Robustness of Randomized Rumour Spreading

Rami Daknama
Department of Mathematics, Ludwig-Maximilians-Universität München, Germany

Konstantinos Panagiotou
Department of Mathematics, Ludwig-Maximilians-Universität München, Germany

Simon Reisser
Department of Mathematics, Ludwig-Maximilians-Universität München, Germany

Abstract

In this work we consider three well-studied broadcast protocols: *push*, *pull* and *push&pull*. A key property of all these models, which is also an important reason for their popularity, is that they are presumed to be very robust, since they are simple, randomized, and, crucially, do not utilize explicitly the global structure of the underlying graph. While sporadic results exist, there has been no systematic theoretical treatment quantifying the robustness of these models. Here we investigate this question with respect to two orthogonal aspects: (adversarial) modifications of the underlying graph and message transmission failures.

We explore in particular the following notion of *local resilience*: beginning with a graph, we investigate up to which fraction of the edges an adversary may delete at each vertex, so that the protocols need significantly more rounds to broadcast the information. Our main findings establish a separation among the three models. On one hand *pull* is robust with respect to all parameters that we consider. On the other hand, *push* may slow down significantly, even if the adversary is allowed to modify the degrees of the vertices by an arbitrarily small positive fraction only. Finally, *push&pull* is robust when no message transmission failures are considered, otherwise it may be slowed down.

On the technical side, we develop two novel methods for the analysis of randomized rumour spreading protocols. First, we exploit the notion of self-bounding functions to facilitate significantly the round-based analysis: we show that for any graph the variance of the growth of informed vertices is bounded by its expectation, so that concentration results follow immediately. Second, in order to control adversarial modifications of the graph we make use of a powerful tool from extremal graph theory, namely Szemerédi's Regularity Lemma.

2012 ACM Subject Classification Mathematics of computing → Probabilistic algorithms; Theory of computation → Graph algorithms analysis

Keywords and phrases Rumour Spreading, Local Resilience, Robustness, Self-bounding Functions, Szemerédi's Regularity Lemma

Digital Object Identifier 10.4230/LIPIcs.ESA.2019.36

Related Version A full version of the paper is available at `https://arxiv.org/abs/1902.07618`.

1 Introduction

Randomized broadcast protocols are highly relevant for data distribution in large networks of various kinds, including technological, social and biological networks. Among many others there are three basic models in the literature, introduced in [19, 9, 24], namely *push*, *pull* and *push&pull* (or short *pp*). Consider a connected graph in which some vertex holds a piece of information; we call this vertex (initially) informed. All three models have the common characteristic that they proceed in rounds. In the *push* model, in every round every informed vertex chooses a neighbour independently and uniformly at random (iuar) and informs it; this of course has only an effect if the target vertex was previously uninformed. Contrary, in the *pull* model every round every *un*informed vertex chooses a neighbour iuar and asks for the information. If the asked vertex has the information, then the asking vertex becomes informed as well. The third model *push&pull* combines both worlds: in each round, each vertex chooses a neighbour iuar, and if one of both vertices is informed, then afterwards both become so. We additionally assume that each message transmission succeeds independently with probability $q \in (0, 1]$. For these algorithms, the main parameter that we consider is the random variable that counts how many rounds are needed until all vertices are informed, and we call these quantities the *runtimes* of the respective algorithms.

In the remainder we will denote the runtime of *push* by $T_{push}(G, v, q)$ where G is the underlying graph, initially the vertex v is informed and we have a transmission success probability of $q \in (0, 1]$. Analogously we denote the runtimes of *pull* and *push&pull* by $T_{pull}(G, v, q)$ and $T_{pp}(G, v, q)$ respectively. If the choice of v does not matter we will omit it in our notation. The most basic case is when G is the complete graph K_n with n vertices. Then, see for example Doerr and Kostrygin [11], it is known that for $\mathcal{P} \in \{push, pull, pp\}$ and $q \in (0, 1]$ in expectation and with probability tending to 1 as $n \to \infty$

$$T_{\mathcal{P}}(K_n, q) = c_{\mathcal{P}}(q) \log n + o(\log n),$$

where, for $q \in (0, 1)$,

$$c_{push}(q) := \frac{1}{\log(1+q)} + \frac{1}{q}, \quad c_{pull}(q) := \frac{1}{\log(1+q)} - \frac{1}{\log(1-q)},$$
$$c_{pp}(q) := \frac{1}{\log(1+2q)} + \frac{1}{q - \log(1-q)},$$

and where we set $c_{\mathcal{P}}(1) := \lim_{q \to 1} c_{\mathcal{P}}(q)$. If q is clear from the context, we write $c_{\mathcal{P}}$ instead of $c_{\mathcal{P}}(q)$. Actually, the results in [11] and also [12] are much more precise, but the stated forms will be sufficient for what follows.

Contribution & Related Work

In this article our focus is on quantifying the *robustness* of all three models. Indeed, robustness is a key property that is often attributed to them, since they are simple, randomized, and, crucially, do not exploit explicitly the structure of the underlying graph (apart, of course, from considering the neighborhoods of the vertices). Clearly, the runtime can vary tremendously between different graphs with the same number of vertices. Hence it is essential to understand which structural characteristics of a graph influence in what way the runtime of rumour spreading algorithms.

One result in this spirit for the *push* model was shown in [25]. Roughly speaking, in that paper it is shown that even on graphs with low density, if the edges are distributed rather uniformly, then *push* is as fast as on the complete graph. This can be interpreted as a robustness result: starting with a complete graph, one can delete a vast amount of edges and as long as this is done rather uniformly, the runtime of *push* is affected insignificantly. To state the result more precisely, we need the following notion.

▶ **Definition 1** ($(n, \delta, \Delta, \lambda)$-graph)**.** *Let G be a connected graph with n vertices that has minimum degree δ and maximum degree Δ. Let $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$ be the eigenvalues of the adjacency matrix of G, and set $\lambda = \max_{2 \leq i \leq n} |\mu_i| = \max\{|\mu_2|, |\mu_n|\}$. We will call G an $(n, \delta, \Delta, \lambda)$-graph.*

In this paper we are interested in the case where G gets large, that is, when $n \to \infty$. Hence all asymptotic notation in this paper is with respect to n; in particular "with high probability", or short whp, means with probability $1 - o(1)$ when $n \to \infty$.

▶ **Definition 2** (Expander Sequence). *Let $\mathcal{G} = (G_n)_{n\in\mathbb{N}}$ be a sequence of graphs, where G_n is a $(n, \delta_n, \Delta_n, \lambda_n)$-graph for each $n \in \mathbb{N}$. We say that $\mathcal{G}$ is an* expander sequence *if $\Delta_n/\delta_n = 1 + o(1)$ and $\lambda_n = o(\Delta_n)$.*

Note that if we consider any sequence $\mathcal{G} = (G_n)_{n\in\mathbb{N}}$ of graphs this always implicitly defines δ_n, Δ_n and λ_n as in Definition 2. Expander graphs have found numerous applications in computer science and mathematics, see for example the survey [23]. If $\mathcal{G}$ is an expander sequence, then intuitively this means that for n large enough, the edges of G_n are rather uniformly distributed. For a more formal statement see Lemma 16. Moreover, note that our definition of expander sequences excludes the case when Δ_n is bounded; this is actually a necessary condition for our robustness results to hold, see [13]. With all these definitions at hand we can state the result from [25] that quantifies the robustness of *push* with respect to the network topology, that is, the runtime is asymptotically the same as on the complete graph K_n.

