1-1-2010

Local versus Global Search in Channel Graphs

A.H. Hunter
University of Washington - Seattle Campus

Nicholas Pippenger
Harvey Mudd College

Local versus Global Search in Channel Graphs

A. H. Hunter
ahh@cs.washington.edu
Department of Computer Science & Engineering
University of Washington
AC101 Paul G. Allen Center, Box 352350
185 Stevens Way
Seattle, WA 98195-2350

Nicholas Pippenger
njp@math.hmc.edu
Department of Mathematics
Harvey Mudd College
1250 Dartmouth Avenue
Claremont, CA 91711

Abstract: Previous studies of search in channel graphs has assumed that the search is *global*; that is, that the status of any link can be probed by the search algorithm at any time. We consider for the first time *local* search, for which only links to which an idle path from the source has already been established may be probed. We show that some well known channel graphs may require exponentially more probes, on the average, when search must be local than when it may be global.

Keywords: Circuit switching, path search, search algorithms, decision trees.

1. Introduction

A *channel graph* is an acyclic directed graph $G = (V, E)$, with *vertices* V and *edges* E, in which there exist a *source* vertex $s \in V$ and a *target* vertex $t \in V$ such that every vertex lies on a directed path from s to t. (Such a source and target, if they exist, are clearly unique.) The vertices other than the source and target are called *links*.

A *state* of a channel graph is an assignment of a *status* (*busy* or *idle*) to each link of the graph. We shall extend such an assignment to all vertices by agreeing that the source and target are always idle. We shall deal in this paper with a particular probability distribution on the states of a channel graph. We choose a real number q, in the range $0 \leq q \leq 1$, which we call the *vacancy probability*; its complement $p = 1 - q$ is called the *occupancy probability*. We then define a random state of a channel graph to be one in which each link is independently idle with probability q (and thus busy with probability p). This probability distribution on states was introduced independently by Lee [L1] and Le Gall [L2, L3].

We shall say that a channel graph is *linked* in a given state if there exists a directed path from the source to the target consisting entirely of idle links. We shall say that a channel graph is *blocked* in a given state if there exists a cut between the source and the target consisting entirely of busy links. (Clearly, a channel graph in a given state is either linked or blocked, but not both.) If a channel graph G is in a random state with vacancy probability q, the *linking probability* will be denoted $Q(G, q)$, and the complementary *blocking probability* will be denoted $P(G, q) = 1 - Q(G, q)$.

Consider now a search algorithm that seeks to determine whether a known channel graph, in an unknown random state with a known vacancy probability, is linked or blocked. The algorithm gathers information about the state of the graph by sequentially probing the status of links until all the links of either an idle path or a busy cut have been probed. (The algorithm may be adaptive, so that the decision as to which link to probe at a given step may depend on the outcomes of all previous probes.)

Such an algorithm may be modeled as a decision tree. The elements of such a tree will be called *nodes* and *arcs* (to distinguish them from the vertices and edges of the channel graph). Each node is either a *probe node*, in which case it is labeled with the name of a link in the channel graph and has two outgoing arcs (one labeled "idle" and one labeled "busy") leading to other nodes, or a *leaf*, in which case it is labeled with one of the two possible outcomes ("linked" or "blocked") and has no outgoing arcs. There is a distinguished probe node, called the *root*, that has no incoming arcs; every other node has exactly one incoming arc. Execution of the algorithm begins at the root and proceeds in an obvious way, probing links and following the appropriate arcs, until it arrives at a leaf that announces the final result. There is an obvious notion of such an algorithm being correct: every trajectory from the root to a leaf labeled "linked" probes every link on a path from the source to the target in the channel graph and departs each of these probe nodes along the "idle" arc, and every trajectory from the root to a leaf labeled "blocked" probes every link on a cut between the source and the target in the channel graph and departs each of these probe nodes along the "busy" arc. This model of a search algorithm was introduced by Lin and Pippenger [L4], and subsequently used by Pippenger [P]. We shall denote by $E(G, q)$ the minimum possible expected number of probes performed by any search algorithm that correctly searches the channel graph G with vacancy probability q. This formulation of the search problem considers algorithms to be deterministic, with only the state of the network being random. It is clear, however, that a formulation allowing randomized algorithms would yield the same function $E(G, q)$: for a randomized algorithm may be regarded as a convex combination of deterministic algorithms (that is, as a probability distribution on decision trees, obtained by making all random choices at the outset), and at

least one of these algorithms must use make an expected number of probes that is at most as large as that of the convex combination.

In the definition of "search algorithm" given above, any node in the decision tree may probe any link in the channel graph, regardless of previous probes and their outcomes. We shall call this situation *global* search. This model is appropriate for circumstances in which all parts of the network being searched are directly accessible by the computer performing the search. It will be unrealistic, however, if the probes themselves require communication over the network being searched. An alternative appropriate to the latter circumstances is to allow a link to be probed only if it is *accessible*, meaning that all the links on some path from the source to that link have previously been probed and found to be idle. This situation will be called *local* search.

