

ON THE PROBABLE PERFORMANCE OF HEURISTICS FOR BANDWIDTH MINIMIZATION

Jonathan S. Turner*

Abstract. Most research in algorithm design relies on worst-case analysis for performance comparisons. Unfortunately, worst-case analysis does not always provide an adequate measure of an algorithm's effectiveness. This is particularly true in the case of heuristic algorithms for hard combinatorial problems. In such cases, analysis of the probable performance can yield more meaningful results and can provide insight leading to better algorithms. The problem of minimizing the bandwidth of a sparse symmetric matrix by performing simultaneous row and column permutations, is an example of a problem for which there are well-known heuristics whose practical success has lacked a convincing analytical explanation. A class of heuristics introduced by Cuthill and McKee in 1969, and referred to here as the *level algorithms*, are the basis for bandwidth minimization routines that have been widely used for over a decade. At the same time, it is easy to construct examples, showing that the level algorithms can produce solutions that differ from optimal by an arbitrarily large factor. This paper provides an analytical explanation for the practical success of the level algorithms, by showing that for random matrices having optimal bandwidth no larger than k , any level algorithm will produce solutions that differ from optimal by a small constant factor. The analysis also suggests another class of algorithms with better performance. One algorithm in this class is shown to produce solutions that are nearly optimal.

Keywords. bandwidth minimization problem, heuristic algorithms, approximation algorithms, probable performance, probabilistic analysis, NP-completeness

1. Introduction. Let M be a symmetric matrix and let k be the largest integer for which there is a non-zero entry $M[i, i + k]$; k is called the bandwidth of M . It is often possible to reduce the bandwidth of a matrix by performing simultaneous row and column permutations. Most common matrix operations can be performed more efficiently if the matrices are in small bandwidth form. The matrices can also be stored more efficiently in this form. The matrix bandwidth minimization problem is usually re-cast as a graph theory problem; for any matrix M , the graph corresponding to M has an edge joining vertices i and j if and only if $M[i, j]$ is non-zero.

Let $G = (V, E)$ be a graph with $V = \{1, 2, \dots, n\}$. A *layout* of G is a permutation on $\{1, 2, \dots, n\}$. Define the *bandwidth* of G with respect to a layout τ by $\phi_\tau(G) = \max_{\{u,v\} \in E} |\tau(u) - \tau(v)|$. The *bandwidth* of G is defined by $\phi(G) = \min_\tau \phi_\tau(G)$. The *bandwidth minimization problem* (for graphs) is to determine for a graph G and an integer k if $\phi(G) \leq k$. Papadimitriou [9] first showed that the bandwidth minimization problem is NP-complete. Garey, Graham, Johnson and Knuth [7] later strengthened this result, showing that the problem remains NP-complete when restricted to free binary trees. Several heuristic algorithms for bandwidth minimization were proposed in the late sixties and early seventies. More recently, Saxe [10] has found a dynamic programming algorithm which can determine if a graph has bandwidth k in time $O(n^{k+1})$ for any fixed value of k . Monien and Sudborough [8] showed how to reduce the time bound to $O(n^k)$. One of the most successful heuristic algorithms is one discovered by Cuthill and McKee [5] which is a member of a class

*Computer Science Department, Campus Box 1045, Washington University, St. Louis, Missouri 63130, UUCP:..!{ihnp4,seismo}!wucs!jst, CSNET:wucs!jst@seismo.ARPA

of algorithms which are referred to here as the *level algorithms*. An algorithm is classified as a level algorithm if for all graphs $G = (V, E)$ the layout τ produced by the algorithm satisfies

$$\forall u, v \in V \quad d(\tau^{-1}(1), u) < d(\tau^{-1}(1), v) \Rightarrow \tau(u) < \tau(v)$$

where $d(x, y)$ denotes the length of the shortest path connecting vertices x and y . The level algorithms are reasonably fast and have proved to be quite successful in practice. On the other hand, one can easily construct examples in which the ratio of the bandwidth of the layout produced by a level algorithm to the actual bandwidth of the graph is arbitrarily large. Consequently one must resort to probabilistic analysis to gain insight to their practical success.

Let $G = (V, E)$ be generated by the following random experiment.

- Let $V = \{1, 2, \dots, n\}$, $E = \emptyset$.
- For each $\{u, v\}$, $1 \leq u < v \leq n$, add the edge $\{u, v\}$ to E with probability p .

