially according to vertex degree (giving a web graph process). Once a vertex has been added the direction of the edges is ignored.

We consider the following models for the graph process $G(t)$. Let $m \geq 1$ be a fixed integer. Initially $G(1)$ consists of a single vertex v_1 plus m loops. For $t \geq 2$, $G(t+1)$ is obtained from $G(t)$ by adding the vertex v_{t+1} and m randomly chosen edges $\{v_{t+1}, v_i\}, i = 1, 2, \dots, m$, as follows.

Model 1: Random graph. Vertices v_1, v_2, \dots, v_m are chosen independently and uniformly with replacement from $\{1, \dots, t\}$.

Model 2: Web-graph. Vertices v_1, v_2, \dots, v_m are chosen proportional to their degree after step t . Thus if $d(v, \tau)$ denotes the degree of vertex v in $G(\tau)$ then for $v \in \{1, \dots, t\}$ and $i = 1, 2, \dots, m$,

$$\Pr(v_i = v) = \frac{d(v, t)}{2mt}.$$

While vertex t is being added, the spider \mathcal{S} is sitting at some vertex X_{t-1} of $G(t-1)$. After the addition of vertex t , and before the beginning of step $t+1$, the spider now makes a random walk of length ℓ , where ℓ is a fixed positive integer independent of t .

Let $\eta_{\ell, m}(t)$ be the expected proportion of vertices which have not been visited by the spider at step t , when t is large. If we allow $m \rightarrow \infty$ we can get precise asymptotic values. Let $\eta_\ell = \lim_{m \rightarrow \infty} \eta_{\ell, m}$, then

(a) For Model 1,

$$\eta_\ell = \sqrt{\frac{2}{\ell}} e^{(\ell+2)^2/(4\ell)} \int_{(\ell+2)/\sqrt{2\ell}}^{\infty} e^{-y^2/2} dy, \quad \eta_1 = 0.57 \dots, \text{ and } \eta_\ell \sim 2/\ell \text{ as } \ell \rightarrow \infty.$$

(b) For Model 2

$$\eta_\ell = e^\ell 2\ell^2 \int_\ell^{\infty} y^{-3} e^{-y} dy, \quad \eta_1 = 0.59 \dots, \text{ and } \eta_\ell \sim 2/\ell \text{ as } \ell \rightarrow \infty.$$

So for large m, t and $\ell = 1$ it is slightly harder for the spider to crawl on a web-graph whose edges are generated by a copying process (Model 2) than on a uniform choice random graph (Model 1).

References

- [1] M. Adler, E. Halperin, R.M. Karp, and V.V. Vazirani. *A stochastic process on the hypercube with applications to peer-to-peer networks*. Proc. of STOC'03, pp. 575–584, 2003.
- [2] D. Aldous, J. Fill. *Reversible Markov Chains and Random Walks on Graphs*, 2001.
<http://stat-www.berkeley.edu/users/aldous/RWG/book.html>
- [3] D. Aldous. *Meeting times for independent Markov chains*. Stochastic Processes and their Applications, 38 pp. 185-193, 1991.
- [4] R. Aleliunas, R.M. Karp, R.J. Lipton, L. Lovász and C. Rackoff. *Random Walks, Universal Traversal Sequences, and the Complexity of Maze Problems*. Proc. of FOCS'79, pp. 218-223, 1979.
- [5] N. Alon, C. Avin, M. Koucký, G. Kozma, Z. Lotker, and M. Tuttle, *Many random walks are faster than one*, Proc. of SPAA'08, pp. 119–128, 2008.
- [6] N. Alon, I. Benjamini, E. Lubetzky, and S. Sodin. *Non-backtracking random walks mix faster*. Communications in Contemporary Mathematics, 9, pp. 585–603, 2007.
- [7] J. Aspnes. *Randomized protocols for asynchronous consensus*. Distributed Computing, 16, pp. 165-176, 2003.
- [8] C. Avin, M. Koucký, and Z. Lotker. *How to Explore a Fast-Changing World (Cover Time of a Simple Random Walk on Evolving Graphs)* Proc. of ICALP'08, pp121-132, 2008.
- [9] C. Avin and B. Krishnamachari. *The Power of Choice in Random Walks: An Empirical Study*. Proc. of MSWiM-06, pp. 219-228, 2006.
- [10] H. Baumann, P. Crescenzi and P. Fraigniaud. *Parsimonious Flooding in Dynamic Graphs*. Proc. of PODC'09, pp. 260-269, 2009.
- [11] S. N. Bhatt, S. Even, D. S. Greenberg, and R. Tayar. *Traversing directed Eulerian mazes*. Journal of Graph Algorithms and Applications, 6(2):157–173, 2002.