We shall denote by $E_1(G, q)$ the minimum possible expected number of probes performed by any local search algorithm that correctly searches the channel graph G with vacancy probability q. We clearly have $E(G, q) \leq E_1(G, q)$, and the main question explored in this paper is: how much larger than $E(G, q)$ can $E_1(G, q)$ be? We shall see that the answer is: for some channel graphs G, and some values of the vacancy q, it can be exponentially larger. Preliminary versions of the results in this paper appeared in the first author's thesis [H2].

The graphs we shall study are called fully parallel graphs. Let T_k be a complete binary tree of depth k, with root r and 2^k leaves, and with all edges directed from the root towards the leaves. Let T'_k be a similar tree of depth k, with root r' and 2^k leaves, and with all edges directed from the leaves towards the root. The *fully parallel* channel graph F_k is obtained by joining each leaf of T_k to the corresponding leaf of T'_k by an edge directed from the former to the latter. The source of F_k is $s = r$ and the target is $t = r'$. The vertices of F_k will be partitioned into *ranks*: the vertices at depth j $(0 \leq j \leq k)$ of T_k will constitute rank j; the vertices at depth j $(0 \leq j \leq k)$ of T'_k will constitute rank $2k + 1 - j$. Thus the source constitutes rank 0, and the target constitutes rank $2k + 1$. An example is shown in Figure 1.

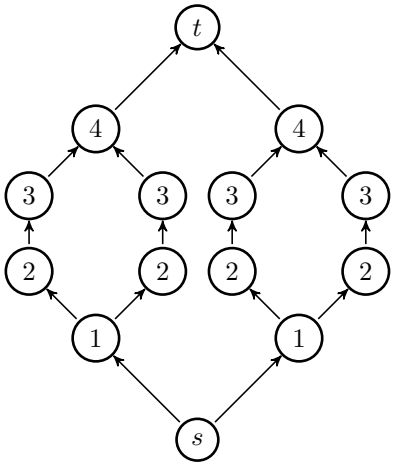

Figure 1. The channel graph F_2. Links are annotated with their ranks.

Our results in this paper will show that global search of F_k can be performed with an expected number of probes linear in k for any fixed q. We shall show, however, that local search of F_k requires an expected number of probes exponential in k for $1/2 < q < 1$, and we shall determine the precise rate of exponential growth for q in this range.

Specifically, we shall present in Section 2 an algorithm `bilat-search` that performs global search in F_k using an expected number of probes at most $4k$ for any fixed k and for all sufficiently large k. (For $0 \le q < 1/2$, the expected number of probes is in fact bounded as $k \to \infty$, but the bound depends on q. Because we are primarily interested in the contrast between linear and exponential growth as a function of k, we shall in what follows frequently replace constants that depend on q with factors of k or $1/k$. This replacement will weaken our results slightly, but simplify them greatly, without affecting the contrast between linear and exponential growth, or between different rates of exponential growth.) We shall also present an algorithm `unilat-search` that performs local search in F_k using an expected number of probes at most $4k \max\{1, \min\{(2q)^k, 1/q^k\}\}$ for any fixed q and for all sufficiently large k. The expected number of probes for this algorithm thus grows linearly for $0 \le q \le 1/2$ and for $q = 1$, but grows exponentially in k for $1/2 < q < 1$, with a base that increases as $2q$ from 1 to $\sqrt{2}$ as q increases from $1/2$ to $1/\sqrt{2}$, then decreases as $1/q$ from $\sqrt{2}$ to 1 as q increases from $1/\sqrt{2}$ to 1. In Section 3, we shall present lower bounds that show that this rate of exponential growth is the best possible. Specifically, we shall show that any algorithm performing local search in F_k must use an expected number of probes at least $1/kq^k$ for $1/\sqrt{2}$ and for all sufficiently large k, and at least $(2q)^k/k^6$ for $1/2 < q < 1/\sqrt{2}$ and for all sufficiently large k. (The factors $1/k$ and $1/k^6$ are consequences of our desire to keep our proofs as simple as possible. We in fact conjecture that the algorithm `unilat-search` optimal not just in its rate of exponential growth, but in the stronger sense of using an expected number of probes *exactly* $E_1(F_k, q)$.)

2. Upper Bounds

In this section we shall obtain upper bounds to $E(F_k, q)$ and $E_1(G, q)$ by presenting and analyzing natural path search algorithms. The upper bound for the global case is actually a special case of a result by Lin and Pippenger [L4], but we shall give a simpler proof that can be adapted to the local case.

The results in this section and the next depend on results that are well known in the theory of branching processes (see Harris [H1]). A *branching process* is a random process that begins with one individual in generation zero, and in which each individual independently contributes an identically distributed random number of offspring to the next generation. Let Z_l denote the number of individuals in the l-th generation. If $f(x)$ is the generating function for the number of offspring contributed by an individual, then the generating function for Z_l is the l-th iterate $f^{(l)}(x)$ of $f(x)$, defined by $f^{(0)}(x) = x$ and $f^{(l+1)}(x) = f(f^{(l)}(x)) = f^{(l)}(f(x))$.