▶ **Theorem 3.** *Let $\mathcal{G} = (G_n)_{n\in\mathbb{N}}$ be an expander sequence. Then whp*

$$T_{push}(G_n) = c_{push}(1)\log n + o(\log n).$$

Apart from expander sequences, results in the form of Theorem 3 (where the asymptotic runtimes of one or more of these algorithms are determined) were also shown for sufficiently dense Erdös-Renyi random graphs [16], random regular graphs [15] as well as hypercubes [25]. Moreover, the order of the runtime on various models that describe social networks was investigated. In [17] the Chung-Lu model was studied, [10] explored preferential attachment graphs and [18] examined geometric graphs. A somewhat different approach is to derive general runtime bounds that hold for all graphs and depend only on some graph parameter, e.g. conductance [20, 6], vertex expansion [21] or diameter [14, 5, 22]. Furthermore, several variants of *push*,*pull* and *push&pull* were studied. These include vertices being restricted to answer only one *pull* request per round [7], vertices being allowed to contact multiple neighbours per round [25, 11], vertices not calling the same neighbour twice [10] and asynchronous versions [4, 26, 1, 2]. Finally, besides [11], robustness of these rumor spreading algorithms with respect to message transmission failures was also studied by Elsässer and Sauerwald in [13]. It was shown for any graph that if a message fails with probability $1 - p$, then the runtime of *push* increases at most by a factor of $6/p$.

In this work our focus is on three subjects concerning the robustness of rumour spreading. Our first (and not unexpected) result extends Theorem 3 to the runtimes of *pull* and *push&pull*. In particular, we show that none of the three protocols slows down or speeds up on graphs with good expansion properties compared to its runtime on the complete graph. This motivates to investigate how severely a graph with good expansion properties has to be modified to increase the respective runtimes.

In our second contribution, which is also the main result and which differs from what was treated in previous works, we propose and study a novel approach to quantifying robustness. In particular, we investigate the impact of adversarial edge deletions, where we use the well-known concept of *local resilience*, see e.g. [28, 8]. To be specific, we explore up to which fraction of edges an adversary may delete at each vertex to slow down the process by a significant amount of time, i.e., by $\Omega(\log n)$ rounds. Here we discover a surprising dichotomy in the following sense. On the one hand, we show that both *pull* and *push&pull* cannot be slowed down by such adversarial edge deletions – in essentially all but trivial cases, where the fraction is so large that the graph may become (almost) disconnected. On the other hand, we demonstrate that even a small number of edge deletions is sufficient to slow down *push* by $\Omega(\log n)$ rounds. In other words, we find that in contrast to *pull* and *push&pull*, the *push* protocol is not resilient to adversarial deletions and lacks (in this specific sense) the robustness of the other two protocols.

As our third subject, we generalise the previous results by additionally considering message transmission failures that occur independently with probability $1-q \in [0,1)$. On the positive side, we show that for arbitrary $q \in (0,1]$ all three algorithms inform *almost* all vertices at least as fast as when run on expander sequence in spite of adversarial edge deletions. However, if we want to inform all vertices, only *pull* is not slowed down by adversarial edge deletions for all values of q; *push* can be slowed down as before; and *push&pull* is a mixed bag, for $q = 1$ it cannot be slowed down, for $q < 1$ it can. Furthermore, in general it is also possible to speed *push&pull* up by deleting edges, which is however not surprising as the star-graph deterministically finishes in at most 2 rounds.

Summarizing, this work expands previous (robustness) results, particularly the ones concerning precise asymptotic runtimes and random transmission failures. Crucially, we introduce and study the concept of local resilience as a method to investigate robustness. However, apart from that, in this paper we develop two new general methods for the analysis of rumour spreading algorithms.

- The most common approach in the current literature for the study of the runtime is to determine the expected number of newly informed vertices in one or more rounds and to show concentration, for example by bounding the variance. Achieving this, however, is often quite complex and makes laborious and lengthy technical arguments necessary. Here we use the theory of *self-bounding* functions, see Section 2, that allows us to cleanly upper bound the variance by the *expected value*. The argument works for all three investigated algorithms and the bound is valid for all graphs. We are certain that this method will also facilitate future work on the analysis of rumour spreading algorithms.
- Studying the robustness of the protocols is a challenging task, as the adversary (as described previously) has various options to modify the graph, for example by introducing a high variance in the degrees of the vertices; this turns out to be particularly problematic in the case of *push&pull*. Here we demonstrate that such types of irregularities can be handled universally by applying a powerful tool from a completely different area, namely extremal graph theory. In particular, we use Szemerédi's regularity lemma (see e.g. [27]), which allows us to partition the vertex set of a graph such that nearly all pairs of sets in the partition behave nearly like perfect regular bipartite graphs. This allows us to apply our methods on these regular pairs; eventually we obtain a linear recursion that can be solved by analysing the maximal eigenvalue of the underlying matrix.

1.1 Results

Our first result addresses the question about how fast rumours spread on expander graphs; in order to obtain a concise statement also the occurrence of independent message transmission failures is considered.

▶ **Theorem 4.** *Let $\mathcal{G} = (G_n)_{n\in\mathbb{N}}$ be an expander sequence and let $q \in (0,1]$. Then whp*
(a) $T_{push}(G_n, q) = c_{push}(q) \log n + o(log(n))$,
(b) $T_{pull}(G_n, q) = c_{pull}(q) \log n + o(log(n))$,
(c) $T_{pp}(G_n, q) = c_{pp}(q) \log n + o(log(n))$.

The first statement is an extension of Theorem 3 and its proof is a straigthforward adaptation of the proof in [25]. We omit it. The contribution here is the proof of (b) and (c). Next we consider the case with edge deletions in addition to the message transmission failures.

▶ **Theorem 5.** *Let* $0 < \varepsilon < 1/2, q \in (0,1]$ *and* $\mathcal{G} = (G_n)_{n\in\mathbb{N}}$ *be an expander sequence. Let* $\tilde{\mathcal{G}} = (\tilde{G}_n)_{n\in\mathbb{N}}$ *be such that each* $\tilde{G}_n$ *is obtained by deleting edges of* G_n *such that each vertex keeps at least a* $(1/2+\varepsilon)$ *fraction of its edges. Then whp*

(a) $T_{pull}(\tilde{G}_n, q) = c_{pull}(q)\log n + o(\log n)$.

(b) $T_{pp}(\tilde{G}_n, 1) \leq c_{pp}(1)\log n + o(\log n)$, *when additionally assuming that* $\delta(G_n) \geq \alpha n$ *for some constant* $0 < \alpha \leq 1$.

This result demonstrates uncoditionally the robustness of *pull*, and conditionally on $q = 1$ the robustness of *push&pull* on dense graphs, in the case of edge deletions, that is, the runtime is asymptotically the same as in the complete graph. It even shows that *push&pull* may potentially profit from edge deletions in contrast to being slowed down. The proof of this result, especially the statement about *push&pull*, is rather involved, since the original graph may become quite irregular after the edge deletions. Here we use, among many other ingredients, the aforementioned decomposition of the graph given by Szemeredi's regularity lemma.

Note that Theorem 5 does not consider *push* and *push&pull* (when $q \neq 1$) at all. Indeed, our next result states that in these cases the behaviour is rather different and that the algorithms may be slowed down.

