Some Notes on Randomized Distributed Protocols

Andrea Clementi
Universitá di Roma Tor Vergata
Rome, Italy
clementi@mat.uniroma2.it

1 Basic Concepts

- Definition of a Randomized Algorithm: Probability Space, Faulty Executions, Error Probability, Expected Complexity
- Notion of Good Randomized Algorithms

A probabilistic algorithm $\mathcal{A}$ is an algorithms that has access to a source S of random bits and it can takes decisions according to the outcome of the random bits it asks to S. Given a fixed input x of size n, consider the execution $\mathcal{A}(x)$ and let $r := r(x)$ be the total number of random bits $\mathcal{A}$ asks to S during the execution $\mathcal{A}(x)$.

Then, we get that the outcome of $\mathcal{A}(x)$ (correct/failure), its convergence time, and any other "dynamic" parameter, are *random variables* over the probability space

$$\Omega \;=\; (\{0,1\}^r, \mathbb{P}(\cdot)), \text{ where } \mathbb{P}(\cdot) \text{ is the Probability Distribution induced by } \mathcal{A}(x)\,.$$

For instance, if $\mathcal{A}$ chooses the r bits *independently and uniformly at random*[1], then $\mathbb{P}(\mathbf{y}) = 1/2^r$, for every binary sequence $\mathbf{y} \in \{0,1\}^r$.

For the sake of simplicity, assume $\mathcal{A}$ be an algorithm for a YES/NO (decision) problem $\Pi(x)$. We can introduce the crucial concept of error probability in terms of specific events in Ω and a notion of good randomized algorithms.

Definition 1.1. *We say that Algorithm $\mathcal{A}$ for problem Π has* error probability ϵ*, for some real $\epsilon \in [0,1]$, if,* for any input x*, it holds that*

$$\mathbb{P}_\Omega(\mathcal{A}(x) \;=\; \Pi(x)) \;\geq\; 1-\epsilon\,.$$

Moreover, we say that $\mathcal{A}$ solves Π WITH HIGH PROBABILITY *if, for sufficiently large n, for any input x of size n, it holds that*

$$\mathbb{P}_\Omega(\mathcal{A}(x) \;=\; \Pi(x)) \;\geq\; 1-\frac{1}{n^\beta}\,,\ \textit{for some constant } \beta > 0\,.$$

Observe that notation $\mathbb{P}_\Omega(\cdot)$ says that the probability of the event is that defined by the Probability Space Ω induced by Algorithm $\mathcal{A}$. In the sequel, we will omit this subscript when it is clear from context.

[1]Notice that t his means that the source S let each used bit y_i be such that $\mathbb{P}(y_i = 1) = 1/2$.

1.1 Randomized Protocols

We can easily transfer all the concepts and definitions given above for centralized algorithms to any *Distributed Computational Model* as follows. A fixed Protocol (i.e. distributed algorithm) $\mathcal{A}$, over a fixed graph $G(V, E)$ with $n = |V|$ computing nodes and starting from a given initial configuration x, specifies all actions of every node. Each node v has access to a *private*, **independent** source $s(v)$ of random bits. The decisions taken by each node v according to $\mathcal{A}$, depends (also) on the outcomes of the random sequence of the random bits flipped by v via $s(v)$.

Let us assume that the *maximum* number of random bits chosen by any node during the execution of $\mathcal{A}$ on G, with starting configuration x, is $r \geq 0$. Then, it turns out that the outcome of Protocol $\mathcal{A}(G, x)$ (as well as its convergence time or its message complexity) is a random variable which is fully determined by the probabilty space

$$\Omega \ = \ (\{0,1\}^{n \cdot r}, \mathbb{P}(\cdot)), \text{ where } \mathbb{P}(\cdot) \text{ is the Probability Distribution induced by } \mathcal{A}(G, x)\,.$$

Then, the concepts "*error probability*" and "*with high probability*" can then be easily extended to the framework of distributed algorithms/protocols.

2 Warming-Up: Leader Election in a Unlabeled Ring

In the previous lecturers we have seen that the *Leader Election* problem (for short LEP) is not deterministically solvable if nodes have no Unique IDs. This strong negative result can be verified even over a ring of 3 nodes (i.e. a triangle).

Now, let us consider a distributed system formed by a ring $G(V, E)$ of size n, where nodes are fully anonymous (i.e. they don't have any global ID's). Each nodes knows the size of the ring n and, clearly, the fact that it is on a ring. Let us take the standard restrictions about faults and sense of direction.

