The Compactness of Interval Routing for Almost All Graphs

Cyril Gavoille[1] and David Peleg[2]

[1] LaBRI, Université Bordeaux I, 351, cours de la Libération, 33405 Talence Cedex, France. E-mail: gavoille@labri.u-bordeaux.fr.

[2] Department of Applied Mathematics and Computer Science, The Weizmann Institute of Science, Rehovot, 76100 Israel. E-mail: peleg@wisdom.weizmann.ac.il.
Supported in part by grants from the Israel Science Foundation and from the Israel Ministry of Science and Art.

Abstract. Interval routing is a compact way for representing routing tables on a graph. It is based on grouping together, in each node, destination addresses that use the same outgoing edge in the routing table. Such groups of addresses are represented by some intervals of consecutive integers. We show that almost all the graphs, i.e., a fraction of at least $1 - 1/n^2$ of all the n-node graphs, support a shortest path interval routing with at most *three* intervals per outgoing edge, even if the addresses of the nodes are arbitrarily fixed in advance and cannot be chosen by the designer of the routing scheme. In case the addresses are initialized randomly, we show that *two* intervals per outgoing edge suffice, and conversely, that two intervals are required, for almost all graphs. Finally, if the node addresses can be chosen as desired, we show how to design in polynomial time a shortest path interval routing with a *single* interval per outgoing edge, for all but at most $O(\log^3 n)$ outgoing edges in each node. It follows that almost all graphs support a shortest path routing scheme which requires at most $n + O(\log^4 n)$ bits of routing information per node, improving on the previous upper bound.

1 Introduction

1.1 Background

A *universal routing strategy* is an algorithm which generates a routing scheme for every given network. One type of trivial universal routing strategy is based on schemes that keep in each node a full routing table which specifies an output port for every destination. Though this strategy can guarantee routing along shortest paths, each router has to store locally $\Theta(n \log d)$ bits of memory, where d is the degree of the router (i.e., the number of output ports) and n is the number of nodes in the network.

The *interval routing scheme* [9, 10] is a compact routing scheme, i.e., a routing scheme that needs to keep only a small amount of information in each node to route messages correctly through the network. The idea of this scheme is to label the n nodes of the network with unique integers from $\{1, \ldots, n\}$, and to label the outgoing arcs in every node with a set of intervals forming a partition of the name range. The routing process sends a message on the unique outgoing arc labeled by an interval that contains the destination label. While the preprocessing stage of such a routing scheme (which is performed once in the initialization of the network) might be complex, the delivery protocol consists of simple decision functions which can be implemented with $O(kd \log n)$ bits in each node of degree d, where k is the maximum number of intervals assigned to an arc. Such a routing scheme supports a compact implementation whenever k is small in comparison with n or d.

In [8], it is showed that there is no universal routing strategy that can guarantees a shortest path routing scheme with less than $\Omega(n \log d)$ bits per node for all the n-node networks of maximum degree d. This result means that there is some worst-case network where for any shortest path routing function, the number of bits required to be stored in a router is not significantly smaller than the size of a routing table, whatever the node labeling[1] and the shortest paths are. Fortunately, such a problematic situation where the routing tables cannot be compressed occurs for a limited number of worst-case networks only.

In particular, in [3], it is shown that for almost all the n-node networks the size of the routing tables can be reduced to $O(n)$ bits per node. More precisely, it is shown that all networks but a $1/n^3$ fraction, can be routed with a scheme that uses $3n + o(n)$ bits of memory, under the assumption that nodes are arbitrary labeled in the range $\{1, \ldots, n\}$, and that every node knows its neighbors for "free". Moreover, if during the initialization process of the network, nodes can be relabeled with binary string of length $O(\log^2 n)$ bits, then $O(\log^2 n)$ bits per node suffice to route along the shortest paths for almost all networks.

1.2 Definitions and Results

In this paper we consider shortest path routing schemes only. An undirected graph $G = (V, E)$ represents the classic model of the underlying topology of the network. An n-node graph G with the nodes labeled in the set $\{1, \ldots, n\}$ *supports a k-interval routing scheme* (k-IRS for short) if there exists an interval routing scheme $\mathcal{R}$ for G with the property that for every (directed) edge e, the set of node labels to which $\mathcal{R}$ routes messages via e is composed of at most k intervals. (An interval means a set of consecutive integers taken from $\{1, \ldots, n\}$, where n and 1 are considered to be consecutive.)

Our goal is to find a labeling of the nodes and a shortest paths system in order to minimize the maximum number of intervals assigned to the edges of the graph. We distinguish three models depending on the freedom we have in labeling the nodes.

1. **[Adversary]:** labels are fixed in advance (by an adversary) and cannot be permuted;
2. **[Random]:** labels are randomly permuted;

[1] from the range $\{1, \ldots, n\}$.

3. **[Designer]:** labels can be chosen (by the routing designer) in order to achieve the smallest possible number of intervals.

In all three models, the routing designer has the freedom of selecting the shortest paths to be used.

Corresponding to these three models, we introduce the following three parameters. We denote by $\mathrm{IRS}_A(G)$ the smallest integer k such that G supports a k-IRS in the adversary model (namely, for every arbitrary labeling of the nodes). We denote by $\mathrm{IRS}_R(G)$ the smallest k such that G supports a k-IRS in the random model (namely, given a random labeling of the nodes of G) with high probability. Finally, we denote by $\mathrm{IRS}(G)$ the smallest k such that G supports a k-IRS in the designer model (namely, under some specifically chosen node labeling of G). Clearly, $\mathrm{IRS}(G) \leq \mathrm{IRS}_R(G) \leq \mathrm{IRS}_A(G)$, for every graph G.