The probability distribution defined by this experiment is denoted $\mathbf{\Gamma}_n(p)$ and the notation $G \in \mathbf{\Gamma}_n(p)$ means that G is a random graph generated in this way. In §2 it is shown that for almost all $G \in \mathbf{\Gamma}_n(p)$, $\frac{n}{\phi(G)} \leq 1 + \epsilon$ when $p \geq (c \ln n)/n$ and $\epsilon > 0$, $c > 0$ are fixed. (We say that a property holds for *almost all* graphs if the probability of the property holding approaches one as the number of vertices gets large. This notion is often described by the phrase “in probability”.) Consequently, if τ is any layout at all, $\frac{\phi_\tau(G)}{\phi(G)} \leq (1 + \epsilon)$ for almost all random graphs $G \in \mathbf{\Gamma}_n(p)$. This makes it pointless to compare the probable performance of bandwidth minimization algorithms on random graphs in $\mathbf{\Gamma}_n(p)$. Therefore another class of probability distributions is introduced and used for most of the results given here. Let $G = (V, E)$ be generated by the following random experiment.

- Let $V = \{1, 2, \dots, n\}$. $E = \emptyset$.
- For each $\{u, v\}$, $1 \leq u < v \leq n$ such that $|u - v| \leq \psi$ include the edge $\{u, v\}$ in E with probability p .

The probability distribution defined by this experiment is denoted $\mathbf{\Psi}_n(\psi, p)$. Now, let $G \in \mathbf{\Psi}_n(\psi, p)$ and randomly re-number the vertices of G . The resulting distribution is denoted $\mathbf{\Omega}_n(\psi, p)$. Note that if $G \in \mathbf{\Omega}_n(\psi, p)$ then $\phi(G) \leq \psi$. Also, if H is a graph with $\phi(H) \leq \psi$, then H can be generated by $\mathbf{\Omega}_n(\psi, p)$. Furthermore, in §2 we show that for large enough ψ , almost all $G \in \mathbf{\Omega}_n(\psi, p)$ satisfy $\frac{\psi}{\phi(G)} \leq 1 + \epsilon$ for any fixed $\epsilon > 0$. The use of $\mathbf{\Omega}_n(\psi, p)$ allows us to explore properties that are common to most graphs having bandwidth $\leq \psi$, but rare for unrestricted graphs. Heuristics like the level algorithms exploit such properties to produce good layouts for most graphs.

It is shown in §3 that if A is any level algorithm and $A(G)$ is the bandwidth of the layout produced by A on the graph G then $A(G) < (1 + \epsilon)(3 - p)\phi(G)$ for almost all $G \in \mathbf{\Omega}_n(\psi, p)$,

where $\epsilon > 0$, $0 < p < 1$ are fixed, and $\ln n = o(\psi)$. If in addition $\psi < n/2$, then $(1-\epsilon)(2-p)\phi(G) < A(G)$. The analysis leads to a new class of algorithms called the *modified level algorithms*, for which it is shown that $A(G) \leq 2\phi(G) + O(\log n)$ for any modified level algorithm A and $G \in \Omega_n(\psi, p)$. In §4, a specific modified level algorithm, *MLA1* is studied and it is shown that $MLA1(G) = \phi(G) + O(\log n)$ for almost all $G \in \Omega_n(\psi, p)$ when $\psi < n/4$. §5 presents several other modified level algorithms, discusses running times and summarizes empirical studies comparing their performance. §6 shows how to improve the running times of the above algorithms through more careful selection of the ‘starting vertex’. Finally, §7 contains several results concerning properties of random graphs. Conditions are given for connectivity of random graphs in $\Psi_n(\psi, p)$ and probable upper bounds are given for the diameter of random graphs in $\Gamma_n(p)$ and $\Psi_n(\psi, p)$.

A word of caution. All but a few of the results proved in this paper are probabilistic in nature. That is, they hold for almost all graphs under some probability distribution. The statements of lemmas and theorems include the phrase “almost all” and specify the probability distribution, but to avoid being tedious, the proofs assert various properties without repeating this qualification.

2. Bandwidth of Graphs in $\Gamma_n(p)$ and $\Psi_n(\psi, p)$. The following results demonstrate that almost all random graphs in the usual model, have bandwidth nearly as large as the number of vertices.