The main algorithmic question is: Can we design a good and efficient *randomized* protocol for LEP in the framework above ?

The answer is yes and the only use of local randomness is in the initial phase where every node in V chooses a label i independently and uniformly at random from an alphabet $[m] = \{1, \ldots m\}$ of sufficiently large size and the hope for the global process is that there is no pair of nodes that get the same label. Then, after this labeling phase, the Protocol works exactly as in the deterministic setting (e.g. take the simplest deterministic protocol we have seen in the previous lectures). We formal describe the protocol over a synchronous discrete-time communication model, however, it is a simple exercise to adapt the protocol (and its analysis) in the asynchronous model.

- Randomized Protocol $\mathcal{RL}(G, n; m)$. $*$ (the right choice of parameter m will be given later).
- Phase 0. Wake-Up. $*$ (all nodes will be activated)
- Phase 1. Every node v, chooses (independently) and uniformly at random an integer $j_v \in [m]$;
- Phase 2. Every node runs a fixed deterministic protocol for LEP assuming Node Labeling $\{j_v : v \in V\}$.

2.1 Protocol Analysis

Let us analyze the correctness of Protocol $\mathcal{RL}(G, n; m)$. For the first time in this course, we cannot prove the protocol is always correct, i.e. it *always* converges to the a valid final state of LEP. Indeed, one can easily verify that landing in the possible scenario where every node, after Phase 2, chooses the same label m, the protocol fails! Clearly, this event is extremely unlike but it is possible!

Our first step is to derive a specific condition, i.e. event, that is sufficient to claim that, assuming that event take place, the Protocol works correctly. In our case, we observe that a sufficient (not necessary) condition is the following event

$$\mathcal{B} \doteq \text{“there is no pair of different nodes } v, w \in V, \text{ with } v \neq w \text{ such that } j_v = j_w \text{ ”}$$

Indeed, as already remarked, if all labels j_v's are mutually different, then the protocol, in Phase 3, works as a standard deterministic Leader Election protocol under the assumption that all labels are different!

We now bound the probability of Event $\mathcal{B}$ in the probability space yielded by Protocol $\mathcal{RL}$ on input $(G, n; m)$. We can see Phase 2 of the protocol as a classic *Balls-into-Bins process* (see Book [3]), where there are n balls (i.e. the random choices j_v's) which are thrown independently and uniformly at random into m bins (the possible values taken by the labels). The question we are interested on is how large is the probability that any two different balls turn out to fall inside the same bin. To this aim, let us consider a fixed bin $i \in [m]$. Then the probability of event

$$\mathcal{B}_i \doteq \text{“at least two balls go to bin } i\text{”}$$

is

$$\begin{aligned}
\text{union of disjoint events:} \quad & \sum_{k=2}^{n} \binom{n}{k} \left(\frac{1}{m}\right)^k \left(1 - \frac{1}{m}\right)^{n-k} \\
& \leq \sum_{k=2}^{n} \binom{n}{k} \left(\frac{1}{m}\right)^k \\
\text{Stirling's Apx} \quad & \leq \sum_{k=2}^{n} \left(\frac{en}{k} \cdot \frac{1}{m}\right)^k \\
& \leq O\left(\frac{n^2}{m^2}\right) + \sum_{k=3}^{n} \left(\frac{en}{k} \cdot \frac{1}{m}\right)^k \\
& \leq O\left(\frac{n^2}{m^2} + \frac{n^4}{m^3}\right) \qquad (1)
\end{aligned}$$

Now, observe that the bound above refers to a fixed (arbitrary) bin i, while the protocol gets in trouble whenever at least one bin gets event $\mathcal{B}_i$, so we need a Union Bound here:

$$\begin{aligned}
\mathbb{P}(\mathcal{B}) = \quad & \mathbb{P}\left(\bigcup_i \mathcal{B}_i\right) \leq \sum_{i=1}^{m} \mathbb{P}\left(\mathcal{B}_i\right) \\
\text{from Eq. (1)} \quad \leq \quad & m \cdot O\left(\frac{n^2}{m^2} + \frac{n^4}{m^3}\right)
\end{aligned}$$

We can thus fix the parameter m in Protocol $\mathcal{RL}(G, n; m)$ as any integer larger than n^3 and get that $\mathbb{P}(\mathcal{B}) = O(1/n)$. We have thus proved that Protocol $\mathcal{RL}(G, n; n^3)$ makes a correct Leader Election on a ring of size n, *with high proabilty.*

3 Majority Consensus via the 3-Majority Dynamics

Let $G = ([n], E)$ be a graph, let C be a finite set of *colors* and let $\mathbf{x} : [n] \to C$ be an initial *coloring* of the nodes of G. If the number of colors is $|C| = h$ we will call $\mathbf{x}$ an *h-coloring.*

The algorithmic goal here is to design an efficient and simple protocol for *Majority Consensus.*

In this task, it is assumed that the initial coloring has some *bias* s towards some *Plurality Color* and the goal is to let the system converge to the monochromatic configuration where all nodes get the plurality color.