In this section we shall be concerned with the branching process for which the generating function for the number of offspring of an individual is $f(x) = (1 - q^2 + q^2 x)^2$, describing the number of successes in two independent trials that each succeed with probability q^2. This branching process governs the blocking probability $P(F_k, q)$ in the following way. Any path from the source to the target in F_k that passes through a link v in T_k also passes through the link v' in T'_k, and conversely. Consider the tree T^*_k obtained from F_k by identifying each vertex of T_k with the corresponding vertex of T'_k. The source and target of F_k are identified to form the root of T^*_k, which we take to be idle. Let every other vertex of T^*_k be idle if and only if the corresponding links in T_k and T'_k are both idle (which occurs with probability q^2) and busy if and only if either of the corresponding links in T_k and T'_k are busy (which occurs with probability $1 - q^2$). Then it is clear that $P(F_k, q)$ is equal to the probability that every path from the root to a leaf in T^*_k contains at least

one busy vertex. This probability is just the probability that $Z_k = 0$, which is the constant term $f^{(k)}(0)$ in $f^{(k)}(x)$. Thus we have the recurrence

$$P(F_k, q) = f(P(F_{k-1}, q)) \tag{2.1}$$

for $k \geq 1$, with the initial condition $P(F_0, q) = 0$. We have $P(F_k, q) \geq P(F_{k-1})$ for $k \geq 1$, since $Z_{k-1} = 0$ implies $Z_k = 0$. Thus the sequence $P(F_k, q)$ is non-decreasing in k. Since it is also bounded above by 1, it tends to a limit, which we shall denote P^*, and we have $P(F_k, q) \leq P^*$ for all $k \geq 0$. Letting k tend to infinity in (2.1), and noting that $f(x)$ is continuous, we see that P^* must be a fixed-point of $f(x)$, specifically the smallest fixed-point greater than or equal to 0. Solving the quadratic equation $f(P^*) = P^*$, we find that P^* is equal to 1 for $0 \leq q \leq 1/\sqrt{2}$, and equal to $(1 - q^2)^2/q^4$ for $1/\sqrt{2} \leq q \leq 1$. Thus we have proved the following lemma.

Lemma 2.1: For any $0 \leq q \leq 1$ and $k \geq 0$, we have

$$P(F_k, q) \leq \begin{cases} 1, & \text{if } 0 \leq q \leq 1/\sqrt{2}; \\ (1 - q^2)^2/q^4, & \text{if } 1/\sqrt{2} \leq q \leq 1. \end{cases}$$

(Note that the two cases agree for $q = 1/\sqrt{2}$.)

Our upper bound for global path search in F_k is based on the following recursive algorithm. It should be clear how, for a given value of $k \geq 0$, it may be transformed into a decision tree of the form described in the introduction. The algorithm has been written to return `true` if its argument is linked and `false` if it is blocked. It would be straightforward to add data structures that would allow it to return an idle path or busy cut as appropriate. Since our main interest is in the cost of the search, however, we have only kept track of enough information to determine the sequence, and thus the number, of probes performed by the algorithm.

```
bilat-search(F_k :  G):
if  k = 0:
  return true
let F_{k-1}:  G', G'' be the two copies of F_{k-1} in G;
if idle(source(G')) and idle(target(G')) and bilat-search(G'):
  return true
if idle(source(G'')) and idle(target(G'')) and bilat-search(G''):
  return true
return false
```

We assume short-circuiting conjunctions; that is, if `f()` evaluates to false, `f() and g()` is false and `g()` is not evaluated.

From the algorithm we can easily write a recurrence for the expected number of probes it performs, which leads to the following recurrence for $E(F_k, q)$. We have

$$E(F_k, q) \leq 1 + q + q^2 E(F_{k-1}, q) + (p + qp + q^2 P(F_{k-1}, q))(1 + q + q^2 E(F_{k-1}, q))$$

$$= (1 + p + qp + q^2 P(F_{k-1}, q))(1 + q + q^2 E(F_{k-1}, q)) \tag{2.2}$$

for $k \geq 1$, with the initial condition $E(F_0, q) = 0$. In the right-hand side of the first line, the initial terms $1 + q + q^2 E(F_{k-1}, q)$ represents the expected cost of searching the first copy of F_{k-1}, the first expression in parentheses represents the probability that no path through the first copy is found, so that the second copy must be searched, and the second expression in parentheses (which is equal to the initial terms) represents the expected cost of searching the second copy. In the second line, we have factored out the expected cost of searching a copy, so the first expression in parentheses now represents the expected number of copies that must be searched. Using the bound from Lemma 2.1, we obtain

$$\begin{aligned} 1 + p + qp + q^2 P(F_{k-1}, q) &\leq 1 + p + qp + q^2 P^* \\ &= \begin{cases} 2, & \text{if } 0 \leq q \leq 1/\sqrt{2}; \\ 1/q^2, & \text{if } 1/\sqrt{2} \leq q \leq 1. \end{cases} \end{aligned} \tag{2.3}$$