Define $\lambda_n(c) = -\frac{\ln n}{\ln c}$. Note that $\lambda_n(c) > 0$ when $0 < c < 1$ and $n > 1$, $c^{\lambda_n(c)} = 1/n$ and $\lim_{n \rightarrow \infty} \lambda_n(c) = \infty$ for c fixed $0 < c < 1$. We will usually write $\lambda(c)$ for $\lambda_n(c)$.

Theorem 2.1. *Let $0 < p < 1$ be fixed. For almost all $G \in \Gamma_n(p)$, $\phi(G) > n - 4\lambda(1-p)$.*

Theorem 2.2. *Let $\epsilon > 0$, $c > 0$ be fixed, $p = (c \ln n)/n$. For almost all $G \in \Gamma_n(p)$, $\phi(G) \geq n(1 - \epsilon)$.*

For $G = (V, E)$, the notation $u-v$ means $\{u, v\} \in E$ and $u \not-v$ means that $\{u, v\} \notin E$. Similarly if $U \subseteq V$ and $W \subseteq V$ then $U-W$ means that some vertex in U is adjacent to some vertex in W . The proofs of Theorems 2.1 and 2.2 require the following lemmas.

Lemma 2.1. *Let $G = (V, E)$ be a graph on n vertices. $\phi(G) \leq n - 2k \Rightarrow \exists V_1, V_2 \subseteq V$ such that $|V_1| = |V_2| = k$ and $V_1 \not-V_2$.*

Proof. If $\phi(G) \leq n - 2k$ then there is a layout τ such that $u-v \Rightarrow |\tau(u) - \tau(v)| \leq n - 2k$. Let $V_1 = \{\tau^{-1}(1), \dots, \tau^{-1}(k)\}$ and $V_2 = \{\tau^{-1}(n - k + 1), \dots, \tau^{-1}(n)\}$. If $V_1 - V_2$ then there are vertices $u \in V_1$ and $v \in V_2$ such that $u-v$. But by the definition of V_1 and V_2 , $\tau(u) \leq k$ and $\tau(v) \geq n - k + 1$, hence $|\tau(u) - \tau(v)| > n - 2k$, which contradicts the definition of τ . \square

Lemma 2.2. *Let $0 < p < 1$ and $G = (V, E) \in \Gamma_n(p)$. $P(\phi(G) \leq n - 2k) \leq \left(\frac{en}{k}(1-p)^{k/2}\right)^{2k}$.*

Proof. By Lemma 2.1, $P(\phi(G) \leq n - 2k) \leq P(\exists V_1, V_2 \text{ such that } |V_1| = |V_2| = k \wedge V_1 \not-V_2)$. Since there are k^2 ‘potential edges’ joining V_1 and V_2 , all of which must be absent if $V_1 \not-V_2$, this last probability is

$$\leq \binom{n}{k} \binom{n-k}{k} (1-p)^{k^2} \leq \left(\frac{en}{k}\right)^2 k (1-p)^{k^2} = \left(\frac{en}{k}(1-p)^{k/2}\right)^{2k} \quad \square$$

Proof of Theorem 2.1. Applying Lemma 2.2 with $k = 2\lambda(1 - p)$ gives

$$\begin{aligned} \mathbb{P}(\phi(G) \leq n - 4\lambda(1 - p)) &\leq \left(\frac{en}{2\lambda(1 - p)} (1 - p)^{\lambda(1 - p)} \right)^{4\lambda(1 - p)} \\ &= \left(\frac{e}{2\lambda(1 - p)} \right)^{4\lambda(1 - p)} \rightarrow 0 \end{aligned}$$

□

Proof of Theorem 2.2. Applying Lemma 2.2 with $k = \epsilon n/2$ gives

$$\mathbb{P}(\phi(G) \leq n(1 - \epsilon)) \leq \left(\frac{\epsilon n}{\epsilon n/2} (1 - p)^{\epsilon n/4} \right)^{\epsilon n} \leq \left(\frac{2e}{\epsilon} e^{-\epsilon n p/4} \right)^{\epsilon n} \quad (1)$$

$$= \left(\frac{2e}{\epsilon} n^{-\epsilon c/4} \right)^{\epsilon n} \rightarrow 0 \quad (2)$$

□

Theorems 2.1 and 2.2 show that even for sparse random graphs $G \in \Gamma_n(p)$, $\frac{n}{\phi(G)} \rightarrow 1$. Consequently, even the most trivial algorithms (for example, the algorithm that always outputs the identity layout) can produce layouts having bandwidth close to $\phi(G)$ as n gets large. If one is to make meaningful distinctions among algorithms based on their probable performance, some other probability distribution is required. The distributions $\Omega_n(\psi, p)$ and $\Psi_n(\psi, p)$ are used here. Obviously, any structural property of a graph occurs with the same probability in both distributions. It is clear that if $G \in \Psi_n(\psi, p)$ then $\phi(G) \leq \psi$. The following theorem gives a probabilistic lower bound on $\phi(G)$.