Let us clarify here the notion of bias of a configuration. Given any configuration $\mathbf{x}$, for any color $j \in C$, define $x(j)$ as the color-j size, i.e., the number of nodes supporting color j in $\mathbf{x}$. Then, let us order the color sizes $x(1), x(2), \ldots, x(h)$ according to the non-increasing ordering. The bias of $\mathbf{x}$ is defined as $s(\mathbf{x}) \doteq x(1) - x(2)$. We will use the notation s whenever the configuration $\mathbf{x}$ is clear from the context.

Definition 3.1. *Given a distributed system $G = ([n], E)$, the ℓ-Majority-Consenus Problem is defined by the following property. Starting from any coloring $\mathbf{x} : [n] \to C$ having bias $s(\mathbf{x}) \geq \ell$, the system must converge to a stable state where all nodes support the majority color.*

Consider the following family of synchronous protocols

Definition 3.2. *The k-*Majority Protocol *(for short k-*MAJ*) works as follows*

- *At every step,*
  each node independently picks k neighbors (including itself and with repetition) u.a.r. and recolors itself according to the majority of the colors it sees (ties are broken arbitrarily).

Exercise 3.1. *1) Consider the 1-MAJ on the complete graph K_n for the binary coloring, i.e., the case $h = 2$. Show that if, at a given step $t \geq 0$, the system takes any fixed coloring configuration $\mathbf{x}$ with an arbitrary bias $s = s(\mathbf{x}) \geq 0$ then the* expected *bias at the next step is s as well. Formally, show that*

$$\mathbf{E}\left[s(X_{t+1}) \,|\, s(x_t) = s\right] = s$$

2) Now, consider the 2-MAJ on the complete graph K_n for the binary coloring, i.e., the case $h = 2$. Prove a similar expected *behaviour to that of 1-MAJ.*

The solutions of exercise above show that *k-MAJ* yields no *drift* towards the majority color, no matter what is the current bias the system has: the bias does not increase in average! This fact has two main consequences. First (show this as a further excercise), it holds that

$$\mathbf{P}\left(\text{the system converges to the majority color}\right) \;=\; \frac{x_0(1)}{n}, \quad \text{where } x_0(1) \text{ is majority size at } t = 0.$$

This implies that, even starting from a bias $s = \Omega(n)$, the error probability of the two dynamics above is still larger than an absolute constant. Second bad news is the fact that the above dynamics are very slow to converge: they take a polynomial number of steps! This fact is rather hard to prove but it is essentially due to the fact that, as remarked above, the dynamics has no expected drift and its convergence is only due to the (unpredictable) randomness of the process.

For the above facts, we will focus on *3-MAJ.*

3.1 Unbalanced 2-coloring with 3-majority

We analyze k-MAJ in the case of 2-colorings.

Definition 3.3. *For a 2-coloring* $\mathbf{x} : [n] \to \{\texttt{red}, \texttt{blue}\}$*, we say that* $\mathbf{x}$ *is* ω-unbalanced *if its bias is s.t.* $s(\mathbf{x}) \geq \omega$.

In the next lemma we show that, if the initial configuration is sufficiently unbalanced, then 3-MAJ solves Majority Consensus within $O(\log n)$ rounds, w.h.p.

Lemma 3.4. *If* $G \equiv K_n$ *and the starting 2-coloring is* $\Omega(\sqrt{n \log n})$*-unbalanced, 3-MAJ converges to the majority color after* $O(\log n)$ *time steps, w.h.p.*

Proof. Let X_t be the random variable counting the number of `red` nodes at time t. For every node i let Y_i the indicator random variable of the event "node i is `red` at the next step". For every $a = 0, 1, \ldots, n$ it holds that

$$\mathbf{P}\left(Y_i = 1 \,|\, X_t = a\right) = \left(\frac{a}{n}\right)^3 + 3\frac{a^2(n-a)}{n^3} = \frac{a^2}{n^3}(3n - 2a)$$

Hence, the expected number of `red` nodes at the next time step is

$$\mathbf{E}\left[X_{t+1} \,|\, X_t = a\right] = \left(\frac{a}{n}\right)^2 (3n - 2a) \tag{2}$$

Wlog, we assume `red` is the minority color, i.e.

$$s = s(\mathbf{x}) = x(\text{blue}) - x(\text{red}) = b - a \geq c\sqrt{n \log n}\,,$$

and then we split the analysis in three phases according to the range the minority a falls in.