The parameter $\mathrm{IRS}(G)$ has been computed for many classes of graphs (see [6] for a recent overview). Notably, in [7] it is shown that for every G, $\mathrm{IRS}_R(G) < n/4 + o(n)$, whereas there exists some worst-case G_0 such that $\mathrm{IRS}(G_0) > n/4 - o(n)$. However, as shown in this paper, the situation is considerably better for the "average" case. Specifically, we will see that $\mathrm{IRS}(G) \leq 2$ for a fraction of at least $1 - 1/n^2$ of all the n-node labeled graphs.

Technically, we use random graphs instead of the Kolmogorov random graphs used in [3]. A discussion about the relationships between random and Kolmogorov random graphs can be found in [4]. The class $\mathcal{G}_{n,p}$ denotes the classic model of n-node labeled random graphs, where $0 \leq p \leq 1$ represents the probability of having an edge between any two nodes. Interval routing on random graphs has been first investigated in [5], where some lower bounds are given for $\mathrm{IRS}(G)$ for $G \in \mathcal{G}_{n,p}$. More precisely, it is shown therein that for $p = n^{-1+1/s}$ for integer $s > 0$, such that there exists a ε satisfying $(\ln^{1+\varepsilon} n)/n < p < n^{-1/2-\varepsilon}$, a graph $G \in \mathcal{G}_{n,p}$ satisfies

$$\mathrm{IRS}(G) \;\geq\; \frac{1}{10}\, n^{1-6/\ln(np) - \ln(np)/\ln n} \tag{1}$$

with high probability. It is also shown that for some $p = n^{-1+1/\Theta(\sqrt{\log n})}$, a graph $G \in \mathcal{G}_{n,p}$ satisfies $\mathrm{IRS}(G) = \Omega(n^{1-1/\Theta(\sqrt{\log n})})$ with high probability. In this paper we investigate the case where p is a fixed constant, e.g. $p = 1/2$, in order to establish some average results on the total space of n-node graphs. (Note that for constant p, Eq. (1) cannot be used since in this case p lies outside the validity range.)

The following table presents our results for each model. The results of the table are proved for a fraction of at least $1 - 1/n^2$ of all the n-node graphs.

Label select	Designer	Random	Adversary
Upper bound	$\mathrm{IRS} \leq 2$	$\mathrm{IRS}_R \leq 2$	$\mathrm{IRS}_A \leq 3$
Lower bound	$\mathrm{IRS} \geq 1$	$\mathrm{IRS}_R \geq 2$	$\mathrm{IRS}_A \geq 2$

At this time, we are still unable to decide whether $\mathrm{IRS}(G) = 1$ or 2 for almost every graph G, if node labels and shortest paths system can be chosen in advance by the designer. However, we present a polynomial time algorithm to design a 2-IRS for all graphs but a $1/n$ fraction, such that for every node, all its outgoing edges are labeled with a single interval, except for up to $O(\log^3 n)$ edges where 2 intervals are required. It follows that almost every graph supports a shortest path routing scheme that can be implemented with $n + O(\log^4 n)$ bits, improving on the best known result (cf. [3]). Note that our result is stated without the assumption that nodes know theirs neighbors.

2 Randomly Assigned Node Labels

In this section we show that in the random model, almost every graph G satisfies $\mathrm{IRS}_R(G) = 2$. This implies, in particular, that almost every graph G satisfies $\mathrm{IRS}(G) \leq 2$. This is done by showing that, with probability at least $1 - 1/n^2$, a random graph G from $\mathcal{G}_{n,1/2}$ satisfies $\mathrm{IRS}_R(G) = 2$. Actually, we show that the result holds for the class $\mathcal{G}_{n,p}$ of random graphs for each fixed probability $0 < p < 1$.

2.1 Upper Bound

In this subsection we shall prove that $\mathrm{IRS}_R(G) \leq 2$.

Assume the node labels $\{1, \ldots, n\}$ are assigned randomly for the graph G. In that case, given that G is a random graph in $\mathcal{G}_{n,p}$, we may assume that the nodes are first marked by the labels 1 through n, and only then do we draw the edges randomly and uniformly with probability p.

For random graphs selected from $\mathcal{G}_{n,p}$ we have the following simple bounds. (We state a variant of the bounds suitable to our needs, and make no attempt to optimize them; see [2] for sharper statements.) We denote by $\Gamma(v)$ the set composed of v and of its neighbors.

Lemma 1. *With probability at least $1 - 1/n^3$, and for every fixed p, $0 < p < 1$, a random graph $G \in \mathcal{G}_{n,p}$ is of diameter 2, and for every node $v \in V$,*

$$np - 3\sqrt{n \ln n} \;\leq\; |\Gamma(v)| \;\leq\; np + 3\sqrt{n \ln n} \;.$$

Let $\mathcal{E}_A$ denote the event that the random graph at hand does not satisfy the properties asserted in Lemma 1. Henceforth we ignore that possibility, and restrict our attention to $\bar{\mathcal{E}}_A$.

For notational convenience we identify nodes with their labels, i.e., denote $V = \{1, \ldots, n\}$.