Substituting the bounds (2.3) into the recurrence (2.2), we obtain

$$E(F_k, q) \leq \begin{cases} 2(1+q) + 2q^2 E(F_{k-1}, q), & \text{if } 0 \leq q \leq 1/\sqrt{2}; \\ (1+q)/q^2 + E(F_{k-1}), & \text{if } 1/\sqrt{2} \leq q \leq 1 \end{cases}$$

for $k \geq 1$. Applying the bounds $2(1+q) \leq 4$ and $2q^2 \leq 1$ for $0 \leq q \leq 1/\sqrt{2}$ and $(1+q)/q^2 \leq 4$ for $1/\sqrt{2} \leq q \leq 1$ yields the recurrence

$$E(F_k, q) \leq 4 + E(F_{k-1}, q).$$

This recurrence, together with the initial condition $E(F_0, q) = 0$, gives the following theorem.

Theorem 2.2: For any $0 \leq q \leq 1$ and $k \geq 0$, we have

$$E(F_k, q) \leq 4k.$$

To obtain an upper bound for local search, we transform the above algorithm as follows.

```
unilat-search(F_k : G):
if k = 0:
  return true
let F_{k-1}: G', G'' be the two copies of F_{k-1} in G;
if (idle(source(G')) and unilat-search(G')) and idle(target(G'))
  return true
if (idle(source(G'')) and unilat-search(G'')) and idle(target(G''))
  return true
return false
```

The difference between these algorithms lies in the order of the conditions in the conjunction `(... and ... and ...)`. This change is necessary to keep the algorithm local; we may not probe `target(G')` until we have successfully found a path through `G'`, using a recursive call. Thus, the algorithm is much more

likely to make such a recursive call (with probability q instead of q^2) and its running time will grow much more quickly. As before, we may read off the recurrence

$$E_1(F_k, q) \leq (1 + p + qp + q^2 P(F_{k-1}, q))(1 + qE_1(F_{k-1}, q) + qQ(F_{k-1}, q))$$

for $k \geq 1$, with the initial condition $E_1(F_0, q) = 0$. Using the bounds $Q(F_{k-1}, q) \leq 1$ and $P(F_{k-1}, q) \leq P^*$ together with Lemma 2.1 as before, we obtain

$$E_1(F_k, q) \leq \begin{cases} 4 + 2qE(F_{k-1}, q), & \text{if } 0 \leq q \leq 1/\sqrt{2}; \\ 2/q^2 + (1/q)E(F_{k-1}), & \text{if } 1/\sqrt{2} \leq q \leq 1 \end{cases}$$

for $k \geq 1$. Applying the bounds $2q \leq 1$ for $0 \leq q \leq 1/2$ and $2/q^2 \leq 4$ for $1/\sqrt{2} \leq q \leq 1$ yields the recurrence

$$E_1(F_k, q) \leq \begin{cases} 4 + E(F_{k-1}, q), & \text{if } 0 \leq q \leq 1/2; \\ 4 + 2qE(F_{k-1}, q), & \text{if } 1/2 \leq q \leq 1/\sqrt{2}; \\ 4 + (1/q)E(F_{k-1}), & \text{if } 1/\sqrt{2} \leq q \leq 1 \end{cases}$$

for $k \geq 1$. This recurrence, together with the initial condition $E_1(F_0, q) = 0$ yields the following theorem.

Theorem 2.3: For any $0 \leq q \leq 1$ and $k \geq 0$, we have

$$E_1(F_k, q) \leq \begin{cases} 4k, & \text{if } 0 \leq q \leq 1/2; \\ 4k(2q)^k, & \text{if } 1/2 \leq q \leq 1/\sqrt{2}; \\ 4k/q^k, & \text{if } 1/\sqrt{2} \leq q \leq 1 \end{cases}$$

$$= 4k \max\{1, \min\{(2q)^k, 1/q^k\}\}.$$

3. Lower Bounds

In this section, we shall prove the following theorem.

Theorem 3.1: For $1/2 < q < 1$, we have

$$E_1(F_k, q) \geq \min\{(2q)^k/k^6, 1/kq^k\}$$

for all sufficiently large k. This result will show that the exponential rates of growth in Theorem 2.3 are the best possible.

To prove Theorem 3.1, we consider an optimal algorithm $\mathcal{T}$ for local search in F_k with vacancy probability q, and define the random variable T to be the number of probes used by $\mathcal{T}$. Since $\mathcal{T}$ is optimal, we have $\mathrm{Ex}[T] = E_1(F_k, q)$, so it will suffice to show that $\mathrm{Ex}[T]$ satisfies the lower bound of Theorem 3.1.