Phase 1: a lies in the range from $n/2 - \Theta\left(\sqrt{n \log n}\right)$ to $n/4$:

Suppose that the number of `red` nodes is $X_t = a$ for some $a \leqslant n/2 - s$ where $c\sqrt{n \log n} \leqslant s \leqslant n/4$ for some positive constant c. Now we show that $X_{t+1} \leqslant n/2 - (9/8)s$, w.h.p.

Observe that function

$$f(a) \;= a^2(3n - 2a) \;\text{ in Eq. (2)}$$

is *increasing* for every $0 < a < n$. Hence, for $a \leqslant n/2 - s$ we have that

$$\begin{aligned} \mathbf{E}\left[X_{t+1} \,|\, X_t = a\right] &= \left(\frac{a}{n}\right)^2 (3n - 2a) \leqslant \left(\frac{n}{2} - s\right)^2 (3n - 2(n/2 - s)) \\ &= \frac{n}{2} - \frac{3}{2} \cdot s + 2 \cdot \frac{s^3}{n^2} \leqslant \frac{n}{2} - \frac{5}{4} \cdot s \end{aligned}$$

where the last inequality holds because $s \leqslant n/4$.

Notice that random variables Y_i's are independent conditional on X_t. We can thus apply the Chernoff bound (Additive Form, i.e. Eq. 18) with

$$\mu \;= \frac{n}{2} - \frac{5}{4} \cdot s \;\text{ and }\; \lambda = \frac{1}{8} \cdot s$$

we then get, for every $a \leqslant s \leqslant n/4$,

$$\mathbf{P}\left(X_{t+1} \geqslant \left(\frac{n}{2} - \frac{5}{4}\cdot s\right) + \frac{1}{8}\cdot s \,\middle|\, X_t = a\right) = \tag{3}$$

$$\mathbf{P}\left(X_{t+1} \geqslant \frac{n}{2} - \frac{9}{8}\cdot s \,\middle|\, X_t = a\right) \leqslant e^{-s^2/(64n)} \tag{4}$$

If $s \geqslant c\sqrt{n\log n}$ for a sufficiently large constant $c > 0$, then the last term above is

$$e^{-s^2/(64n)} \;\leq\; \frac{1}{n^{\Theta(1)}}\,.$$

so we get that $X_{t+1} \leqslant (n/2) - (9/8)s$ w.h.p. Thus, when $c\sqrt{n\log n} \leqslant s \leqslant n/4$ the *unbalance* of the coloring increases exponentially w.h.p.

Let us name $\mathcal{E}_t$ the event

$$\mathcal{E}_t = \text{“}X_t \leqslant \max\left\{\frac{n}{4},\ \frac{n}{2} - (9/8)^t\right\}\text{”}$$

Observe that from (4) it follows that, for every $t \in \mathbb{N}$, we have

$$\mathbf{P}\left(\mathcal{E}_{t+1} \,\middle|\, \bigcap_{i=1}^{t}\mathcal{E}_i\right) \geqslant 1 - n^{-\alpha}$$

Thus, for $T = \frac{\log(n/4)}{\log(9/8)} = \mathcal{O}(\log n)$ the probability that the number of `red` nodes has gone below $n/4$ within the first T time steps is

$$\begin{aligned}\mathbf{P}\left(\exists t \in [0,T] \,:\, X_t \leqslant n/4\right) \;&\geqslant\; \mathbf{P}\left(\bigcap_{t=1}^{T}\mathcal{E}_t\right) \geqslant \prod_{t=1}^{T}\mathbf{P}\left(\mathcal{E}_t \,\middle|\, \bigcap_{i=1}^{t-1}\mathcal{E}_i\right)\\ &\geqslant\; (1-n^{-\alpha})^T \geqslant 1 - 2Tn^{-\alpha} \geqslant 1 - n^{-\alpha/2}\end{aligned}$$