Consider a node $v_0 \in V$. We need to argue that with high probability, the edges of v_0 can be labeled with at most two intervals per edge so that for every possible destination $v_d \in V$, the selected edge is along a shortest path from v_0 to v_d.

Let $A = \Gamma(v_0) \setminus \{v_0\}$ and $B = V \setminus \Gamma(v_0)$. Since G satisfies the event $\bar{\mathcal{E}}_A$,

$$np - 3\sqrt{n \ln n} \;\leq\; |A|, |B| \;\leq\; np + 3\sqrt{n \ln n} \;.$$

Let

$$C \;=\; \{v \in B \mid v+1 \in A \;\text{ and }\; (v, v+1) \in E\}.$$

Lemma 2. *With probability at least $1 - 1/n^3$, the size of the set C is bounded by*

$$np^3 - 6\sqrt{n \ln n} \;\leq\; |C| \;\leq\; np^3 + 6\sqrt{n \ln n}.$$

Proof. Consider a vertex $v \in B$, and let I_v denote the event that $v \in C$. This event happens precisely if $v+1 \in A$ and $(v, v+1) \in E$. These two sub-events are independent and both occur with probability p, hence $\mathbb{P}(I_v) = p^2$. Note also that the events I_v for $v \in B$ are mutually independent. Let Z be a random variable denoting the size of $|C|$. Then $Z = \sum_{v \in B} z_v$, where z_v is the characteristic random variable of the event I_v. Hence Z is the sum of $|B|$ mutually independent Bernoulli variables, and its expected value is $\mathbb{E}(Z) = |B|p^2$, and hence applying Chernoff's bound (cf. [1]) we get,

$$\begin{aligned}
\mathbb{P}\left(Z > np^3 + 6\sqrt{n \ln n}\right) &< \mathbb{P}\left(Z > \left(1 + \frac{3}{p^2}\sqrt{\frac{\ln n}{n}}\right)\mathbb{E}(Z)\right) \\
&< \exp\left(-\frac{9 \ln n}{3np^4}\,\mathbb{E}(Z)\right) \\
&< \exp\left(-\frac{3}{p}\ln n + o(1)\right) \\
&< \frac{1}{n^3}
\end{aligned}$$

and

$$\mathbb{P}\left(Z < np^3 - 6\sqrt{n \ln n}\right) \;<\; \frac{1}{n^3}$$

and the lemma follows. □

Let $\mathcal{E}_B$ denote the event that the random graph at hand does not satisfy the property asserted in Lemma 2 for *some* node v_0. Note that the probability for this event is bounded above by $1/n^3$. Henceforth we ignore that possibility, and restrict our attention to $\bar{\mathcal{E}}_B$.

Let us now define one interval per emanating edge of v_0 to take care of routing to the nodes in $A \cup C$. For every node $w \in A$, mark the edge (v_0, w) by the interval $[w-1, w]$ if $w - 1 \in C$, and by the interval $[w]$ if $w - 1 \notin C$.

It is thus left to show how the remaining interval per edge of v_0 can be used to route optimally towards the nodes in $X = B \setminus C$. This is done as follows. Let $X = \{x_1, \ldots, x_m\}$. Note that since G satisfies the events $\bar{\mathcal{E}}_A$ and $\bar{\mathcal{E}}_B$,

$$np(1-p^2) - 9\sqrt{n \ln n} \;\leq\; m \;\leq\; np(1-p^2) + 9\sqrt{n \ln n}.$$

We will now describe a process for selecting a subset of A, denoted $Y = \{y_1, \ldots, y_m\} \subseteq A$, such that there is an edge $(x_i, y_i) \in E$ for every $1 \leq i \leq m$. Once this is done, we mark each edge (v_0, y_i) by the interval $[x_i]$, thus completing our task.

The selection process is a straightforward greedy one. Let $Q = A$. Having already selected $y_1, \ldots, y_{i-1}$, the ith step consists of selecting y_i to be some arbitrary neighbor of x_i in Q, and discarding y_i from Q. If, at any stage, the node x_i considered by the process has no neighbors in the remaining set Q, then the process fails and we abort our attempt to provide a 2-IRS for G.

We need to argue that with very high probability, the process does not fail. Let F_i be the event that the process fails in the ith step. Note that at the beginning of step i, the current set Q is of size

$$|Q| \;=\; |A| - i + 1 \;\geq\; |A| - m \;\geq\; np - np(1-p^2) - 12\sqrt{n \ln n} \;>\; \frac{np^3}{2}$$

for sufficiently large n.

Event F_i occurs only if x_i is not connected to any node of Q. This is the intersection of $|Q|$ independent events of probability $1-p$ each, hence

$$\mathbb{P}(F_i) \;\leq\; (1-p)^{np^3/2} \;=\; \left((1-p)^{p^3/2}\right)^n \;<\; c^{-n}$$

For constant $c > 1$. Letting $\mathcal{E}_F(v_0)$ denote the event that the process fails for v_0, i.e., $\mathcal{E}_F(v_0) = \bigcup_i F_i$, we have $\mathbb{P}(\mathcal{E}_F(v_0)) < m\,c^{-n}$. It follows that for a sufficiently large n, the event $\mathcal{E}_F = \bigcup_{v_0} \mathcal{E}_F(v_0)$ has probability $\mathbb{P}(\mathcal{E}_F) \leq 1/n^3$.