Our lower bounds will be based on the following principle. If, at any point during the execution of the algorithm, we give the algorithm, at no cost, some information that it has not asked for, that gift can only decrease the expected number of probes it needs to make to complete its task. This principle can be formalized in terms of decision trees in the following way. We shall transform the original decision tree $\mathcal{T}$ into a modified decision tree $\mathcal{T}^*$. The tree $\mathcal{T}^*$ will contain, in addition to internal nodes that make probes,

internal *gift* nodes that branch according to the information given for free. We shall prune the tree below each gift node, eliminating probe nodes whose outcome is determined by the gift nodes above them. It is easy to show that this transformation preserves correctness and can only decrease the expected number of probes used: any state of the network corresponds to a path π from the root to a leaf L in $\mathcal{T}$, and a path π^* from the root to a leaf L^* in $\mathcal{T}^*$; the leaves L and L^* will have the same label ("linked" or "blocked"), and any link of the network probed by node on π^* will also be probed by a node on π. We shall not dwell further on this process of formalization; instead we shall proceed directly to an informal presentation of our lower bounds based on this principle.

We shall begin by telling the algorithm, before it begins its execution, the status of all accessible links in ranks 1 through k, thereby informing it of which links in rank $k+1$ are accessible. The accessible links in rank $k+1$ will be called *candidate* links. The first probe by the algorithm will thus be to a candidate link. Whenever the algorithm probes a candidate link, we shall tell it the status of all accessible links on the path from that candidate link to the target. It follows that every probe by the algorithm will be to a candidate link. This process is illustrated in Figure 2. Each such probe either reveals an idle path from the probed candidate link to target (in which case the algorithm can announce "linked"), or discovers a busy link that blocks the paths from some subset of the candidate links to the target (in which case the algorithm need not probe these candidate links, since probing them could neither reveal an idle path to the target, nor discover a busy node that blocks any additional paths from candidate links to the target). The algorithm will thus probe a sequence of candidate links until it either reveals an idle path to the target or discovers a set of busy links that together block the paths from all candidate links to the target. It is clear that an optimal algorithm will never make a probe after the outcome of the search ("linked" or "blocked") has been determined, nor will it probe a candidate link after a busy link has already be discovered on the path from that candidate link to the target.

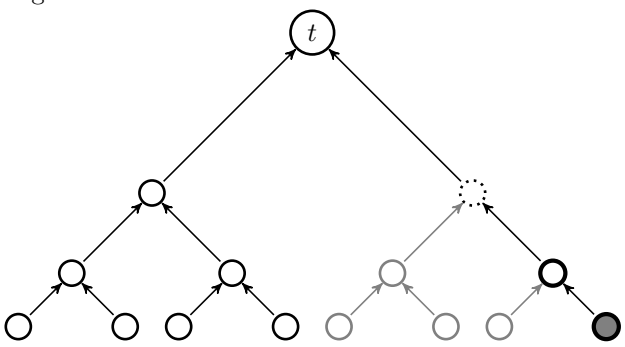

Figure 2. After probing the marked candidate link (gray), discovering the status of all accesible links above it (heavy for idle, dotted for busy), and pruning now-useless links (light), only candidate links are accessible.

We can now prove the easier case, $1/\sqrt{2} < q < 1$, of Theorem 3.1. Let I_t be the event "$T \geq t$" (so that the algorithm performs a t-th probe of a candidate link), and let J_t be the event "the t-th probe reveals an idle path to the target" (so that the algorithm announces "linked" after the t-th probe). We have $\Pr[J_t \mid I_t] = q^k$ for any probe of a candidate link (since the path from any probed candidate link to the

target contains k links, each of which is independently idle with probability q). Furthermore, by Lemma 2.1 we have $Q(F_k, q) = 1 - P(F_k, q) \geq 1 - (1-q^2)^2/q^4 \geq 1/k$ for all sufficiently large k. Thus we have

$$\begin{aligned}
\mathrm{Ex}[T] &= \sum_{t \geq 1} \Pr[I_t] \\
&= \frac{1}{q^k} \sum_{t \geq 1} \Pr[J_t \mid I_t] \Pr[I_t] \\
&= \frac{1}{q^k} Q(F_k, q) \\
&\geq \frac{1}{kq^k}
\end{aligned}$$

for all sufficiently large k. This proves Theorem 3.1 for the case $1/\sqrt{2} < q < 1$.

For the remaining case, $1/2 < q \leq 1/\sqrt{2}$, we shall need to work with an additional branching process $Y_0, Y_1, \ldots Y_l, \ldots$, for which $Y_0 = 1$ and the generating function for the number of offspring of an individual is $g(x) = (1 - q + qx)^2$, describing the number of successes in two independent trials that each succeed with probability q. This branching process governs the number of accessible links in the successive ranks of T_k: the number of links in rank $1 \leq l \leq k$ that are idle and accessible through an idle path from the source is Y_l. In particular the set of candidate links has cardinality Y_k. Let A denote the event "$Y_k < (2q)^k/2k^2$".

Lemma 3.2: For fixed $1/2 < q \leq 1$, we have

$$\Pr[A] \leq 1 - \frac{5}{k}$$

for all sufficiently large k.