Theorem 2.3. *Let $0 < p < 1$ be fixed, $\ln n \leq \psi < n$. For almost all $G \in \Psi_n(\psi, p)$, $\phi(G) > \psi - 4\lambda_\psi(1 - p)$.*

Proof. Let $G' \subseteq G$ be the subgraph induced by vertices $\{1, 2, \dots, \psi\}$. Note that G' is a random graph with distribution $\Gamma_\psi(p)$. Applying Theorem 2.1, $\phi(G') > \psi - 4\lambda_\psi(1 - p)$. The theorem follows from the fact that $\phi(G) \geq \phi(G')$. □

An immediate consequence of this result is that as ψ gets large, it comes within a factor of $1 + \epsilon$ of $\phi(G)$, for any fixed $\epsilon > 0$. While Theorem 2.3 is sufficient for the results proved here, it is interesting to consider a tighter relationship between ψ and $\phi(G)$.

Conjecture. *Let $0 < p < 1$ be fixed. There is some constant $c = c(p) > 0$ such that if $c \ln n \leq \psi \leq n - c \ln n$ then for almost all $G \in \Psi_n(\psi, p)$, $\phi(G) = \psi$.*

3. Probabilistic Algorithms for Bandwidth Minimization. Before proceeding we need the following definitions. Let $G = (V, E)$ and define $V_i(u) = \{v \mid d(u, v) = i\}$ for all $u \in V$. Also let $V_i = V_i(1)$. Next, define $l_i(u) = \min V_i(u)$ and $r_i(u) = \max V_i(u)$. Let $l_i = l_i(1)$ and $r_i = r_i(1)$. Note that $|V_i| \leq r_i - l_i$. Define

$$\text{level}(G) = \min_{u \in V} \max_{i \geq 0} |V_i(u)|$$

$$\text{LEVEL}(G) = \min_{u \in V} \max_{i \geq 0} |V_i(u) \cup V_{i+1}(u)| - 1$$

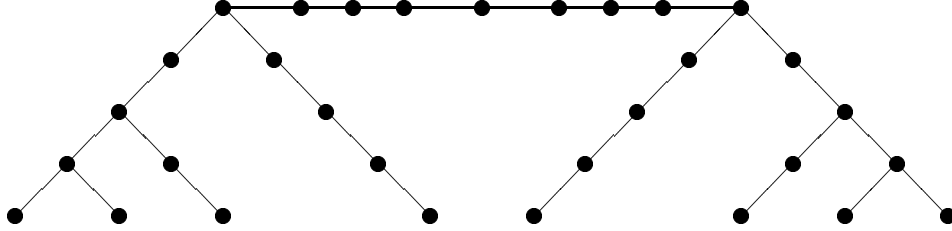


Figure 1: Tree Demonstrating Poor Worst-Case Performance of Level Algorithms

Note that if A is any level algorithm at all

$$level(G) \leq A(G) \tag{1}$$

and if A makes the best possible choice for $\tau^{-1}(1)$

$$A(G) \leq LEVEL(G) \tag{2}$$

In the next few sections, we will consider only algorithms that do always make the best choice. We can satisfy this requirement by trying all possible choices for $\tau^{-1}(1)$, at a cost of a factor of n in the running time. In §6, we will relax this restriction.

Consider the tree T in Figure 1. It is not difficult to see that $\phi(T) = 2$ and $level(T) = 4$. The example is readily extended. For any integer $k > 0$ one can construct a tree T_k such that $\phi(T_k) = 2$ and $level(T_k) = k$. (This result can be improved. There is a similar but more complicated construction which gives trees T_k with n vertices, $level(T_k) = \Omega(n/\log n)$ and $\phi(T_k) = o(\log n)$.) This implies that the worst case performance of the level algorithms can be arbitrarily poor. In spite of this, the level algorithms perform quite well on random graphs.