- [12] P. Berenbrink, C. Cooper, R. Elsässer, T. Radzik, T. Sauerwald. *Speeding up random walks with neighborhood exploration*. Proc. of SODA'10, pp 1422–1435, 2010.
- [13] P. Berenbrink, C. Cooper and T. Friedetzky. *Random walks which prefer unexplored edges can cover in linear time*. (preprint) 2011.
- [14] A. Broder, A. Karlin, A. Raghavan, and E. Upfal, *Trading space for time in undirected $s - t$ connectivity*. Proc of STOC'89, pp. 543–549, 1989.
- [15] A. Clementi, C. Macci, A. Monti, F. Pasquale, and R. Silvestri. *Flooding Time in Edge- Markovian Dynamic Graphs*. Proc. of PODC'08 , pp. 213–222, 2008.
- [16] C. Cooper, A. Frieze. *Crawling on simple models of web-graphs*. *Internet Mathematics* 1, 57-90 (2003).
- [17] C. Cooper and A. Frieze. *The cover time of random regular graphs*. *SIAM Journal on Discrete Mathematics*, 18, 728-740 (2005).
- [18] C. Cooper, A. M. Frieze, and T. Radzik. *Multiple Random Walks in Random Regular Graphs*. *SIAM J. Discrete Math.*, 23(4) (2009), 1738-1761.
- [19] J. T. Cox. *Coalescing random walks and voter model consensus times on the torus in \mathbb{Z}^d* . *The Annals of Probability*, 17(4) (1989), 1333-1366.
- [20] J. Czyzowicz, S. Dobrev, L. Gasieniec, D. Ilcinkas, J. Jansson, R. Klasing, I. Lignos, R. Martin, K. Sadakane and W. Sung. *More Efficient Periodic Traversal in Anonymous Undirected Graphs*. In proceedings of the sixteenth International Colloquium on Structural Information and Communication Complexity, 167-181, 2009.
- [21] J. Ding, J. Lee and Y. Peres. *Cover times, blanket times, and majorizing measures* <http://arxiv.org/abs/1004.4371v3>
- [22] B. Doerr, L.A. Goldberg, L. Minder, T. Sauerwald, and C. Scheideler: *Stabilizing Consensus with the Power of Two Choices*. (Manuscript), 2010, full version available at www.upb.de/cs/scheideler.

- [23] T. Dimitriou, S. Nikolettseas and P. Spirakis. *The infection time of graphs*. Discrete Applied Mathematics 154 (2006) 2577–2589.
- [24] P. Donnelly and D. Welsh. *Finite particle systems and infection models*. Math. Proc. Camb. Phil. Soc. 94 (1983), 167-182.
- [25] K. Efremenko and O. Reingold. *How well do random walks parallelize?* Proc. of RANDOM'09, pp. 376–389, 2009.
- [26] R. Elsässer and T. Sauerwald. Tight Bounds for the Cover Time of Multiple Random Walks. In *ICALP 2009*, 415–426, 2009.
- [27] U. Feige, *A tight upper bound for the cover time of random walks on graphs*. Random Structures and Algorithms 6 51–54 (1995).
- [28] U. Feige, *A tight lower bound for the cover time of random walks on graphs*. Random Structures and Algorithms 6 433–438 (1995).
- [29] C. Gkantsidis, M. Mihail, A. Saberi. *Random walks in peer-to-peer networks: algorithms and evaluation*. Perform. Eval. 63.3 241–263 (2006).
- [30] L. Gasieniec and T. Radzik. *Memory Efficient Anonymous Graph Exploration*. Graph-Theoretic Concepts in Computer Science, LNCS 5344, 14–29 (2008).
- [31] Y. Hassin and D. Peleg. Distributed probabilistic polling and applications to proportionate agreement. *Information & Computation* 171 (2002), 248-268.
- [32] S. Hoory, N. Linial, and A. Wigderson: Expander Graphs and their Applications. *Bulletin of the American Mathematical Society*, 43 (2006), 439-561.
- [33] S.Ikeda, I, Kubo, N.Okumoto, and M. Yamashita, *Fair circulation of a token*. IEEE Transactions on Parallel and Distributed Systems. 13.4 (2002).
- [34] S.Ikeda, I, Kubo, N.Okumoto, and M. Yamashita, *Impact of Local Topological Information on Random Walks on Finite Graphs*. In proceedings of the 32st International Colloquium on Automata, Languages and Programming (ICALP), 1054-1067, (2003).

- [35] A. Israeli and M. Jalfon. *Token management schemes and random walks yeild self stabilizing mutual exclusion*. Proc. of PODC'90, pp. 119-131, 1990.
- [36] J. Kahn, J. H. Kim, L. Lovasz, and V. H. Vu. *The cover time, the blanket time, and the Matthews bound* Proc. of FOCS'00, pp 467-475, 2000.
- [37] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. *Proc. of FOCS'03*, pp. 482-491, 2003.
- [38] K. Koba, S. Kijima and M. Yamashita. *Random walks on dynamic graphs*. preprint 2010 (in Japanese).
- [39] F. Kuhn, T. Locher, and R. Wattenhofer. Tight bounds for distributed selection. *Proc. of SPAA '07*, pp. 145-153, 2007.
- [40] L. Lovász. *Random walks on graphs: A survey*. Bolyai Society Mathematical Studies, 2:353–397, Budapest, 1996.
- [41] T. Nakata, H. Imahayashi, and M. Yamashita. Probabilistic local majority voting for the agreement problem on finite graphs. In *Proc. 5th Computing and Combinatorics Conference - COCOON'99*. Springer LNCS vol. 1627, pp. 330-338, 1999.
- [42] A. Pettarin, A. Pietracaprina, G. Pucci, E. Upfal. *Infectious Random Walks*. <http://arxiv.org/abs/1007.1604v2>
- [43] V.B. Priezzhev, D. Dhar, A. Dhar, S. Krishnamurthy. *Eulerian walkers as a model of self organized criticality*. In Physics Review Letters 77:5079–5082 (1996)
- [44] I. A. Wagner, M. Lindenbaum, and A. M. Bruckstein. *Distributed Covering by Ant-Robots Using Evaporating Traces*. IEEE Transactions on Robotics and Automation, 15(5):918–933, 1999.
- [45] V. Yanovski, I. A. Wagner, and A. M. Bruckstein. *A Distributed Ant Algorithm for Efficiently Patrolling a Network*. Algorithmica, 37:165–186, 2003.