Proof: Let $M = (2q)^k/2k^2$. Then for any $x \leq 1$ we have

$$\begin{aligned}
\Pr[A] &= \sum_{0 \leq m < M} \Pr[Y_k = m] \\
&\leq \frac{1}{x^M} \sum_{0 \leq m < M} \Pr[Y_k = m]\, x^m \\
&\leq \frac{1}{x^M} \sum_{0 \leq m \leq 2^k} \Pr[Y_k = m]\, x^m \\
&= \frac{1}{x^M}\, g^{(k)}(x).
\end{aligned}$$

Let $\delta = k/(2q)^k$. Then taking $x = 1 - \delta$ yields

$$\Pr[A] \leq \frac{1}{(1-\delta)^M}\, g(k)(1-\delta).$$

We bound the first factor by using the inequality $(1-\delta)^M \geq 1 - M\delta$ (which holds for $0 \leq \delta \leq 1$ and $M \geq 1$), obtaining

$$\begin{aligned}
\frac{1}{(1-\delta)^M} &\leq \frac{1}{1 - (1/2k)} \\
&= 1 + \frac{1}{2k-1}.
\end{aligned}$$

It will now suffice to show that

$$g^{(k)}(1-\delta) \le 1 - \frac{6}{k}, \tag{3.1}$$

for then we shall have

$$\begin{aligned}\Pr[A] &\le \left(1 + \frac{1}{2k-1}\right)\left(1 - \frac{6}{k}\right) \\ &\le 1 - \frac{6}{k} + \frac{1}{2k-1} \\ &\le 1 - \frac{5}{k},\end{aligned}$$

completing the proof of the lemma.

The function $g(x)$ has two fixed points, at $x = 1$ and at $x = (1-q)^2/q^2 < 1$. Let $\gamma = (1-q)^2/q^2$ denote this smaller fixed point. Define $G(y) = \big(g\big(\gamma + (1-\gamma)y\big) - \gamma\big)/\big(1-\gamma\big) = 2(1-q)y + (2q-1)y^2$. Since the transformation $y \mapsto \gamma + (1-\gamma)y$ is inverse to $x \mapsto (x-\gamma)/(1-\gamma)$, iterating $G(y)$ is equivalent to iterating $g(x)$: $g^{(k)}(x) = \gamma + (1-\gamma)G^{(k)}\big((x-\gamma)/(1-\gamma)\big)$. Next define $H(y) = y/\big(2q + (1-2q)y\big)$. Then $G(y) \le H(y)$ for all $y \ge 0$, since $\big(2(1-q)y + (2q-1)y^2\big)\big(2q + (1-2q)y\big) - y = -y(2q-1)^2(y-1)^2 \le 0$. Straightforward induction shows that

$$H^{(k)}(y) = \frac{y}{(2q)^k + (1-(2q)^k)y}.$$

Substituting $x = 1 - \delta$ in $y = (x-\gamma)/(1-\gamma)$, we obtain $y = (1-\delta-\gamma)/(1-\gamma) < 1 - \delta$. From this we obtain, since $H(y)$ and thus $H^{(k)}(y)$ are increasing functions of y for $0 \le y \le 1$,

$$\begin{aligned}H^{(k)}(y) &= \frac{1-\delta}{(2q)^k + (1-(2q)^k)(1-\delta)} \\ &= \frac{1-\delta}{(1-\delta) + \delta(2q)^k} \\ &\le \frac{1-\delta}{(1-\delta) + k} \\ &\le \frac{1}{1+k}.\end{aligned}$$

Since $G^{(k)}(y) \le H^{(k)}(y)$, we obtain

$$G^{(k)}(y) \le \frac{1}{1+k},$$

so we have

$$\begin{aligned}g^{(k)}(1-\delta) &= \gamma + (1-\gamma)G^{(k)}(y) \\ &\le \gamma + \frac{1-\gamma}{1+k}.\end{aligned}$$

Since $\gamma < 1$, this inequality establishes (3.1), and thus completes the proof of the lemma. □

For each link v in rank $k-l$ of F_k for some $0 \le l \le k-1$, let X_v denote the number of links in rank k that are accessible through idle paths from v. For each link v in rank $k-l$, let B_v be the event $X_v > k^2(2q)^{l}$", and let B be the event "for some l in the range $0 \le l \le k-1$ and some link v in rank $k-l$, $X_v > k^2(2q)^{l}$".

Lemma 3.3: For fixed $1/2 < q \le 1$, we have

$$\Pr[B] \le \frac{1}{k}$$

for all sufficiently large k.

Proof: There are at most $2 \cdot 2^k$ links in T_k. Thus it will suffice to prove that for any one link v in rank $k-l$,

$$\Pr[B_v] \leq \frac{4}{e^k}, \tag{3.2}$$

for then, by Boole's inequality, we shall have

$$\begin{aligned}\Pr[B] &\leq \sum_v \Pr[B_v] \\ &\leq \frac{8 \cdot 2^k}{e^k} \\ &\leq \frac{1}{k}\end{aligned}$$

for all sufficiently large k.