Phase 2: a lies in the range from $n/4$ to $\mathcal{O}(\log n)$: If $X_t = a$ with $a \leqslant (1/4)n$, from (2) we get

$$\begin{aligned}\mathbf{E}\left[X_{t+1} \mid X_t = a\right] = \left(\frac{a}{n}\right)^2(3n-2a) \leqslant\\ a\cdot\left(\frac{n/4}{n^2}\right)(3n) \leq \frac{3}{4}a\end{aligned}$$

We can apply Chernoff bound (Multiplicative form) Eq. 13 with

$$\mu \;= \frac{3}{4}a \;\text{ and }\; \delta = \frac{1}{20}\,.$$

and, we can fix a a suitable positive constant β, such that

$$\mathbf{P}\left(X_{t+1} \geqslant \frac{4}{5}a \,\middle|\, X_t = a\right) \leqslant e^{-\beta a}$$

Hence as long as $a = \Omega(\log n)$ then the number of `red` nodes decreases exponentially w.h.p. By reasoning as in the previous phase we get that after further $\mathcal{O}(\log n)$ time steps the number of `red` nodes is $\mathcal{O}(\log n)$.

Phase 3: a lies in the range from $\mathcal{O}(\log n)$ to zero: Observe that for $a = \mathcal{O}(\log n)$, in Eq. (2) we have that

$$\mathbf{E}\left[X_{t+1} \,|\, X_t = a\right] \leqslant c/n$$

for a suitable positive constant c. Hence, by using Markov inequality (see Eq. 12) with

$$t \;= 1 \;\text{ and }\; \mu \;=\; c/n\,,$$

we get

$$\mathbf{P}\left(X_{t+1} \geqslant 1 \,|\, X_t = a\right) \leqslant c/n$$

and since X_{t+1} is integer valued it follows that all nodes are `blue` w.h.p. □

Exercise 3.2. $**$ *In the previous lemma we showed that, if 3-MAJ starts from a 2-coloring that is sufficiently unbalanced then after $O(\log n)$ time steps the graph is monochromatic. A natural question is whether the lemma still holds if we use a suitable "lazy" version of 2-MAJ. For the 2-MAJ protocol over a 2-coloring we need to specify a way of breaking ties. A natural way for that is the* inertial *way:* In case of ties keep your current color. *Observe that this updating rule depends on 3 values: the two sampled ones plus that supported by the node.*
Show that if each node runs this "lazy" version of 2-MAJ then, if we start from a $\Theta(\sqrt{n \log n})$-unbalanced 2-coloring, after $O(\log n)$ time steps all nodes have the same color, w.h.p.

4 Distributed Construction of Sparse Expanders

The construction of scalable, sparse graphs having good connectivity properties is a crucial issue in Network Design. We here focus on simple and efficient distributed protocols for this task which have strong applications in Peer-To-Peer Netwoks and Opportunistic Networks.

For constant positive parameter d (the parameter d is an integer), we are interested in the following synchronous protocol that is run on a the complete graph K_n over the set V of n nodes. The randomized protocol RTA (*Request-Then-Accept*) works as follows :

- RTA(n, d)
- **Sending Action:** Each node v picks d random other notes $w_1, \ldots, w_d$ u.i.r, and sends each of them a request of establishing edge (v, w_i) and it stores edge (v, w_i) in its neighborhood list N_v
- **Receiving Action:** Each node v receiving a link-request from node w, accept them and stores link (v, w) in N_v.
- The output graph is $G(V, E)$ s.t. $E = \cup_v N_v$.

Notice that G is in general a multigraph with self-loop. We can erase multiple edges without any relevant consequences on the results described in this note.

In the next subsections we derive some properties of the random graph G such as small maximum node-degree and good expansion.

4.1 Average and Maximum Degree of G

We label the nodes in an arbitrary fixed ordering s.t. we let $V = \{1, 2, \ldots, n\} = [n]$. By construction, the degree $d(v)$ of a vertex v is given by

$$\Delta(v) = d + \Delta_v^{\text{in}},$$

where Δ_v^{in} is the random variable (r.v.) counting the number of link requests received (and accepted) by v. We have the following simple

Fact 4.1. *Given any node v, it holds that $\mathbb{E}(\Delta_v^{\text{in}}) = d$ and, thus, $\mathbb{E}(\Delta(v)) = 2d$.*