Combining all possible failure events (namely, $\mathcal{E}_A \cup \mathcal{E}_B \cup \mathcal{E}_F$), we get that for sufficiently large n, the probability that our process fails to generate an interval routing scheme for the graph with two intervals per edge is bounded from above by $1/n^2$.

Theorem 1. *For sufficiently large n, with probability at least $1 - 1/n^2$, and for every fixed p, $0 < p < 1$, a random graph $G \in \mathcal{G}_{n,p}$ satisfies* $\mathrm{IRS}_R(G) \leq 2$. □

2.2 Lower Bound

In this subsection we shall prove that $\mathrm{IRS}_R(G) \geq 2$ for almost every graph G, for a random assignment of node labels.

Again, we assume the node labels $\{1, \ldots, n\}$ are assigned randomly for the graph G, so given that G is a random graph in $\mathcal{G}_{n,p}$, for p fixed, we may assume that the nodes are first labeled 1 through n and the edges are randomly drawn only later. As in the previous subsection, we assume the event $\bar{\mathcal{E}}_A$.

We need to show that with high probability, a single interval per edge will not be sufficient for producing shortest paths.

Consider a node $x \in \{1, \ldots, n\}$. Suppose that x is connected to $x+1$ and $x+3$, and that $x+2$ is not connected to any node from $\{x, x+1, x+3\}$. Let $I(x,u)$ be the interval assigned to the edge (x,u) that contains $x+2$. Since the diameter of G is 2, it follows that $u \notin \{x+1, x+2, x+3\}$. $I(x,u)$ must contain u and $x+2$, but neither $x+1$ nor $x+3$ which are connected to x. A contradiction for the fact that $I(x,u)$ is composed of a single interval.

Let $x_i = 4i - 3$, for every $i \in \{1, \ldots, m\}$, with $m = \lfloor n/4 \rfloor$. Let K_i denote the event x_i is in the previous configuration, and let $\mathcal{E}_K$ denote the event that there exists an event K_{i_0} that occurs. Note that from the above discussion, the probability of $\mathrm{IRS}_R(G) > 1$ (under the event $\bar{\mathcal{E}}_A$) is lower bounded by $\mathbb{P}(\mathcal{E}_K)$.

Let Z_i be the characteristic random variable of the event K_i, and $Z = \sum_{i=1}^{m} Z_i$. The events K_i are independent, and each one occurs with probability p^5. So, $\mathbb{P}(Z = 0) = (1 - p^5)^m < 1/n^3$, for a sufficiently large n. It follows that $\mathbb{P}(\mathcal{E}_K) \geq 1 - 1/n^3$.

Theorem 2. *For sufficiently large n, with probability at least $1 - 1/n^2$, and for every fixed p, $0 < p < 1$, a random graph $G \in \mathcal{G}_{n,p}$ satisfies $\mathrm{IRS}_R(G) \geq 2$.* □

3 Adversely Assigned Labels

We next assume the adversary model, in which the assignment of the node labels $\{1, \ldots, n\}$ to vertices is done by an adversary, aiming to cause the routing scheme to use the maximum number of intervals.

In this case, the lower bound of Section 2.2 remains unchanged, but the only upper bound we can prove is that $\mathrm{IRS}_A(G) \leq 3$ for almost every graph G. Actually, we show that the result holds for the class $\mathcal{G}_{n,p}$ of random graphs for each fixed probability p, $0 < p < 1$.

Once again, by Lemma 1, we are allowed to restrict our attention to the event $\bar{\mathcal{E}}_A$, and assume the graph $G = (V, E)$ at hand is of diameter 2, and such that for every node $v \in V$, $np - 3\sqrt{n \ln n} \leq |\Gamma(v)| \leq np + 3\sqrt{n \ln n}$.

Consider a node $v_0 \in V$. We need to argue that with high probability, the edges of v_0 can be labeled with at most three intervals per edge so that for every possible destination $v_d \in V$, the selected edge is along a shortest path from v_0 to v_d.

Let $A = \Gamma(v_0) \setminus \{v_0\}$ and $B = V \setminus \Gamma(v_0)$. Let us first define one interval per emanating edge of v_0 to take care of routing to the nodes of A. Namely, for every node $w \in A$, mark the edge (v_0, w) by the interval $[w]$. It is left to show how the remaining two intervals per edge of v_0 can be used to route optimally towards the nodes of B. This is done as follows. Let $B = \{b_1, \ldots, b_m\}$. Recall that

$$np - 3\sqrt{n \ln n} \;\leq\; |A|, |B| \;\leq\; np + 3\sqrt{n \ln n}.$$

We will now describe a process for selecting an intermediate node $a_i \in A$ for every $1 \leq i \leq m$, such that the routing from v_0 to b_i will go through a_i. For this, we need to ensure that there is an edge $(a_i, b_i) \in E$ for every $1 \leq i \leq m$. Once this is done, we mark each edge (v_0, a_i) by the interval $[b_i]$, thus completing our task.

The selection process is similar to the greedy process of Section 2.1. Let $Q = A$, and define a counter $C(a)$ for each node $a \in A$, initially setting all counters to zero. Having already selected $a_1, \ldots, a_{i-1}$, the ith step consists of selecting a_i to be some arbitrary neighbor of b_i in Q, increasing the counter $C(a_i)$ by one, and discarding a_i from Q if the counter has reached two. If, at any stage, the node b_i considered by the process has no neighbors in the remaining set Q, then the process fails and we abort our attempt to provide a 3-IRS for G.