▶ **Theorem 6.** *Let* $\varepsilon > 0$ *and* $q \in (0,1]$*. Then there is an expander sequence* $\mathcal{G} = (G_n)_{n\in\mathbb{N}}$ *and a sequence of graphs* $\tilde{\mathcal{G}} = (\tilde{G}_n)_{n\in\mathbb{N}}$ *with the following properties. Each* $\tilde{G}_n$ *is obtained by deleting edges of* G_n *such that each vertex keeps at least a* $(1-\varepsilon)$ *fraction of its edges. Moreover, whp*

(a) $T_{push}(\tilde{G}_n, q) \geq c_{push}(q)\log n + \varepsilon/(2q)\log n + o(\log n)$.

(b) $T_{pp}(\tilde{G}_n, q) \geq c_{pp}(q)\log n + \big(\varepsilon/(8q) - \varepsilon q^3/5\big)\log n + o(\log n)$.

Nevertheless, not all hope is lost. On the positive side, the next result states that *push* and *push&pull* are able to inform *almost* all vertices as fast as on the complete graph in spite of adversarial edge deletions. In this sense, we obtain an almost-robustness result for these cases.

▶ **Theorem 7.** *Let* $0 < \varepsilon < 1/2, q \in (0,1]$ *and* $\mathcal{G} = (G_n)_{n\in\mathbb{N}}$ *be an expander sequence. Let* $\tilde{\mathcal{G}} = (\tilde{G}_n)_{n\in\mathbb{N}}$ *be such that each* $\tilde{G}_n$ *is obtained by deleting edges of* G_n *such that each vertex keeps at least a* $(1/2+\varepsilon)$ *fraction of its edges. For* $\mathcal{P} \in \{push, pp\}$ *let* $\tilde{T}_{\mathcal{P}}$ *denote the number of rounds needed to inform at least* $n - n/\log n$ *vertices. Then whp*

(a) $\tilde{T}_{push}(\tilde{G}_n) = \log_{1+q}(n) + o(\log n)$.

(b) $\tilde{T}_{pp}(\tilde{G}_n) \leq \log_{1+2q}(n) + o(\log n)$, *when additionally assuming that* $\delta(G_n) \geq \alpha n$ *for some constant* $0 < \alpha \leq 1$.

We conjecture that there is also a version of Theorem 7b that is true for *push&pull* on sparse graphs; to be precise we conjecture that in the setting of Theorem 7b it is $\tilde{T}_{pp}(\tilde{G}_n) \leq \log_{1+2q}(n) + o(\log n)$, without further restrictions on G_n, i.e. that *push&pull* cannot be slowed down informing *almost* all vertices.

As a final remark note that Theorems 5 and 7 are tight in the sense that if an adversary may delete up to half of the edges at each vertex, then there are expander graphs that become disconnected. On those graphs a linear fraction of the vertices will remain uninformed forever.

Outline

The rest of this paper is structured as follows. The first part of Section 2 contains our technical contribution concerning the analysis through self-bounding functions. In the second part we state the Expander Mixing Lemma and give some applications to our setting with deleted edges. The remaining sections contain the proofs to the main theorems. The proof of Theorem 4 has two steps: determining the expected growth rates of the number of informed vertices after performing one round, then concluding the proof for the runtime by using the tools developed in Section 2. This proof is not included in this version, here instead we focus on the case with edge deletions, where for every protocol we use a different method to show the claimed results. In Subsection 3.1 we show that edge deletions do not slow down *pull*, by analysing the number of edges between informed and uninformed vertices. Showing that adversarial edge deletions cannot slow down the time until *push* has informed almost all vertices will be archived in Section 3.2 by giving a coupling to the case without edge deletions. Then, in Subsection 3.3 we show that *push&pull* informs almost all vertices of dense graphs fast in spite of adversarial edge deletions. We utilize a version of Szemerédis Regularity Lemma to get a well-behaved partition of the vertex set that is suitable for performing a round based analysis. However, if $q < 1$, adversarial edge deletions can slow down the time until *push&pull* has informed all vertices for nearly all values of q; we show this in Section 3.4. The same example as given there also yields Theorem a. Finally, an unabridged version of this paper, that contains any proofs that are omitted here, is available at `https://arxiv.org/abs/1902.07618`.

Further Notation

Let $G = (V, E)$ denote a graph with vertex set V and edge set $E \subseteq \binom{V}{2}$. Consider $v \in V$ and $U, W \subseteq V$ with $U \cap W = \emptyset$. We will denote the set of neighbours of v in G by $N_G(v)$ or by $N(v)$ and we will denote its degree by $d_G(v) := |N_G(v)|$ or by $d(v)$; δ_G or δ and Δ_G or Δ denote minimum and maximum degree of G. Similarly the neighbourhood of any set of vertices $S \subseteq V$ is defined by $N_G(S) := \cup_{v \in S} N_G(v)$. Furthermore let $E(U, W) = E_G(U, W)$ denote the set of edges with one vertex in U and one vertex in W and let $e(U, W) := e_G(U, W) := |E_G(U, W)|$. With $E_G(U)$ we denote the set of edges with both vertices in U; $e_G(U) = |E_G(U)|$. For any round $t \in \mathbb{N}$ and $\mathcal{P} \in \{push, pull, pp\}$, we denote by $I_t^{(\mathcal{P})}(G)$ the set of vertices of G informed by *push*, *pull* and *push&pull* respectively at the beginning of round t and $|I_1^{(\mathcal{P})}| = 1$; if the underlying graph is clear from the context we will omit it; if we consider a sequence of graphs $\mathcal{G} = (G_n)_{n \in \mathbb{N}}$ and a sequence of times $t = (t(n))_{n \in \mathbb{N}}$, then $I_t^{(\mathcal{P})}(\mathcal{G}) = (I_{t(n)}^{(\mathcal{P})}(G_n))_{n \in \mathbb{N}}$ is also a sequence. Similarly, $U_t^{(\mathcal{P})} := V \backslash I_t^{(\mathcal{P})}$ denotes the set of uninformed vertices. With log we refer to the natural logarithm. For any event A we will write $\mathbb{E}_t[A]$ instead of $\mathbb{E}[A \big| I_t]$ for the conditional expectation and $P_t[A]$ instead of $P[A \big| I_t]$ for the conditional probability. Finally we want to clarify our use of Landau symbols. Let $a, b \in \mathbb{R}$ and f be a function. The terms $a \le b + o(f)$ and $a \ge b - o(f)$ mean that there exist positive functions $g, h \in o(f)$ such that $a \le b + g$ and $a \ge b - h$. Consequently $a = b + o(f)$ means that there exists a positive function $g \in o(f)$ such that $a \in [b - g, b + g]$

2 Tools & Techniques

In this section we collect and prove statements about our protocols and properties of expander sequences. We begin with applying the previously mentioned notion of self-bounding functions to derive universal and simple-to-apply concentration results for our random variables, i.e., the number of informed vertices after a particular round. Then we extend the concentration results to more than one round. In the last part we recall the well known Expander Mixing Lemma and utilize it to derive properties (weak expansion, path enumeration) for the case where we delete edges from our graphs.

Self-bounding functions

Our main technical new result in this section is the following bound on the variance for the number of informed vertices in any given round; it is true for any graph and any set of informed vertices.