Proof. Observe that the protocol produces dn total link-requests. Let us give them any fixed ordering $\{1, 2, \ldots, dn\} = [dn]$. Consider any fixed $v \in [n]$ and define, for any $j \in [dn]$ the binary r.v. X_j^v such that

$$X_j^v = 1 \text{ iff link-request } j \text{ has sent to } v.$$

It is then easy to prove that

$$\Delta_v^{\text{in}} = \sum_{j \in [dn]} X_j^v. \tag{5}$$

Moreover, since each request has been flipped i.u.r., it easily holds that

$$\mathbb{P}(X_j^v = 1) = \frac{1}{n}.$$

So,

$$\mathbb{E}(\Delta_v^{\text{in}}) = \mathbb{E}\left(\sum_{j \in [dn]} X_j^v\right) = \sum_{j \in [dn]} \mathbb{E}\left(X_j^v\right) = \sum_{j \in [dn]} \mathbb{P}(X_j^v = 1) = \frac{dn}{n} = d. \tag{6}$$

□

The above fact is not surprisingly at all! Indeed, the total number of link-requests is (deterministically) equal to dn, so $|E| = dn$ and each link request is selected randomly, so it is clear that, *in average*, each node must accept d link-requests. However, the bound $O(d)$ holds only *in average* and says not too much about what is the *maximum* (in-)degree of any node. We thus need stronger, concentration arguments to derive a bound on the maximum degree of a node which holds *with high probability*.

We prove the following bound.

Theorem 4.2. *The graph $G(V, E)$ constructed by* RTA*(n, d) has maximum degree $\Theta(\log n / \log\log n)$, w.h.p.*

Proof. From Fact 4.1, we already know that, for any fixed node $v \in [n]$, it holds that

$$\mathbb{E}(\Delta(v) = d + \Delta_v^{\text{in}}) = 2d.$$

We can thus focus only on the distribution of the r.v. Δ_v^{in}. As in the proof of Fact 4.1, we can write again

$$\Delta_v^{\text{in}} = \sum_{j\in[dn]} X_j^v\,, \tag{7}$$

where X_j^v are binary r.v.s which are mutually independent, uniformly distributed, and their expected sum is $\mu = d$. Hence, we can apply the Chernoff's bound in Eq. (16) for a sufficiently large β so that

$$\mathbb{P}\left(\Delta_v^{\text{in}} \geq \beta\frac{\log n}{\log\log n}\right) \leq \frac{1}{n^\alpha},\ \text{for some fixed}\ \alpha = \alpha(\beta) \geq 2\,.$$

We have not completed our job, yet! The above bound is for *one*, arbitrarily fixed node v. To bound the maximum degree we need the bound hold for all nodes, w.h.p. However, we are lucky since we can consider the union of the "bad" events

$$\mathcal{E}_v = \text{“}\ \Delta_v^{\text{in}} \geq \beta\frac{\log n}{\log\log n}\ \text{”},\ v\in[n]\,.$$

Then, from the last inequality, and since $\alpha \geq 2$, we get that

$$\mathbb{P}\left(\bigcup_v \mathcal{E}_v\right) \leq \sum_v \mathbb{P}(\mathcal{E}_v) \leq \frac{1}{n}\,.$$

The fact that $\Delta_v^{\text{in}} = \Omega(\log n/\log\log n)$ w.h.p. is rather more difficult to show: a good way to get it is to use a tight approximation of Binomial Distribution via a Poisson one. This part can be found in [3].

□

Remark. The student may verify the r.v.s Δ_v^{in}'s *are not mutually independent*. This is a crucial issue we will cope with in the next section.

4.2 Vertex Expansion of G (and its Diameter)

- Def. of Expansion
- . Good expansion implies small Diameter and good fault tolerance.
- Proof of expansion of subsets of size $O(n^\beta)$, for some constant $0 < \beta < 1$.
- Proof of good expansion of subsets of large size: the compression argument (next year?)

4.3 Expansion of G and Some Consequences

In the previous subsection, we proved that random graph G has a linear number of edges and its maximal degree is bounded by a sublogarithmic function, w.h.p. In network applications, these properties often represent necessary topology constraints in order to have efficient topology control and low local congestion. Other fundamental issues in this setting are those concerning the diameter and the fault-tolerance of G. It is clear that we would like G to have a small diameter and a good connectivity even if some edges will not work. To analyze the above two features, we need to introduce a fundamental concept in graph theory:

Definition 4.3. *Given a graph $G(V, E)$ with $V = [n]$, the (node)-expansion of any subset $S \subset [n]$ is defined as*

$$|N(S)| \quad \textit{where} \quad N(S) \doteq \{w \in V - S \,:\, (v, w) \in E \ \textit{for some}\ v \in S\}.$$

Then, for a fixed $\alpha \geq 0$, we say graph G is an α-expander *if any subset $S \subset [n]$ with $|S| \leq n/2$ has expansion at least $\alpha \cdot |S|$.*

Our interest in $\Omega(1)$-expander graphs is well-motivated by the following fact.