Consider a fixed link v in rank $k-l$. Let $M = k^2(2q)^l$. Then for any $x \geq 1$ we have

$$\begin{aligned}\Pr[B_v] &= \sum_{M<m\leq 2^l} \Pr[X_v = m] \\ &\leq \frac{1}{x^M} \sum_{M<m\leq 2^l} \Pr[X_v = m]\, x^m \\ &\leq \frac{1}{x^M} \sum_{0\leq m\leq 2^l} \Pr[X_v = m]\, x^m \\ &\leq \frac{1}{x^M} g^{(l)}(x).\end{aligned}$$

Let $\delta = 1/k(2q)^l$. Then taking $x = 1+\delta$ yields

$$\Pr[B_v] \leq \frac{1}{(1+\delta)^M}\, g^{(l)}(1+\delta).$$

We bound the first factor by using the inequality $\log(1+z) \geq z - z^2/2$ (which holds for all $z \geq 0$), obtaining

$$\begin{aligned}\frac{1}{(1+\delta)^M} &\leq \exp\left(-M\log(1+\delta)\right) \\ &\leq \exp\left(-M(\delta+\delta^2/2)\right) \\ &= \exp\left(-k + 1/2(2q)^l\right) \\ &\leq \exp\left(-k+1/2\right) \\ &= \frac{e^{1/2}}{e^k} \\ &\leq \frac{2}{e^k},\end{aligned} \tag{3.3}$$

for all sufficiently large k. It will now suffice to show that

$$g^{(l)}(1+\delta) \leq 2, \tag{3.4}$$

for all sufficiently large k, for this combined with (3.3) will yield (3.2), completing the proof of the lemma.

Let $G(\delta) = g(1+\delta) - 1 = 2q\delta + q^2\delta^2$, and let $H(\delta) = 2q\delta/(1-\delta)$. Then $G(\delta) \le H(\delta)$ for all $0 \le \delta < 1$, since $(2q\delta + q^2\delta^2)(1-\delta) = 2q\delta - (2-q)q\delta^2 - q^2\delta^3 \le 2q\delta$. Since $G(\delta)$ and $H(\delta)$ are both nondecreasing functions of δ for $0 \le \delta < 1$, a straightforward induction on $l \ge 0$ shows that

$$\begin{aligned} G^{(l)}(\delta) &\le H^{(l)}(\delta) \\ &\le \frac{(2q)^l \delta}{1 - l(2q)^{l-1}\delta} \end{aligned}$$

for $0 \le \delta < 1/l(2q)^{l-1}$. Taking $\delta = 1/k(2q)^l$, we obtain

$$\begin{aligned} g^{(l)}(1+\delta) &= 1 + G^{(l)}(\delta) \\ &\le 1 + \frac{1/k}{1 - 1/(2q)} \\ &\le 2 \end{aligned}$$

for all sufficiently large k. This inequality establishes (3.4), and thus completes the proof of the lemma. □

Let C be the event "neither A nor B occurs". From Lemmas 2.2 and 2.3 we have

$$\begin{aligned} \Pr[C] &\ge 1 - \Pr[A] - \Pr[B] \\ &\ge 1 - \left(1 - \frac{5}{k}\right) - \frac{1}{k} \\ &= \frac{4}{k}. \end{aligned} \tag{3.5}$$

As before, let I_t denote the event "$T \ge t$". If the t-th probe of a candidate link discovers l accessible idle links on the path from the candidate link to the target (before reaching a busy link or the target), we shall associate with with that probe a "payoff" $K_t = k^2\,(2q)^l$. The number L_t of accessible idle links discovered on the path from the probed candidate link to the target has the distribution $\Pr[L_t = l \mid I_t] = (1-q)q^l$ for $0 \le l \le k-1$ and $\Pr[L_t = k \mid I_t] = q^k$, and thus the payoff K_t of the t-th probe has the distribution $\Pr[K_t = k^2\,(2q)^l \mid I_t] = (1-q)q^l$ for $0 \le l \le k-1$ and $\Pr[K_t = k^2\,(2q)^k \mid I_t] = q^k$. We thus have

$$\begin{aligned} \mathrm{Ex}[K_t \mid I_t] &\le k^2 \sum_{0 \le l \le k} (2q)^l\, q^l \\ &\le 2k^3, \end{aligned}$$

since $2q^2 \le 1$. We note that these distributions and expectations are independent of A, B and C.

Let $K = \sum_{y \ge 1} K_t$ denote the total payoff from all probes. If C occurs, and after T probes of candidate links the algorithm announces "linked" or "blocked", the total payoff from all probes of candidate links must satisfy $K \ge (2q)^k/2k^2$. For if the algorithm announces "blocked", then (because A did not occur) there must have been at least $(2q)^k/2k^2$ candidate links, and (because B did not occur) the number of candidate links whose paths to the target were found to be blocked by the t-th probe was at most the payoff K_t (since the candidate links whose paths to the target blocked by a busy link in rank $k+1+l$ are accessible through idle paths from a link in rank $k-l$). And if the algorithm announced "linked", the T-th probe must have revealed an idle path from a candidate link to the target, and this probe alone had payoff $K_T = k^2\,(2q)^k \ge (2q)^k/2k^2$.