▶ **Lemma 8.** *Let G be a graph, $t \in \mathbb{N}$ and $I_t = I_t^{(\mathcal{P})}(G)$ for $\mathcal{P} \in \{push, pull, pp\}$. Then*

$$Var\big[|I_{t+1}|\big|I_t\big] \leq \mathbb{E}\big[|I_{t+1}|\big|I_t\big].$$

Lemma 8 follows directly from Lemmas 10 and 11. Before stating them we introduce the notion of self-bounding functions.

▶ **Definition 9** (Self-bounding function). *Let X be a set and $m \in \mathbb{N}$. A non-negative function $f : X^m \to \mathbb{R}$ is self-bounding, if there exist functions $f_i : X^{m-1} \to \mathbb{R}$ such that for all $x_1, ..., x_m \in X$ and all $i = 1, ..., m$*

$$0 \leq f(x_1, ..., x_m) - f_i(x_1, ..., x_{i-1}, x_{i+1}, ..., x_m) \leq 1$$

and

$$\sum_{1 \leq i \leq m} (f(x_1, ..., x_m) - f_i(x_1, ..., x_{i-1}, x_{i+1}, ..., x_m)) \leq f(x_1, ..., x_m).$$

A striking property of self-bounding function is the following bound on the variance.

▶ **Lemma 10** ([3]). *For a self-bounding function f and independent random variables $X_1, ..., X_m$, $m \in \mathbb{N}$*

$$Var\left[f(X_1, ..., X_m)\right] \leq \mathbb{E}\left[f(X_1, ..., X_m)\right].$$

▶ **Lemma 11.** *Let G be a graph, $t \in \mathbb{N}$, and let $I_t = I_t^{(\mathcal{P})}(G)$ for $\mathcal{P} \in \{push, pull, pp\}$. Then, conditional on I_t, there exist $m \in \mathbb{N}$, independent random variables $X_1, ..., X_m$ and a self-bounding function $f = f^{(\mathcal{P})}$ such that $|I_{t+1}| = f(X_1, ..., X_m)$.*

▶ Remark 12. Let $G = (V, E)$ be a graph. Lemma 11 also applies to subsets of I_{t+1}, i.e for any $U \subset V$ and conditioned on I_t we have that $|I_{t+1} \cap U|$ and $|(I_{t+1} \cap U) \setminus I_t|$ are self-bounding.

The following lemma gives a tool that we will use in order to extend our round-wise analysis to longer phases.

▶ **Proposition 13.** *Let $\mathcal{P} \in \{push, pull, pp\}, I_t = I_t^{(\mathcal{P})}$ and $t_1 \geq t_0 \geq 1$ such that $|I_{t_0}| \geq \sqrt{\log n}$. Let further $(\mathcal{A}_i)_{i \in \mathbb{N}}$ be a sequence of events, $c > 1$, and $\delta > 0$ such that*

$$P_{t_0}[\mathcal{A}_t \mid \mathcal{A}_{t_0}, \ldots, \mathcal{A}_{t-1}] \geq 1 - \delta \left(c^{t-t_0}|I_{t_0}|\right)^{-1/3} \quad \text{for all } t_0 \leq t \leq t_1.$$

Then

$$P_{t_0}\left[\bigcap_{t=t_0}^{t_1} \mathcal{A}_t\right] \geq 1 - O(|I_{t_0}|^{-1/3})$$

We give two typical example applications of this lemma below. The first example addresses the case where we have a lower bound for the expected number of informed vertices after one round.

▶ **Example 14.** Let $\mathcal{P} \in \{push, pull, pp\}, I_t = I_t^{(\mathcal{P})}$. Assume that there is some $c > 1$ such that $\mathbb{E}_t[|I_{t+1}|] \geq c\,|I_t|$ for all t as long as $n/f(n) \leq |I_t| \leq n/g(n)$ for some functions $1 \leq f, g \leq n, f = o(n)$. Let t_0 be such that $|I_{t_0}| \geq n/f(n)$. Then according to Lemma 8 we have that $\mathrm{Var}_t[|I_{t+1}|] \leq \mathbb{E}_t[|I_{t+1}|]$ and applying Chebychev's inequality gives

$$P_t\left[\big||I_{t+1}| - \mathbb{E}_t[|I_{t+1}|]\big| \leq \mathbb{E}_t[|I_{t+1}|]^{2/3}\right] \geq 1 - \mathbb{E}_t[|I_{t+1}|]^{-1/3} \geq 1 - |I_t|^{-1/3}. \tag{1}$$

Consider the events

$$\mathcal{A}_t = \text{“}|I_t| \geq \mathbb{E}_{t-1}[|I_t|] - \mathbb{E}_{t-1}[|I_t|]^{2/3} \quad \text{or} \quad |I_t| \geq n/g(n)\text{”}$$

The intersection of $\mathcal{A}_{t_0+1}, \ldots, \mathcal{A}_t$ implies inductively that either $|I_t| \geq n/g(n)$ or

$$|I_t| \geq \left(1 - \mathbb{E}_{t-1}[|I_t|]^{-1/3}\right)\mathbb{E}_{t-1}[|I_t|] \geq \left(\left(1 - (c|I_{t_0}|)^{-1/3}\right)c\right)^{t-t_0}|I_{t_0}|.$$

We obtain with (1)

$$P_{t_0}[\mathcal{A}_{t+1} \mid \mathcal{A}_{t_0+1}, \ldots, \mathcal{A}_t, |I_t| < n/g(n)] \geq 1 - \left(\left(1 - (c|I_{t_0}|)^{-1/3}\right)c\right)^{-(t-t_0)/3}|I_{t_0}|^{-1/3},$$

and otherwise $P_{t_0}[\mathcal{A}_{t+1} \mid \mathcal{A}_{t_0+1}, \ldots, \mathcal{A}_t, |I_t| \geq n/g(n)] = 1$. Choose $\tau := t - t_0 = \log_c(f(n)/g(n)) + o(\log n)$ as small as possible such that this lower bound for $|I_{t+1}|$ is $\geq n/g(n)$, that is, this lower bound is $< n/g(n)$ for $t = t_0 + \tau$. Combining the two conditional probabilities we obtain for all $t_0 \leq t \leq t_0 + \tau$

$$P_{t_0}[\mathcal{A}_{t+1} \mid \mathcal{A}_{t_0+1}, \ldots, \mathcal{A}_t] \geq 1 - \left(\left(1 - (c|I_{t_0}|)^{-1/3}\right)c\right)^{-(t-t_0)/3}|I_{t_0}|^{-1/3}.$$

Applying Proposition 13 then yields whp

$$|I_{t_0+\tau+1}| \geq n/g(n).$$

In the second example we make the stronger assumption that we can determine asymptotically the expected number of informed vertices after one round. Here we assume that we begin with a “small” set of informed vertices, say of size $\sqrt{\log n}$, and want to reach a set of size nearly linear in n.