We need to argue that with very high probability, the process does not fail. Let F_i be the event that the process fails in the ith step. Note that at the beginning of step i, the counters sum up to $i-1$, hence at most $\lfloor (i-1)/2 \rfloor$ nodes were discarded from Q, so the current set Q is of size

$$|Q| \ \geq\ |A| - \left\lfloor \frac{i-1}{2} \right\rfloor \ \geq\ |A| - \frac{m}{2} \ \geq\ np - \frac{np}{2} - 6\sqrt{n \ln n} \ \geq\ \frac{np}{3}$$

for sufficiently large n.

Event F_i occurs only if b_i is not connected to any node of Q. This is the intersection of $|Q|$ independent events of probability $1-p$ each, hence $\mathbb{P}(F_i) < c^{-n}$, for constant $c > 1$. Letting $\mathcal{E}_F(v_0)$ denote the event that the process fails for v_0, i.e., $\mathcal{E}_F(v_0) = \bigcup_i F_i$, we have $\mathbb{P}(\mathcal{E}_F(v_0)) \leq m\, c^{-n}$. It follows that for a sufficiently large n, the event $\mathcal{E}_F = \bigcup_{v_0} \mathcal{E}_F(v_0)$ has probability $\mathbb{P}(\mathcal{E}_F) \leq 1/n^3$.

Combining all possible failure events (namely, $\mathcal{E}_A \cup \mathcal{E}_F$), we get that the probability that our process fails to generate an interval routing scheme for the graph with three intervals per edge is bounded from above by $1/n^2$.

Theorem 3. *For sufficiently large n, with probability at least $1 - 1/n^2$, and for every fixed p, $0 < p < 1$, a random graph $G \in \mathcal{G}_{n,p}$ satisfies $\mathrm{IRS}_A(G) \leq 3$.* □

4 Designer Chosen Labels

We next assume the designer model, in which the assignment of the node labels $\{1, \ldots, n\}$ to vertices is done be the designer of the routing scheme, aiming to minimize the number of intervals used by the routing scheme.

In this case, the only lower bound we have at the moment is the trivial $\mathrm{IRS}(G) \geq 1$ for every graph G. In the opposite direction, we are also unable so far to prove an upper bound of 1 on the maximum number of intervals per edge.

However, we will show that it is possible to assign the node labels in such a way that, while some edges might still require two intervals, the number of such violations will be very small, and more specifically, bounded by $O(\log^3 n)$ with high probability. In this section we restrict our attention to the case $p = 1/2$, so $G \in \mathcal{G}_{n,1/2}$.

The idea behind the selection process is the following. Suppose that the vertex set of the given random graph is partitioned into cliques $V = C_1 \cup \ldots \cup C_m$. Label the nodes of V according to this partition, so that the nodes of each clique C_i are numbered consecutively. Now use this partition to define the routing scheme as follows. Consider a sender v_0. Suppose that $v_0 \in C_J$, and consider some other clique C_I. The central property we rely upon is that if v_0 is adjacent to *some* of the nodes of C_I, then all the vertices of C_I can be provided for using a single interval on each edge going from v_0 to the nodes of C_I, as follows. Let $C_I = \{p, p+1, \ldots, q\}$. If v_0 has a unique neighbor ℓ in C_I, then mark the edge from v_0 to ℓ by the interval $[p, q]$. Otherwise, suppose v_0 has neighbors $\ell_1 < \ell_2 < \ldots < \ell_k$ in C_I. Then the edges $e_j = (v_0, \ell_j)$ leading from v_0 to these nodes can be labeled by intervals $I(e_j)$ as follows.

$$I(e_j) = \begin{cases} [p, \ldots, \ell_2 - 1], & j = 1, \\ [\ell_j, \ldots, \ell_{j+1} - 1], & 1 < j < k, \\ [\ell_k, \ldots, q], & j = k. \end{cases}$$

(Note that this choice of intervals takes care also of the special case of $C_I = C_J$ itself, where every node other than v_0 itself is a neighbor of v_0.)

Thus we are left only with the need of handling the cliques C_i none of whose vertices are adjacent to v_0. Call these cliques the "remote" cliques. The nodes of these remote cliques must be reached through vertices of other cliques, potentially using additional intervals, and at worst, using a unique new interval for each node. It is thus required to bound from above the maximum number of nodes in the remote cliques. Towards this goal, we rely intuitively on the fact that large cliques are unlikely to be remote. More precisely, the probability that a clique of size k is remote is roughly $1/2^k$. It thus becomes necessary to explore the distribution of clique sizes in a clique partition of random graphs, or at least generate partitions with favorable size distributions.

We make use of the following two properties of random graphs. (In the following the function log denotes the logarithm in base 2.) First, regarding the size of the maximum clique, we have (cf. Chapt. XI.1 of [2])

Lemma 3. *With probability at least $1 - 1/n^{\log\log n}$, the maximum clique in a random graph $G \in \mathcal{G}_{n,1/2}$ is of size at most $2\log n$.*

Let $\mathcal{E}_C$ denote the event that the random graph at hand does not satisfy the property asserted in Lemma 3. Henceforth we ignore that possibility, and restrict our attention to $\bar{\mathcal{E}}_C$. As before, we also restrict ourselves to $\bar{\mathcal{E}}_A$.