Theorem 3.1. *Let $\epsilon > 0$, $0 < p < 1$ be fixed, $\psi < n$, $\ln n = o(\psi)$. For almost all $G \in \Omega_n(\psi, p)$, $LEVEL(G) < (1 + \epsilon)(3 - p)\phi(G)$.*

The theorem is proved by deriving probable upper bounds on $|V_i|$ and then using the definition of $LEVEL$. These bounds are contained in the following lemma.

Lemma 3.1. *Let $\epsilon > 0$, $0 < p < 1$ be fixed, $\psi < n$, $\ln n = o(\psi)$. For almost all $G \in \Psi_n(\psi, p)$,*

$$\begin{aligned} |V_1| &< (1 + \epsilon)p\psi \\ |V_2| &< (1 + \epsilon)(2 - p)\psi \\ |V_i| &< (1 + \epsilon)\psi \quad \text{for } i \geq 3 \end{aligned}$$

The proof of Lemma 3.1 appears in §3.1 along with several technical lemmas required for its proof. We now use it to prove Theorem 3.1.

Proof of Theorem 3.1. By Lemma 3.1 there exists a vertex u for which $\max_{i \geq 0} |V_i(u) \cup V_{i+1}(u)| - 1 < (1 + \epsilon')(3 - p)\psi$ for any fixed $\epsilon' > 0$. Hence, $LEVEL(G) < (1 + \epsilon')(3 - p)\psi$ for

any fixed $\epsilon' > 0$. By Theorem 2.3, $\psi < (1 + \epsilon')\phi(G)$. Selecting ϵ' so that $(1 + \epsilon')^2 = (1 + \epsilon)$ yields the theorem. \square

In Lemma 3.5, it is shown that $|V_2| > (1 - \epsilon)(2 - p)\psi$ when $\ln n = o(\psi)$ and $\psi < n/2$. This result is easily extended to show that for all $u \in V$, $|V_2(u)| > (1 - \epsilon)(2 - p)\psi$ and hence $level(G) > (1 - \epsilon)(2 - p)\psi$. The details are left to the reader. A consequence of this is that the level algorithms are not capable of near optimal performance. However a related class of algorithms, called the *modified level algorithms* is. We will now describe the class of modified level algorithms. In the next section, we describe a specific member of this class that achieves near optimal performance. Define

$$\begin{aligned} V_2'(u) &= \begin{cases} V_2(u) & \text{if } V_3(u) = \emptyset \\ V_2(u) \cap \{v \mid v-w \in V_3(u)\} & \text{if } V_3(u) \neq \emptyset \end{cases} \\ V_1'(u) &= (V_1(u) \cup V_2(u)) - V_2'(u) \\ V_i'(u) &= V_i(u) \quad i = 0, i \geq 3 \end{aligned}$$

Also, let $V_i' = V_i'(1)$, $l_i'(u) = \min V_i'(u)$, $r_i'(u) = \max V_i'(u)$, $l_i' = l_i'(1)$, $r_i' = r_i'(1)$. Formally, A is a modified level algorithm, if the layout τ produced for the graph $G = (V, E)$ satisfies

$$\forall u, v \in V \quad u \in V_i'(\tau^{-1}(1)) \wedge v \in V_{i+1}'(\tau^{-1}(1)) \Rightarrow \tau(u) < \tau(v)$$

Let

$$\begin{aligned} level'(G) &= \min_{u \in V} \max_{i \geq 0} |V_i'(u)| \\ LEVEL'(G) &= \min_{u \in V} \max_{i \geq 0} |V_i'(u) \cup V_{i+1}'(u)| - 1 \end{aligned}$$

If A is any modified level algorithm then $level'(G) \leq A(G)$ and if A makes the best possible choice for the starting vertex then $A(G) \leq LEVEL'(G)$. For the modified level algorithms, we can show that for almost all $G \in \Psi_n(\psi, p)$, $|V_i'| < \psi + O(\log n)$ for all $i \geq 0$ when $\ln n = o(\psi)$, $\psi \leq (1 - \epsilon)n/2$. From this we obtain the following result.

Theorem 3.2. *Let $0 < p < 1$ be fixed, $\ln n = o(\psi)$, $\psi < n$. For almost all $G \in \Omega_n(\psi, p)$ $LEVEL'(G) = 2\phi(G) + O(\log n)$.*

The proof of Theorem 3.2 is contained in §3.2. This result shows that the class of modified level algorithms is capable of better performance than the class of level algorithms. In §4, we focus on a specific modified level algorithm and show that it produces nearly optimal layouts.