Thus we have

$$\begin{aligned}
\mathrm{Ex}[T] &\geq \mathrm{Ex}[T \mid C]\ \Pr[C] \\
&\geq \frac{1}{k}\mathrm{Ex}[T \mid C] \\
&\geq \frac{1}{k}\sum_{t\geq 1}\Pr[I_t \mid C] \\
&\geq \frac{1}{k}\,\frac{1}{2k^3}\sum_{t\geq 1}\mathrm{Ex}[K_t \mid I_t]\ \Pr[I_t \mid C] \\
&\geq \frac{1}{k}\,\frac{1}{2k^3}\sum_{t\geq 1}\mathrm{Ex}[K \mid C] \\
&\geq \frac{1}{k}\,\frac{1}{2k^3}\,\frac{(2q)^k}{2k^2} \\
&= \frac{(2q)^k}{4k^6}.
\end{aligned}$$

This completes the proof of Theorem 3.1 in the case $1/2 < q \leq 1/\sqrt{2}$.

4. Conclusion

We have presented a sequence F_k of channel graphs for which global path search is easy, in the sense that its cost is $O(k)$ for any vacancy probability q, as is shown by the natural bilateral depth-first search algorithm presented in Section 2. We have shown in Section 3 that the cost of local path search in F_k is exponential in k for all $1/2 < q < 1$. We have also presented in Section 2 a natural algorithm, unilateral depth-first search, that shows that the exponential rate of growth in shown in Section 3 is the best possible, in the sense that it matches the exponential rate of growth of the cost of unilateral depth-first search. We conjecture that unilateral depth-first search is in fact optimal in the stronger sense of making the smallest possible expected number of probes for every $k \geq 0$ and $0 \leq q \leq 1$.

The foregoing results may be summarized by saying that global search in F_k is cheap, but local search is expensive. If, however, we consider "bilateral local search", wherein a link may be probed if either there is an idle path from the source to it, or an idle path from it to the target, then search in F_k is again cheap, because the global search algorithm `bilat-search` probes only vertices meeting one of these conditions.

Are there channel graphs in which global search is cheap, but even bilateral local search is expensive? The answer is "yes", as can be seen by considering the graph $F_k \circ F_k$ obtained by connecting two copies of F_k in series, with the target of the first copy identified with the source of the second copy to form a link u. Global search of $F_k \circ F_k$ is cheap: we first probe u; if it is idle, we then search the first copy of F_k; and if that copy is linked, we search the second copy. Thus

$$E(F_k \circ F_k, q) \leq 1 + q(1 + Q(F_k, q))E(F_k, q).$$

If, however, we condition on the link u being idle, and give this information for free to the search algorithm, then a path in $F_k \circ F_k$ is the concatenation of two paths, one in each copy of F_k, while a cut in $F_k \circ F_k$ must include a cut in at least one copy of F_k. Since F_k is symmetric under reversal, every link probed by a bilateral local search in $F_k \circ F_k$ can be probed in a "unilateral" local search of the first copy of F_k, or in a "reverse unilateral" local search of the second copy. Thus the lower bound of Section 3 for unilateral

search in F_k, when multiplied by the probability q that u is idle, becomes a lower bound for bilateral search in $F_k \circ F_k$. Thus even bilateral local search of $F_k \circ F_k$ is expensive.

5. Acknowledgment

The research reported in this paper was supported in part by Grant CCF 0646682 from the National Science Foundation.

6. References

[H1] T. E. Harris, *The Theory of Branching Processes*, Springer-Verlag, 1963.

[H2] A. H. Hunter, *Locality & Complexity in Path Search*, B. S. Thesis, Department of Mathematics, Harvey Mudd College, May 2009.

[L1] C. Y. Lee, "Analysis of Switching Networks", *Bell System Technical Journal*, 34 (1955) 1287–1315.

[L2] P. Le Gall, "Étude du blocage dans les systèmes de commutation téléphonique automatique utilisant des commutateurs électroniques du type crossbar", *Ann. des Télécomm.*, 11 (1956) 159–171; 180–194; 197.

[L3] P. Le Gall, ""Méthode de calcul de l'encombrement dans les systèmes téléphonique automatique à marquage", *Ann. des Télécomm.*, 12 (1957) 374–386.

[L4] G. Lin and N. Pippenger, "Routing Algorithms for Switching Networks with Probabilistic Traffic", *Networks*, 28:1 (1996) 21–29.

[P] N. Pippenger, "Upper and Lower Bounds for the Average-Case Complexity of Path Search", *Networks*, 33:4 (1999) 249–259.