▶ **Example 15.** Assume that there is some $c > 1$ such that $\mathbb{E}_t[|I_{t+1}|] = (1 + o(1))c\,|I_t|$ for all t as long as $\sqrt{\log n} \leq |I_t| \leq n/\log n$. Let $\mathcal{A}_t$ be the event “$\big||I_t| - \mathbb{E}_{t-1}[|I_t|]\big| \leq \mathbb{E}_{t-1}[|I_t|]^{2/3}$” and let t_0 be such that $|I_{t_0}| \geq \sqrt{\log n}$. There is $h(n) \in o(1)$ such that for $c^- := (1 - h(n))c$ and $c^+ := (1 + h(n))c$ we have that $\mathbb{E}_t[|I_{t+1}|] \leq c^+\,|I_t|$ and $\mathbb{E}_t[|I_{t+1}|] \geq c^-\,|I_t|$. Using this notation, the events $\mathcal{A}_{t_0+1}, \ldots, \mathcal{A}_{t+1}$ imply together inductively that

$$|I_{t+1}| \leq \left(1 + \mathbb{E}_t[|I_{t+1}|]^{-1/3}\right)\mathbb{E}_t[|I_{t+1}|] \leq \left(\left(1 + (c^-|I_{t_0}|)^{-1/3}\right)c^+\right)^{t-t_0}|I_{t_0}|$$

for all t such that the right-hand side is bounded by $n/\log n$. Moreover, for all such t

$$|I_{t+1}| \geq \left(1 - \mathbb{E}_t[|I_{t+1}|]^{-1/3}\right)\mathbb{E}_t[|I_{t+1}|] \geq \left(\left(1 - (c^-|I_{t_0}|)^{-1/3}\right)c^-\right)^{t-t_0}|I_{t_0}|.$$

Thus, as $\mathcal{A}_t$ only depends on I_t it follows with (1)

$$P_{t_0}[\mathcal{A}_{t+1} \mid \mathcal{A}_{t_0+1}, \ldots, \mathcal{A}_t] \geq 1 - \left(\left(1 - (c^-|I_{t_0}|)^{-1/3}\right)c^-\right)^{-(t-t_0)/3}|I_{t_0}|^{-1/3}.$$

Applying Proposition 13 then immediately gives that there is $\tau_1 = \log_c(n/|I_{t_0}|) + o(\log n)$ such that whp $|I_{t_0+\tau_1}| \leq n/\log n$. Example 14, setting $f = n/\sqrt{\log n}$ and $g = \log n$, gives an additional $\tau_2 = \log_c(n/|I_{t_0}|) + o(\log n)$ such that $|\tau_1 - \tau_2| = o(\log n)$ and whp

$$|I_{t_0+\tau_1}| \leq \frac{n}{\log n} \leq |I_{t_0+\tau_2}|.$$

Expander Sequences

In this section we collect some important properties of expander sequences that we are going to use later. We start by stating a version of the well-known expander mixing lemma applied to our setting of expander sequences.

▶ **Lemma 16** ([25, Cor. 2.4]). *Let $\mathcal{G} = (G_n)_{n\in\mathbb{N}} = ((V_n, E_n))_{n\in\mathbb{N}}$ be an expander sequence. Then for $S_n \subseteq V_n$ such that $1 \le |S_n| \le n/2$ it is*

$$\left| e(S_n, V_n \backslash S_n) - \frac{\Delta_n |S_n|(n - |S_n|)}{n} \right| = o(\Delta_n)|S_n|.$$

The following result is a consequence of the Expander Mixing Lemma that applies to graphs in which some edges were removed. It seems very simple but it turns out to be surprisingly useful.

▶ **Lemma 17.** *Let $\mathcal{G} = (G_n)_{n\in\mathbb{N}} = ((V_n, E_n))_{n\in\mathbb{N}}$ be an expander sequence. Let $\varepsilon > 0$ and set $\tilde{\mathcal{G}} = (\tilde{G}_n)_{n\in\mathbb{N}}$, where each $\tilde{G}_n$ it is obtained from G_n by deleting edges such that each vertex keeps at least a $(1/2 + \varepsilon)$ fraction of its edges. For each $n \in \mathbb{N}$ let further $S_n \subseteq V_n$, then there is $n_0 \in \mathbb{N}$ such that for all $n \ge n_0$*

$$e_{\tilde{G}_n}(S_n, V_n \backslash S_n) \ge \varepsilon e_{G_n}(S_n, V_n \backslash S_n).$$

3 Proofs

3.1 Proof of Theorems 4b, 5a – edge deletions do not slow down *pull*

Let $0 < \varepsilon \le 1/2$. In this section we study the runtime of *pull* in the case in which the input graph is an expander, and where at each vertex at most an $(1/2 - \varepsilon)$ fraction of the edges is deleted. The runtime on expander sequences without edge deletions, that is, the setting in Theorem 4b, is included as the special case where we set $\varepsilon = 1/2$. In contrast to previous proofs, in the analysis of *pull* the "standard" approach that consists of showing, for example, that $\mathbb{E}_t[|I_{t+1} \setminus I_t|] \approx |I_t|$ fails. The main reason is that the graph between I_t and U_t might be quite irregular, so that, depending on the actual state, $\mathbb{E}_t[|I_{t+1} \setminus I_t|] \approx c|I_t|$ for some $c < 1$. However, we discover a different invariant that is preserved, namely that the number of edges between I_t and U_t behaves in an exponential way. With Lemmas 16 and 17 we can then relate this to the number of informed vertices.

▶ **Lemma 18.** *Consider the setting of Theorem 5a and let $I_t = I_t^{(pull)}$.*

(a) *Let $\sqrt{\log n} \le |I_t| \le n/\log n$. Then $|e(U_{t+1}, I_{t+1}) - (1+q)e(U_t, I_t)| \le |I_t|^{-1/3} e(U_t, I_t)$ with probability at least $1 - O(|I_t|^{-1/3})$.*

(b) *Let $|U_t| \le n/\log n$. Then $\mathbb{E}_t[|U_{t+1}|] = (1 - q + o(1))|U_t|$.*

Lemma 19 gives a lower bound, that together with an upper bound provided by Lemma 20 imply Theorems 4b and 5a.

▶ **Lemma 19** (Upper bound in Theorem 5a). *Consider the setting of Theorem 5a and let $I_t = I_t^{(pull)}$, then the following statements hold whp.*

(a) *Let $\sqrt{\log n} \le |I_t| \le n/\log n$. Then there are $\tau_1, \tau_2 = \log_{1+q}(n/|I_t|) + o(\log n)$ such that $|I_{t+\tau_2}| < n/\log n < |I_{t+\tau_1}|$.*

(b) *Let $n/\log n \le |I_t| \le n - n/\log n$. Then there is $\tau = o(\log n)$ such that $|I_{t+\tau}| > n - n/\log n$.*

(c) *Let $|I_t| \ge n - n/\log n$.*

1. *Case $q = 1$: Then there is $\tau = o(\log n)$ such that $|I_{t+\tau}| = n$.*
2. *Case $q \ne 1$: Then there is $\tau \le -\log n/\log(1-q) + o(\log n)$ such that $|I_{t+\tau}| = n$.*

Note that for $q = 1$ this already implies Theorems 4b and 5a. This leaves the case for $q \neq 1$.

▶ **Lemma 20.** *Let $0 < \varepsilon \leq 1/2, q \in (0, 1]$ and $\mathcal{G} = (G_n)_{n\in\mathbb{N}}$ be an expander sequence. Let $\tilde{\mathcal{G}} = (\tilde{G}_n)_{n\in\mathbb{N}}$ be such that each $\tilde{G}_n$ is obtained by deleting edges of G_n such that each vertex keeps at least a $(1/2+\varepsilon)$ fraction of its edges and abbreviate $I_t = I_t^{(pull)}$. Let further $q \in (0, 1)$ and $|I_t| \leq n/2$. Then for $\tau = -\log n/ \log (1 - q)$ and all $c < 1$ whp $|I_{t+c\tau}| < n$.*

3.2 Proof of Theorem 7a – *push* informs almost all vertices fast in spite of edge deletions

To shorten the notation let us call the setting with deleted edges "new model" and the setting without "old model", that is, the term new model corresponds to the graphs in $\tilde{\mathcal{G}}$, while old model refers to the (original) graphs in $\mathcal{G}$. We prove Lemma 21 that directly implies Theorem a. We write $I_t = I_t^{(push)}$ throughout.