Secondly, we make use of a natural technique for generating a clique partition of a given graph. This technique is the "mirror image" of the greedy algorithm often used to generate a legal coloring for a graph. This simple algorithm operates as follows. Start by ordering the vertices arbitrarily, numbering them as $1, 2, \ldots, n$. Assign the vertices to cliques $C_1, C_2, \ldots, C_n$ one by one, assigning each node to the smallest-indexed admissible clique. Vertex 1 is thus assigned to C_1, vertex 2 is assigned to C_1 if it is a neighbor of vertex 1, otherwise it is assigned to C_2, and so on. It is known (cf. Chapt. XI.3 of [2]) that with high probability this process will pack the vertices of the given random graph G in fewer than $n/\log n$ cliques. Moreover, analyzing the process in more detail, we will derive bounds on the number of small cliques generated. Specifically, there will be no more than $2^k \log n$ cliques of size k, with high probability. Coupled with Lemma 3, this can be used to show that the total number of nodes in remote cliques is bounded (with high probability) by about

$$\sum_{k=1}^{2\log n} k \cdot \frac{1}{2^k} \cdot 2^k \log n = O(\log^3 n).$$

The problem that makes formalizing this argument somewhat more difficult is that once the partition is calculated, the graph can no longer be treated as random, as the fact, say, that v_0 is not in the clique C_i, bears some implications on the probability that v_0 is connected to some node of C_i, and prevents us from assuming that all the events considered in the analysis are independent. Nevertheless, the dependencies can be bounded and turn to have but little effect on the resulting probabilities.

Let us fix our attention on a vertex v_0, belonging to the clique C_J, and on another clique C_I. We would like to bound the probability that v_0 is not connected to any node of C_I.

For every clique C_i and vertex $v \in V$, partition C_i into $C_i = \mathcal{A}_i(v) \cup \mathcal{B}_i(v)$, where $\mathcal{A}_i(v)$ consists of all the vertices that entered C_i before v was considered by the algorithm, namely, $\mathcal{A}_i(v) = \{w \in C_i \mid w < v\}$, and $\mathcal{B}_i(v) = C_i \setminus \mathcal{A}_i(v)$, the vertices added to C_i after v was added to some clique. Let $\alpha_i(v) = |\mathcal{A}_i(v)|$ and $\beta_i(v) = |\mathcal{B}_i(v)|$. In particular, let $\mathcal{A} = \mathcal{A}_I(v_0)$, $\mathcal{B} = \mathcal{B}_I(v_0)$, $\alpha = \alpha_I(v_0)$ and $\beta = \beta_I(v_0)$.

Lemma 4. *If $I < J$ then the probability that C_I is remote from v_0 is $1/2^{|C_I|-1}$.*

Proof. We will actually prove the somewhat stronger claim that if $I < J$ then the probability that v_0 is not connected to any vertex in C_I is $\frac{1}{2^\beta(2^\alpha-1)}$.

Since $I < J$, when the greedy algorithm considered v_0 it had to examine (and reject) the possibility of adding it to C_I, before actually adding it to C_J. The fact that v_0 was not added to C_I implies that there is some node in $\mathcal{A}$ that does not neighbor v_0. However, of all 2^α possible connection configurations between v_0 and the nodes of $\mathcal{A}$, the event $\mathcal{E}_N$ = "v_0 has a non-neighbor in $\mathcal{A}$" only excludes the possibility that v_0 neighbors all nodes of $\mathcal{A}$, and leaves us with $2^\alpha - 1$ other possibilities. Hence conditioned on $\mathcal{E}_N$, we have

$$\mathbb{P}(v_0 \text{ has no neighbors in } \mathcal{A}) \;=\; \frac{1}{2^\alpha - 1}.$$

As for the nodes of $\mathcal{B}$, each such node v was added to C_I after v_0 was considered, and since $I < J$, the decision to add v into C_I was reached before considering clique C_J, hence it was independent of the existence (or nonexistence) of the edge (v, v_0). Hence

$$\mathbb{P}(v_0 \text{ has no neighbors in } \mathcal{B}) \;=\; \frac{1}{2^\beta}.$$

The lemma follows. □

Lemma 5. *If $I > J$ then the probability that C_I is remote from v_0 is*

$$\frac{1}{2^\alpha} \cdot \prod_{v \in \mathcal{B}} \frac{2^{\alpha_J(v)-1}}{2^{\alpha_J(v)} - 1}.$$

Proof. Since $I > J$, when the greedy algorithm considered each node v of C_I it had to first examine (and reject) the possibility of adding it to C_J. For $v \in \mathcal{A}$, the decision not to add v to C_I was clearly independent of the edge (v, v_0). Hence

$$\mathbb{P}(v_0 \text{ has no neighbors in } \mathcal{A}) \;=\; \frac{1}{2^\alpha}.$$

It remains to consider nodes $v \in \mathcal{B}$.

The fact that a node $v \in \mathcal{B}$ was not added to C_J implies that there exists a node in $\mathcal{A}_J(v)$ that does not neighbor v. But again, of all $2^{\alpha_J(v)}$ possible connection configurations between v and the nodes of $\mathcal{A}_J(v)$, the event $\mathcal{E}_N(v) =$ "v has a non-neighbor in $\mathcal{A}_J(v)$" only excludes the possibility that v neighbors all nodes of $\mathcal{A}_J(v)$, and leaves us with $2^{\alpha_J(v)} - 1$ other possibilities. Of those, v neighbors v_0 in exactly $2^{\alpha_J(v)-1}$ possibilities. Hence conditioned on $\mathcal{E}_N(v)$, the probability that v does not neighbor v_0 is $\frac{2^{\alpha_J(v)-1}}{2^{\alpha_J(v)}-1}$. Hence

$$\mathbb{P}(v_0 \text{ has no neighbors in } \mathcal{B}) \;=\; \prod_{v \in \mathcal{B}} \frac{2^{\alpha_J(v)-1}}{2^{\alpha_J(v)} - 1}.$$

The lemma follows. □

The product appearing in the bound of Lemma 5 is small only when the values $\alpha_J(v)$ involved in it are sufficiently large. Fortunately, there cannot be too many nodes v with small $\alpha_J(v)$ values, as we prove next.