Theorem 4.4. *Consider an infinite family of graphs of increasing size*

$$\{G_n(V_n, E_n) \quad \textit{with}\ V_n = [n]\,,\ n \geq 1\}.$$

If, an absolute constant *$\alpha > 0$ exists such that, for sufficiently large n, graph G_n is an α-expander, then its diameter is $O(\log n)$.*
Moreover, under the same assumption above, in order to fully-disconnect any node subset S from the rest of the graph, the number of faulty links must be at least linear in the size of S.

Proof. We consider a fixed, sufficiently large graph $G_n(V_n, E_n)$ and omit the subscrit n for all the rest of the subsection, i.e.

$$G(V, E) \ \doteq \ G_n(V_n, E_n)\,.$$

Fix any $s \in V$, and let's make a Breadth-First Search (BFS) starting from s. Since G is α-expander, we define

$$L_t \ \doteq \ \{v \in V \ :\ d(s, v) = t\}\ , t \ = \ 0, 1, \ldots n - 1.$$

Observe that $L_0 = \{s\}$ and $L_1 = N(s) \geq \alpha$, so $L_1 \geq 1$ since $|N(s)|$ is an integer. Now, let us define the following family of subsets:

$$I_0 \doteq L_0 \ \text{ and } \ I_t \doteq I_{t-1} \cup L_t\,,\ t \ = \ 0, 1, \ldots.$$

Notice that, by construction, it holds that $|I_t| = |I_{t-1}| + |L_t|$ and, hence, since G is an α-expander, as long as $|I_{t-1}| \leq n/2$, we get

$$|I_t| \ = \ |I_{t-1}| + |L_t| \ \geq \ (1 + \alpha) \cdot |I_{t-1}| \ \geq \ (1 + \alpha)^{t-1}.$$

From the above inequality, for some $\tau = O(\log n)$, it must hold that the size $|I_\tau| \geq n/2$. This means that the number of nodes within distance τ from s is at least $n/2$. Now, consider a node (if any) $w \in V - I_\tau$ and repeat the same BFS process starting from w. Then, again, thanks to the expansion of G, after $\tau' = O(\log n)$ levels of the BFS tree rooted at w, we get that the corresponding subset $I'_{\tau'}$ has reached size at least $n/2$. The proof is now completed by observing that the two BFS trees (the one rooted at s and that rooted at w) must share at least one node.

□

4.3.1 Analysis of expansion by Chernoff's Bound: a non-useful counting argument

Let's consider a fixed subset $S \subseteq [n]$ s.t. $s = |S| \leq n/2$ and look at the following binary r.v.s:

$$\forall v \in V - S,\ Y_v = 1 \ \text{ iff } \ v \in \mathcal{N}^o(S)\ ,$$

where $\mathcal{N}^o(S)$ is the set of nodes in $V - S$ having (at least) one link to/from S. We easily get:

$$\mathbb{P}(Y_v = 1) \geq 1 - \left(1 - \frac{s}{n}\right)^d \approx \frac{ds}{n},\ \text{for } \frac{ds}{n} < 1 \tag{8}$$

So,

$$\mathbb{E}(|\mathcal{N}^o(S)|) \ = \ \sum Y_v \geq (n-s)\frac{ds}{n} \ \geq \ \frac{1}{2}\,ds \tag{9}$$

since Y_v are mutually independent, we can apply Chernoff's bound (14), and get that

$$\mathbb{P}(|\mathcal{N}^o(S)| \leq (1 - \frac{1}{2})\frac{1}{2}\,ds) \ \leq \ e^{-\frac{1}{8}\,ds} \tag{10}$$

Now this is the probability the one fixed S has a bad expansion. We now would like to apply the Union Bound over all subset of size s. If we do that, we would get that the probaility that a "bad" subset exist is at most

$$\binom{n}{d} \cdot e^{-\frac{1}{8}\,ds} \ \leq \ \left(\frac{en}{s}\right)^s \cdot e^{-\frac{1}{8}\,ds} \ = \ e^{s + s\log n - s\log s - 1/8ds} \leq ???? \tag{11}$$

Now, if $s = \Omega(n)$, we have no problem: for a suff. large d we would get a good bound. If $s = n^\beta$ for some constant $\beta < 1$, then, with another counting argument we can make a good bound as well. But what about, for instance, for $s = n/\log n$? Any hint are welcome!