▶ **Lemma 21.** *Under the assumptions of Theorem 7a the following holds for the new model:*
(a) *There are $\tau, \tilde{\tau} = \log_{1+q}(n) + o(\log n)$ such that whp $|I_{\tilde{\tau}}| < n/ \log n < |I_\tau|$.*
(b) *Assume $|I_t| \geq n/ \log n$. Then there is a $\tau = o(\log n)$ such that whp $|I_{t+\tau}| \geq n - n/ \log n$.*

For the proof of Lemma 21 we will need the following statements, the first one taken from [25].

▶ **Lemma 22** (Proof of Lemma 2.5 in [25])**.** *Consider the old model. Assume $|I_t| < n/ \log n$ and $q = 1$. Then*

$$P_t\big[|I_{t+1}| = |I_t| + (1 - o(1))|I_t|\big] = 1 - o(1). \tag{2}$$

▶ **Lemma 23.** *Consider* push *on a sequence of graphs $(G_n)_{n\in\mathbb{N}}$, where G_n has n vertices. Assume that $|I_t| = \omega(1)$ and that (2) holds for $q = 1$, that is, assume that $P_t\big[|I_{t+1}| = |I_t| + (1 - o(1))|I_t|\big] = 1 - o(1)$ for $q = 1$. Then for $q \in (0, 1]$*

$$P_t\big[|I_{t+1}| = |I_t| + (q - o(1))|I_t|\big] = 1 - o(1). \tag{3}$$

Moreover, assume that whenever $|I_t| < n/ \log n$, for $q = 1$, (2) holds. Then there are $\tau, \tilde{\tau} = \log_{1+q}(n) + o(\log n)$ such that whp

$$|I_{\tilde{\tau}}| < n/ \log n < |I_\tau|. \tag{4}$$

3.3 Proof of Theorems 5b, 7b – *push&pull* informs almost all vertices fast in spite of edge deletions

Before we show the actual proof we will first present an informal argument that contains all relevant ideas and important observations. Let $\sqrt{\log n} \leq |I_t| \leq n/ \log n$ and assume $q = 1$. In Section 3.2 we proved that for *push* the informed vertices nearly double in every round for an arbitrary expander sequence with edge deletions and an otherwise arbitrary set I_t. For *pull* this is not true; however, we proved in Section 3.1 that the number of edges between the informed and the uninformed vertices nearly doubles in every round. The first attempt towards the proof of Theorems b, b then seems obvious: one would try to show that either the vertices triple every round, or the the edges do so, or for example that the product of the two quantities increases by a factor of 9. As it turns out, this is in general not the case; indeed, it is possible to choose an expander sequence, to delete edges such that each vertex keeps at least an $(1/2 + \varepsilon)$-fraction of its neighbors, and to choose a (large) set of informed vertices I_t such that after one round whp either $|I_{t+1}| < c|I_t|$ or $e(I_{t+1}, U_{t+1}) < ce(I_t, U_t)$ or

$|I_{t+1}|e(I_{t+1}, U_{t+1}) < c^2|I_t|e(I_t, U_t)$ for some $c < 3$. On the other hand and although we have no explicit description of these "malicious" sets, it seems rather unlikely that such sets will occur several times during the execution of *push&pull*.

In order to show the claimed running time of *push&pull* we will impose some additional structure. Let $\varepsilon > 0$. In the subsequent exposition we assume that our graph G – obtained from an expander by deleting edges such that each vertex keeps at least an $(1/2 + \varepsilon)$ fraction of the edges – has a *very special* structure. In particular, we assume that there is a partition $\Pi = (V_i)_{i\in[k]}$ of the vertex set of G into a bounded number k of equal parts such that $E_G(V_i) = \emptyset$ for all $1 \le i \le k$ and such that the induced subgraph (V_i, V_j) looks like a random regular bipartite graph for all $1 \le i < j \le k$. Of course, not every relevant G admits such a partition; however, Szemeredi's regularity lemma guarantees that every sufficiently large graph has a partition that is in a well-defined sense *almost* like the one described previously, and a substantial part of our proof is concerned with showing that being "almost special" does not hurt significantly.

Assuming that G is very special let us collect some easy facts. Denote the degree of $u \in V_i$ in the induced subgraph (V_i, V_j) with d_{ij}; this immediately gives that $d_G(u) = \sum_{\ell=1}^{k} d_{i\ell}$, and note that $d_{ii} = 0$ as there are no edges in V_i. Moreover, regular bipartite random graphs satisfy an expander property, that is, for all $W_i \subseteq V_i, W_j \subseteq V_j, 1 \le i < j \le k$ we have

$$e(W_i, W_j) = d_{i,j}|W_i||W_j|/|V_j| + o(d_{i,j})|W_i| \approx |W_i||W_j|d_{ij}k/n$$

where we used that all $|V_i|$'s are of equal size. This is quite similar to the property that we used in our preceding analysis on expander sequences, see Lemma 16. As a pair in Π behaves like a bipartite expander sequence we can easily compute the expected number of informed vertices. We do so now for *pull*. Let $\left|I_{t+1}^{i,j}\right|$ be the number of vertices in V_i informed after round $t+1$ by *pull* from vertices only in V_j and set $I_t^i := I_t \cap V_i, U_t^i := U_t \cap V_i \ \forall\ 1 \le i \le k$. Thus, as long as I_t^i is much smaller than V_i (and thus also $U_t^i \approx |V_i| = n/k$) we get

$$\mathbb{E}_t\left[\left|I_{t+1}^{(pull),i,j} \backslash I_t\right|\right] = \sum_{u\in U_t^i} \frac{|N(u) \cap I_t^j|}{d(u)} = \frac{e(U_t^i, I_t^j)}{\sum_{1\le\ell\le k} d_{i\ell}} \approx \frac{d_{ij}}{\sum_{1\le\ell\le k} d_{i\ell}}|I_t^j|.$$

A similar calculation, which we don't perform in detail, yields for *push*

$$\mathbb{E}_t\left[\left|I_{t+1}^{(push),i,j} \backslash I_t\right|\right] \approx \frac{d_{ij}}{\sum_{1\le\ell\le k} d_{\ell j}}|I_t^j|.$$

Moreover, as in previous proofs it turns out that the number of vertices informed simultaneously by *push* as well as *pull* is negligible. Thus we obtain that more or less

$$\mathbb{E}_t\left[\left|I_{t+1}^{(pp),i,j}\right|\right] \approx |I_t^i| + \left(\frac{d_{ij}}{\sum_{1\le\ell\le k} d_{i\ell}} + \frac{d_{ij}}{\sum_{1\le\ell\le k} d_{\ell j}}\right)|I_t^j|$$

and by linearity of expectation

$$\mathbb{E}_t\left[\left|I_{t+1}^{(pp),i}\right|\right] \approx |I_t^i| + \sum_{1\le j\le k}\left(\frac{d_{ij}}{\sum_{1\le\ell\le k} d_{i\ell}} + \frac{d_{ij}}{\sum_{1\le\ell\le k} d_{\ell j}}\right)|I_t^j|.$$