For integer $k \geq 1$, let $X_J(k)$ denote the set of vertices v that were considered by the algorithm during the period when C_J contained exactly k nodes, and were rejected from C_J. In particular, we are interested in the collection of such vertices for small values of k, i.e., $\hat{X} = \bigcup_{k=1}^{\log\log n} X_J(k)$.

Corollary 1. *Suppose that the clique C_I, $I > J$, contains no vertex from $\hat{X}$. Then the probability that v_0 is not connected to any vertex in C_I is at most $\gamma/2^{|C_I|}$, for some fixed constant $\gamma > 0$.*

Proof. Under the assumption of the corollary, $\alpha_J(v) > \log\log n$ for every $v \in \mathcal{B}$. Therefore

$$\frac{2^{\alpha_J(v)-1}}{2^{\alpha_J(v)} - 1} \;=\; \frac{1}{2}\left(1 + \frac{1}{2^{\alpha_J(v)} - 1}\right) \;\leq\; \frac{1}{2}\left(1 + \frac{1}{2^{\log\log n + 1} - 1}\right) \;\leq\; \frac{1}{2}\left(1 + \frac{1}{\log n}\right).$$

The bound of Lemma 5 thus becomes

$$\frac{1}{2^{\alpha}} \cdot \left(\frac{1}{2}\left(1 + \frac{1}{\log n}\right)\right)^{\beta}.$$

As the size of the maximum clique in a random graph is at most $2\log n$ (with probability at least $1 - 1/n^{\log\log n}$), this bound is no greater than

$$\frac{1}{2^{\alpha}} \cdot \frac{1}{2^{\beta}}\left(1 + \frac{1}{\log n}\right)^{2\log n} \;\leq\; \frac{1}{2^{\alpha+\beta}} \cdot e^2\,,$$

and the claim follows. □

Lemma 6. *With probability at least $1 - 1/n^3$, the set $X_J(k)$ is of size $|X_J(k)| \leq 2^{k+2}\ln n$ for every $k \geq 1$.*

Proof. Suppose that $|X_J(k)| > 2^{k+2}\ln n$. For every $v \in X_J(k)$, the probability for v not joining C_J (on account of a missing edge from v to some node in C_J) is $1/2^k$. Thus the probability of all of those vertices being rejected from C_J is

$$\left(\frac{1}{2^k}\right)^{|X_J(k)|} < \left(\frac{1}{2^k}\right)^{2^{k+2}\ln n} \le e^{-4\ln n} = \frac{1}{n^4} .$$

Summing these probabilities over all k yields the desired claim. □

Let $\mathcal{E}_D$ denote the event that the random graph at hand does not satisfy the property asserted in Lemma 6. Henceforth we ignore that possibility, and restrict our attention to $\bar{\mathcal{E}}_D$. Under this restriction, the size of the set $\hat{X}$ is bounded above by

$$|\hat{X}| \le \sum_{k=1}^{\log\log n} 2^{k+2}\ln n = O(\log^2 n).$$

It remains to bound the number of remote cliques C_I (that have no neighbor of v_0). Let $f(k)$ denote the number of cliques of size k.

Lemma 7. *With probability at least $1 - 1/n^2$, $f(k) \le 2^{k+2}\ln n$ for every $k \ge 1$.*

Proof. Let us bound the probability of the event that there are more than $2^{k+1}\log n$ cliques of size k, $C_{i_1}, \ldots, C_{i_{f(k)}}$. Let $m = 2^{k+2}\ln n$ and consider the time when clique C_{i_m} was formed by the greedy algorithm (for the purpose of hosting the currently inspected vertex v'). For any vertex v considered after v', the probability that it could not have joined the clique C_{i_j} is

$$1 - \frac{1}{2^{\alpha_{i_j}(v)}} \le 1 - \frac{1}{2^k} .$$

Hence the probability that v could not have joined any of those m cliques is at most

$$\left(1 - \frac{1}{2^k}\right)^m \le \left(1 - \frac{1}{2^k}\right)^{2^{k+2}\ln n} \le e^{-4\ln n} = \frac{1}{n^4} .$$

Consequently, the probability that any of the remaining vertices to be considered by the algorithm after v' could not join an existing clique, and a new clique must be formed, is at most $1/n^3$. Summing these probabilities for every k, the lemma follows. □

Let $\mathcal{E}_H$ denote the event that the random graph at hand does not satisfy the property asserted in Lemma 7. Henceforth we ignore that possibility, and restrict our attention to $\bar{\mathcal{E}}_H$.