A Useful inequalities

A.1 Markov Inequality

Let X by any random variable assuming only non-negative values and which has expectation μ. Then, for any real $t \geq 0$, it holds that

$$\mathbf{P}\left(X \geq t\right) \ \leq \ \frac{\mu}{t}\,. \tag{12}$$

A.2 Chernoff Bound moltiplicative form

Let $X_1, \ldots, X_n$ be independent 0-1 random variables. Let $X = \sum_{i=1}^n X_i$ and $\mathbf{E}\left[X\right] \leq \mu$. Then, for any $0 < \delta < 1$ the following Chernoff bounds hold:

$$\mathbf{P}\left(X \geq (1+\delta)\mu\right) \leq e^{-\mu\delta^2/3}. \tag{13}$$

$$\mathbf{P}\left(X \leq (1-\delta)\mu'\right) \leq e^{-\mu'\delta^2/2}. \tag{14}$$

Moreover, for any $\delta > 0$, the following Chernoff bounds hold:

$$\mathbf{P}\left(X \geq (1+\delta)\mu\right) \leq \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mu}. \tag{15}$$

From the last inequality, we get the following useful fact

Fact A.1. *Let $\mu \geq 1$ and let α be any positive constant. Then, for a sufficiently large constant $\beta > 0$, sufficiently large n, it holds that*

$$\mathbf{P}\left(X \geq \left(1 + \beta \cdot \frac{\log n}{\log\log n}\right)\mu\right) \leq e^{-\alpha\mu\log n}. \tag{16}$$

Proof. Consider Eq. 15 and fix $\delta := \beta \cdot \frac{\log n}{\log\log n}$, then make simple calculations and asymptotical approximations (the details are left as an homework). □

A.3 Chernoff Bound additive form

Let $X_1, \ldots, X_n$ be independent 0-1 random variables. Let $X = \sum_{i=1}^{n} X_i$ and $\mu = \mathbf{E}[X]$. Then the following Chernoff bounds hold:
for any $0 < \lambda < n - \mu$,

$$\mathbf{P}\left(X \leq \mu - \lambda\right) \leq e^{-2\lambda^2/n}, \tag{17}$$

for any $0 < \lambda < \mu$,

$$\mathbf{P}\left(X \geq \mu + \lambda\right) \leq e^{-2\lambda^2/n}. \tag{18}$$

A.4 The Method of Bounded Differences

In Section 3.2, we used the *Method of Bounded Differences*. In particular, we applied the following concentration bound [1, 2].

Theorem A.2. *Let $\mathbf{Y} = (Y_1, \ldots, Y_m)$ be independent r.v.s, with Y_j taking values in a set A_j. Suppose the real-valued funcion f defined on $\prod A_j$ satisfies*

$$|f(\mathbf{y}) - f(\mathbf{y}')| \leq \beta_j$$

whenever vectors $\mathbf{y}$ and $\mathbf{y}'$ differs only in the j-th coordinate. Let μ be the expected value of r.v. $F(\mathbf{Y})$. Then, for any $M > 0$, it holds that

$$\Pr\left(F(\mathbf{Y}) - \mu \geq M\right) \leq e^{-\frac{2M^2}{\sum_{j=1}^{m}\beta_j^2}}.$$

A.5 Reverse Chernoff Bound

Let $X_1, \ldots, X_n$ be independent 0-1 random variables, $X = \sum_{i=1}^{n} X_i$, $\mu = \mathbf{E}[X]$ and $\delta \in (0, 1/2]$. If $\mu \leq \frac{1}{2}n$ and $\delta^2\mu \geq 3$ then the following bounds hold:

$$\mathbf{P}\left(X \geq (1+\delta)\mu\right) \geq e^{-9\delta^2\mu}, \tag{19}$$

$$\mathbf{P}\left(X \leq (1-\delta)\mu\right) \geq e^{-9\delta^2\mu}. \tag{20}$$

References

[1] D.P. Dubhashi and A. Panconesi. *Concentration of measure for the analysis of randomized algorithms*. Cambridge University Press, 2009.

[2] Colin McDiarmid. Concentration. in probabilistic methods for algorithmic discrete mathematics, mcdiarmid c., ramirez-alfonsin j., reed b. (eds). *Algorithms and Combinatorics, Springer, Berlin, Heidelberg*, 16:195–248, 1998.

[3] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, New York, NY, USA, 2005.