Set $X_t = (|I_t^i|)_{i\in[k]}$ and $A = (A_{ij})_{1\le i,j\le k}$, the matrix with entries

$$A_{ij} = \frac{d_{ij}}{\sum_{1\le\ell\le k} d_{i\ell}} + \frac{d_{ij}}{\sum_{1\le\ell\le k} d_{\ell j}} \qquad \text{for } 1 \le i \ne j \le k$$

and $A_{ii} = 1$ for $1 \leq i \leq k$. With this notation we obtain the recursive relation

$$\mathbb{E}_t[X_{t+1}] \approx A \cdot X_t, \tag{5}$$

that is, we may expect that $X_t \approx \mathbb{E}_t[X_t] \approx A^t X_0$. If we then denote by $\lambda_{\max}$ the greatest eigenvalue of A, then we obtain that in leading order

$$|I_t| \approx \lambda_{\max}^t.$$

Our aim is to show that *push&pull* is (at least) as fast as on the complete graph, that is, $|I_t| \precsim 3^t$, and so we take a closer look at the eigenvalues of A. By construction A is symmetric, so that the largest eigenvalue equals $\sup_{\|x\|=1} \|x^T Ax\|$, and the simple choice $x = k^{-1/2}\mathbf{1}$ yields

$$\begin{aligned}\lambda_{\max} &\geq \frac{\sum_{(i,j)} A_{i,j}}{k}\\ &= \frac{\sum_{j=1}^k 1 + \sum_{i=1}^k \sum_{j=1}^k d_{ij}/\left(\sum_{\ell=1}^k d_{i\ell}\right) + \sum_{j=1}^k \sum_{i=1}^k d_{ij}/\left(\sum_{\ell=1}^k d_{\ell j}\right)}{k} = 3.\end{aligned}$$

This neat property leads us to the expected result $T_{pp}(G) = (1 + o(1)) \log_{\lambda_{\max}} n \leq (1 + o(1)) \log_3 n$, and it also completes the informal argument that justifies the claim made in Theorems 5b and 7b. In the unabridged version we turn this argument step by step into a formal proof by filling in all missing pieces.

3.4 Proof of Theorem 6b – edge deletions may slow down *push&pull*

For any $0 < \varepsilon < 1/2$, $q \in (0, 1)$ we consider a sequence of graphs $(G_n(\varepsilon))_{n\in\mathbb{N}} = ((V_n, E_n))_{n\in\mathbb{N}}$. Let $V_n = A_n \cup B_n$ with $A_n := \{1, \dots, \lfloor n/2\rfloor\}, B_n := \{\lfloor n/2\rfloor + 1, \dots, n\}$ and $\deg(v) = n - 1$ for all $v \in A_n$. Let the induced subgraph of B_n be a random graph in which each edge is included independently with probability $p = 1 - 2\varepsilon$. We know and it is easy to show, see for example [15, Section IV], that whp this subgraph is almost regular, i.e.,

$$d_{B_n}(v) = (1 + o(1))(1 - 2\varepsilon)n/2 \quad \text{for all } v \in B_n, \tag{6}$$

and is an expander, which means that for every $S_n \subseteq B_n$, $1 \leq |S_n| \leq n/4$ and $d_{B_n} := (1 - 2\varepsilon)n/2$ we have

$$e(S_n, B_n \backslash S_n) = (1 + o(1))\frac{d_{B_n}|S_n||B_n \setminus S_n|}{|B_n|} = (1 - 2\varepsilon + o(1))|S_n||B_n \setminus S_n|. \tag{7}$$

At first we give a statement that describes the expected number of informed vertices after performing one round of *push&pull*.

▶ **Lemma 24.** *Let $G_n(\varepsilon) = (A_n \cup B_n, E_n)$ be as above.*

(a) *Let $\sqrt{\log n} \leq |I_t| \leq n/\log n$ and set*

$$X_t = \left(\left|I_t^{(pp),(A)}\right|, \left|I_t^{(pp),(B)}\right|\right) := \left(\left|I_t^{(pp)} \cap A_n\right|, \left|I_t^{(pp)} \cap B_n\right|\right).$$

Then $\mathbb{E}_t[X_{t+1}] = (1 + o(1))MX_t$, where

$$M = \begin{pmatrix} 1 + q & q\big(1 + \varepsilon/(2 - 2\varepsilon)\big) \\ q\big(1 + \varepsilon/(2 - 2\varepsilon)\big) & 1 + q\big(1 - 2\varepsilon/(2 - 2\varepsilon)\big) \end{pmatrix}.$$

(b) *Let $|U_t^{(pp)}| \leq n/\log n$. Then $\mathbb{E}_t[|U_{t+1}^{(pp)}|] \leq (1 + o(1))e^{-q(1/2+(1/2-\varepsilon)/(1-\varepsilon))}\,(1 - q)\,|U_t|$.*

▶ Remark 25. Let $\lambda_{\max}$ be the greatest eigenvalue of M as defined in Lemma 24a. Then

$$\lambda_{max} = 1 + 2q + (2q(\sqrt{(\varepsilon^2/2 - \varepsilon + 1)} - 1) + q\varepsilon)/(2 - 2\varepsilon) > 1 + 2q.$$

Next comes a lemma that bounds the runtime of *push&pull* on $G_n(\varepsilon)$. In particular, Lemma 26 $a)$ and $c)$ provide a lower bound on the runtime and Lemma 26 $a), b)$ and $d)$ provides an upper bound.

▶ **Lemma 26.** *Let $I_t = I_t^{(pp)}$, $\varepsilon > 0$ and $\lambda = \lambda_{\max}(M)$ be the greatest eigenvalue of M as given in Lemma 24a. Consider $G_n(\varepsilon)$.*

(a) *Let $\sqrt{\log n} \le |I_t| \le n/\log n$. Then there are $\tau_1, \tau_2 = \log_\lambda(n/|I_t|) + o(\log n)$ such that $|I_{t+\tau_1}| < n/\log n < |I_{t+\tau_1}|$.*

(b) *Let $n/\log n \le |I_t| \le n - n/\log n$. Then there is $\tau = o(\log n)$ such that $|I_{t+\tau}| > n - n/\log n$.*

(c) *Let $|I_t| \le n/\log n$. Then there is $\tau \ge \log n/\log((1-q)^{-1}\exp(q(1/2 + (1/2-\varepsilon)/(1-\varepsilon)))) - o(\log n)$ such that $|I_{t+\tau}| < n$.*

(d) *Let $|I_t| \ge n - n/\log n$ and $q \in (0,1)$. Then there is $\tau \le \log n/\log((1-q)^{-1}\exp(q(1/2 + (1/2-\varepsilon)/(1-\varepsilon)))) + o(\log n)$ such that $|I_{t+\tau}| = n$.*

Lemma 26 gives that

$$T_{pp}(G_n(\varepsilon), q) = \log_\lambda n + \frac{1}{q(1-1.5\varepsilon)/(1-\varepsilon) - \log(1-q)} \log n + o(\log n)$$

where $\lambda = 1 + 2q + (2q(\sqrt{(\varepsilon^2/2 - \varepsilon + 1)} - 1) + q\varepsilon)/(2 - 2\varepsilon) > 1 + 2q$. To see whether *push&pull* actually slowed down (in terms of order $\log n$) one has to compare the runtime on this sequence of graphs to $c_{pp}(q)\log n$; the runtime on expander sequences. In the Figure 1 we can see that it slows down for nearly all values of ε and q in question; however, there are admissible values of ε and q such that the process even speeds up.