Lemma 8. *The number of remote cliques is at most $O(\log^2 n)$, with probability $1 - 1/n^2$.*

Proof. Assuming event $\bar{\mathcal{E}}_D$, the total number of remote cliques that contain a vertex of $\hat{X}$ is at most $O(\log^2 n)$. It remains to count the remote cliques among the cliques that do not contain any vertex of $\hat{X}$. The probability of such a clique

C_I being remote is bounded, in Lemma 4 and Corollary 1, by $\delta/2^{|C_I|}$ for some constant $\delta > 1$.

For every clique C_i of size k, let R_i be the event that C_i is remote. Let R be a random variable representing the number of remote cliques of size k, and let $f_R(k)$ denote its expectation. Since R is the sum of $f(k)$ Bernoulli random variables R_i, each with probability $\delta/2^k$, $f_R(k)$ is at most $\delta f(k)/2^k$. Assuming event $\bar{\mathcal{E}}_H$, we have

$$f_R(k) \;\leq\; \frac{\delta 2^{k+2} \ln n}{2^k} \;=\; 4\delta \ln n.$$

Applying Chernoff's bound, we get that

$$\mathbb{P}(R \geq 8\delta \ln n) \;<\; \exp(-2\delta \ln n) \;=\; n^{-2\delta} \;<\; \frac{1}{n^2}\,.$$

Hence for the cliques not containing any vertices from $\hat{X}$, with probability at least $1 - 1/n^2$, the total number of remote cliques of any size k is bounded (recalling Lemma 3) by $O(\log^2 n)$.

Combining both clique types together, we get the claim of the lemma. □

It follows that for every v_0, the number of "problematic" vertices (namely, those of remote cliques) that need to be assigned an individual interval is bounded by $O(\log^3 n)$, with probability $1 - 1/n$.

Combining all possible failure events (namely, $\mathcal{E}_A \cup \mathcal{E}_C \cup \mathcal{E}_D \cup \mathcal{E}_H$), we get that for a random graph in $\mathcal{G}_{n,1/2}$, with probability at least $1 - 1/n$, it is possible to assign node labels and design an interval routing scheme in such a way that for every node v_0, there is a single interval on every edge except at most $O(\log^3 n)$ edges with two intervals each. (Spreading the problematic nodes so that each adds an interval to a different edge - is done by a greedy process similar to those of sections 2.1 and 3.)

Theorem 4. *For sufficiently large n, with probability at least $1-1/n$, a random graph $G \in \mathcal{G}_{n,1/2}$ can be given an assignment of node labels and a shortest path interval routing scheme (polynomial time constructible) using a single interval per edge, except for at most $O(\log^3 n)$ edges per vertex where two intervals must be used.* □

Corollary 2. *For almost every n-node graph G there exists an assignment of node labels from the set $\{1, \ldots, n\}$, and a shortest path routing scheme using at most $n + O(\log^4 n)$ bits of memory per vertex. Moreover the routing scheme is constructible in polynomial time.*

Proof. Assume G satisfies Theorem 4. The interval routing scheme on G can be implemented in each vertex by a table of $O(\log^3 n)$ integers, and a binary vector of n bits. Indeed, every 1-IRS can be implemented in each vertex in $n + O(\log n)$ bits (cf. [6]). Note that the $O(\log^3 n)$ problematic nodes contribute for at most $O(\log^3 n)$ single intervals, each one composed of exactly one node label. We store in a table of $O(\log^3 n)$ entries the label of these nodes, and the output port number that makes an overhead of $O(\log^4 n)$ bits in total. These nodes are treated as exceptions, and checked first in the routing process. So, the label of these nodes can be merged to the remaining intervals in order to simulate a 1-IRS. □

References

1. N. Alon and J. H. Spencer, *The Probabilistic Method*, John Wiley & Sons, 1992.
2. B. Bollobás, *Random Graphs*, Academic Press, New York, 1975.
3. H. Buhrman, J.-H. Hoepman, and P. Vitányi, *Optimal routing tables*, in 15^{th} Annual ACM Symposium on Principles of Distributed Computing (PODC), May 1996.
4. H. Buhrman, M. Li, and P. Vitányi, *Kolmogorov random graphs and the incompressibility method*, in IEEE Conference on Compression and Complexity of Sequences, IEEE Comp. Soc. Press, 1997.
5. M. Flammini, J. van Leeuwen, and A. Marchetti-Spaccamela, *The complexity of interval routing on random graphs*, in 20^{th} International Symposium on Mathematical Foundations of Computer Sciences (MFCS), J. Wiederman and P. Hájek, eds., vol. 969 of Lecture Notes in Computer Science, Springer-Verlag, Aug. 1995, pp. 37–49.
6. C. Gavoille, *A survey on interval routing scheme*, Research Report RR-1182-97, LaBRI, University of Bordeaux, 351, cours de la Libération, 33405 Talence Cedex, France, Oct. 1997. Submitted for publication.
7. C. Gavoille and D. Peleg, *The compactness of interval routing*, Research Report RR-1176-97, LaBRI, University of Bordeaux, 351, cours de la Libération, 33405 Talence Cedex, France, Sept. 1997. Submitted for publication.
8. C. Gavoille and S. Pérennès, *Memory requirement for routing in distributed networks*, in 15^{th} Annual ACM Symposium on Principles of Distributed Computing (PODC), ACM PRESS, ed., May 1996, pp. 125–133.
9. N. Santoro and R. Khatib, *Labelling and implicit routing in networks*, The Computer Journal, 28 (1985), pp. 5–8.
10. J. van Leeuwen and R. B. Tan, *Interval routing*, The Computer Journal, 30 (1987), pp. 298–307.