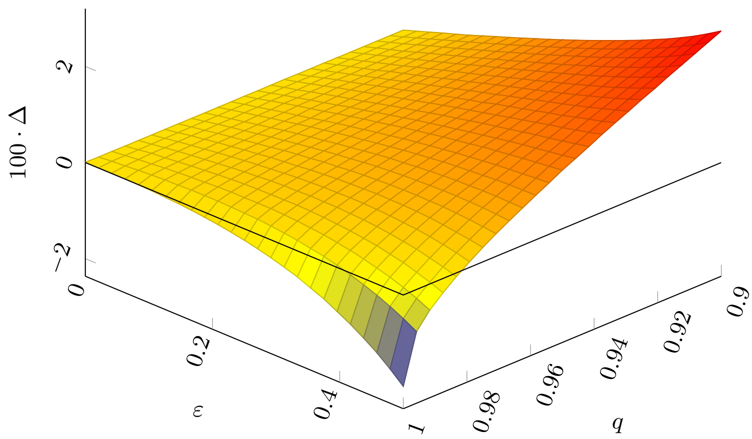

Figure 1 Plotted values of Δ in $T_{pp}(G_n(\varepsilon), q) - c_{pp}\log n = \Delta \log n + o(\log n)$, for $0.9 < q < 1$ and $0 < \varepsilon < 1/2$.

References

1 Hüseyin Acan, Andrea Collevecchio, Abbas Mehrabian, and Nick Wormald. On the push&pull protocol for rumour spreading. In *Extended Abstracts Summer 2015*, pages 3–10. Springer, 2017.

2 Omer Angel, Abbas Mehrabian, and Yuval Peres. The string of diamonds is tight for rumor spreading. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

3 Stéphane Boucheron, Gábor Lugosi, and Olivier Bousquet. Concentration inequalities. In *Advanced Lectures on Machine Learning*, pages 208–240. Springer, 2004.

4 Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE/ACM Transactions on Networking (TON)*, 14(SI):2508–2530, 2006.

5 Keren Censor-Hillel, Bernhard Haeupler, Jonathan Kelner, and Petar Maymounkov. Global computation in a poorly connected world: fast rumor spreading with no dependence on conductance. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 961–970. ACM, 2012.

6 Flavio Chierichetti, George Giakkoupis, Silvio Lattanzi, and Alessandro Panconesi. Rumor Spreading and Conductance. *Journal of the ACM (JACM)*, 65(4):17, 2018.

7 Sebastian Daum, Fabian Kuhn, and Yannic Maus. Rumor Spreading with Bounded In-Degree. In *Structural Information and Communication Complexity - 23rd International Colloquium, SIROCCO 2016, Helsinki, Finland, July 19-21, 2016, Revised Selected Papers*, pages 323–339, 2016. doi:10.1007/978-3-319-48314-6_21.

8 Domingos Dellamonica, Yoshiharu Kohayakawa, Martin Marciniszyn, and Angelika Steger. On the Resilience of Long Cycles in Random Graphs. *Electr. J. Comb.*, 15(1), 2008. URL: http://www.combinatorics.org/Volume_15/Abstracts/v15i1r32.html.

9 Alan Demers, Dan Greene, Carl Houser, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dan Swinehart, and Doug Terry. Epidemic algorithms for replicated database maintenance. *ACM SIGOPS Operating Systems Review*, 22(1):8–32, 1988.

10 Benjamin Doerr, Mahmoud Fouz, and Tobias Friedrich. Social networks spread rumors in sublogarithmic time. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 21–30. ACM, 2011.

11 Benjamin Doerr and Anatolii Kostrygin. Randomized Rumor Spreading Revisited. In *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, pages 138:1–138:14, 2017. doi:10.4230/LIPIcs.ICALP.2017.138.

12 Benjamin Doerr and Marvin Künnemann. Tight analysis of randomized rumor spreading in complete graphs. In *Proceedings of the Meeting on Analytic Algorithmics and Combinatorics*, pages 82–91. Society for Industrial and Applied Mathematics, 2014.

13 Robert Elsässer and Thomas Sauerwald. On the runtime and robustness of randomized broadcasting. *Theoretical Computer Science*, 410(36):3414–3427, 2009.

14 Uriel Feige, David Peleg, Prabhakar Raghavan, and Eli Upfal. Randomized broadcast in networks. *Random Structures & Algorithms*, 1(4):447–460, 1990.

15 Nikolaos Fountoulakis, Anna Huber, and Konstantinos Panagiotou. Reliable broadcasting in random networks and the effect of density. In *2010 Proceedings IEEE INFOCOM*, pages 1–9. IEEE, 2010.

16 Nikolaos Fountoulakis and Konstantinos Panagiotou. Rumor spreading on random regular graphs and expanders. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 560–573. Springer, 2010.

17 Nikolaos Fountoulakis, Konstantinos Panagiotou, and Thomas Sauerwald. Ultra-fast rumor spreading in social networks. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 1642–1660. SIAM, 2012.

18 Tobias Friedrich, Thomas Sauerwald, and Alexandre Stauffer. Diameter and Broadcast Time of Random Geometric Graphs in Arbitrary Dimensions. *Algorithmica*, 67(1):65–88, September 2013. doi:10.1007/s00453-012-9710-y.

19 Alan M Frieze and Geoffrey R Grimmett. The shortest-path problem for graphs with random arc-lengths. *Discrete Applied Mathematics*, 10(1):57–77, 1985.

20 George Giakkoupis. Tight bounds for rumor spreading in graphs of a given conductance. In *28th International Symposium on Theoretical Aspects of Computer Science, STACS 2011, March 10-12, 2011, Dortmund, Germany*, pages 57–68, 2011. doi:10.4230/LIPIcs.STACS.2011.57.

21 George Giakkoupis. Tight Bounds for Rumor Spreading with Vertex Expansion. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 801–815, 2014. doi:10.1137/1.9781611973402.59.

22 Bernhard Haeupler. Simple, fast and deterministic gossip and rumor spreading. *Journal of the ACM (JACM)*, 62(6):47, 2015.

23 Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006.

24 Richard M. Karp, Christian Schindelhauer, Scott Shenker, and Berthold Vöcking. Randomized Rumor Spreading. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pages 565–574, 2000. doi:10.1109/SFCS.2000.892324.

25 Konstantinos Panagiotou, Xavier Pérez-Giménez, Thomas Sauerwald, and He Sun. Randomized Rumour Spreading: The Effect of the Network Topology. *Combinatorics, Probability & Computing*, 24(2):457–479, 2015. doi:10.1017/S0963548314000194.

26 Konstantinos Panagiotou and Leo Speidel. Asynchronous rumor spreading on random graphs. *Algorithmica*, 78(3):968–989, 2017.

27 Vojtěch Rödl and Mathias Schacht. Regularity lemmas for graphs. In *Fete of combinatorics and computer science*, pages 287–325. Springer, 2010.

28 Benny Sudakov and Van H. Vu. Local resilience of graphs. *Random Struct. Algorithms*, 33(4):409–433, 2008. doi:10.1002/rsa.20235.