3.1. Technical Lemmas. The following lemmas are used in the proof of Lemma 3.1.

Lemma 3.2. *Let $\epsilon > 0$, $0 < p < 1$ be fixed, $\alpha = (1 + \epsilon)\lambda(1 - p^2) \leq \psi < n$. For almost all $G \in \Psi_n(\psi, p)$, there exists a path of length two between every pair of vertices u, v such that $|u - v| \leq 2\psi - \alpha$.*

Proof. Let $u, v \in V$ with $|u - v| \leq 2\psi - \alpha$. Let $i = 2\psi - |u - v|$. The probability that $d(u, v) > 2$ is $\leq (1 - p^2)^i$. Since for each i there are no more than n such pairs, the

probability that any pair is not joined by a 2-path is

$$\leq \sum_{i=\lceil \alpha \rceil}^{2\psi-1} n(1-p^2)^i < n(1-p^2)^\alpha \sum_{i=0}^{\infty} (1-p^2)^i = p^{-2}n^{-\epsilon} \rightarrow 0 \quad \square$$

Lemma 3.3. *Let $\epsilon > 0$, $0 < p < 1$ be fixed, $\alpha = (1+\epsilon)\lambda(1-p^2) \leq \psi < n$. For almost all $G \in \Psi_n(\psi, p)$, $r_i - 3\psi \leq r_{i-3} < l_i - (2\psi - \alpha)$, for all $i \geq 3$.*

Proof. The shortest path from 1 to r_i must pass through some $u \in V_{i-3}$. Clearly $r_i - u \leq 3\psi$, hence $r_i - 3\psi \leq u \leq r_{i-3}$. To see that $r_{i-3} < l_i - (2\psi - \alpha)$, assume otherwise. Then there is some vertex v on the shortest path from 1 to r_{i-3} such that $l_i - (2\psi - \alpha) \leq v < l_i$ and $d(1, v) \leq i - 3$. By Lemma 3.2 there is a 2-path from v to l_i , giving $d(1, l_i) \leq i - 1$, which is a contradiction. \square

Lemma 3.4. *Let $\epsilon > 0$, $0 < p < 1$ be fixed, $\alpha = (1+\epsilon)\lambda(1-p^2) \leq \psi < n$. For almost all $G \in \Psi_n(\psi, p)$, $r_i - l_i < \psi + \alpha$, for all $i \geq 3$.*

Proof. By Lemma 3.3 $r_i \leq r_{i-3} + 3\psi$ and $l_i > r_{i-3} + (2\psi - \alpha)$. Hence,

$$r_i - l_i < (r_{i-3} + 3\psi) - (r_{i-3} + (2\psi - \alpha)) = \psi + \alpha \quad \square$$

From Lemma 3.4, we conclude that $|V_i| < \psi + \alpha$ for $i \geq 3$, but the lemma says nothing about the size of V_1 and V_2 . As we shall see, these cases differ from the rest and will be handled in Lemma 3.5. First however, we need a proposition concerning the binomial distribution, $\mathbf{B}(n, p)$. By definition if $x \in \mathbf{B}(n, p)$ then $P(x = k) = \binom{n}{k} p^k (1-p)^{n-k}$. The following proposition is from Angluin and Valiant [1].

Proposition 3.1. *If $x \in \mathbf{B}(n, p)$ then for any ϵ , $0 < \epsilon < 1$, $P(x \leq (1-\epsilon)np) < e^{-\epsilon^2 np/2}$ and $P(x \geq (1+\epsilon)np) < e^{-\epsilon^2 np/3}$.*

Lemma 3.5. *Let $\epsilon > 0$, $0 < p < 1$ be fixed, $c = -(1+\epsilon)/\ln(1-p^2)$, $\alpha = c \ln n \leq \psi < n$. For almost all $G \in \Psi_n(\psi, p)$, $(1-\epsilon)p\psi < |V_1| < (1+\epsilon)p\psi$ and $|V_2| < (1+\epsilon)(2-p)\psi$. Also, if $\psi < n/2$ then $(1-\epsilon)(2-p)\psi - \alpha < |V_2|$.*

Proof. $|V_1|$ is a binomial random variable in $\mathbf{B}(\psi, p)$. By Proposition 3.1,

$$P(|V_1| < (1-\epsilon)p\psi) < e^{-\epsilon^2 p\psi/2} \rightarrow 0$$

$$P(|V_1| > (1+\epsilon)p\psi) < e^{-\epsilon^2 p\psi/3} \rightarrow 0$$

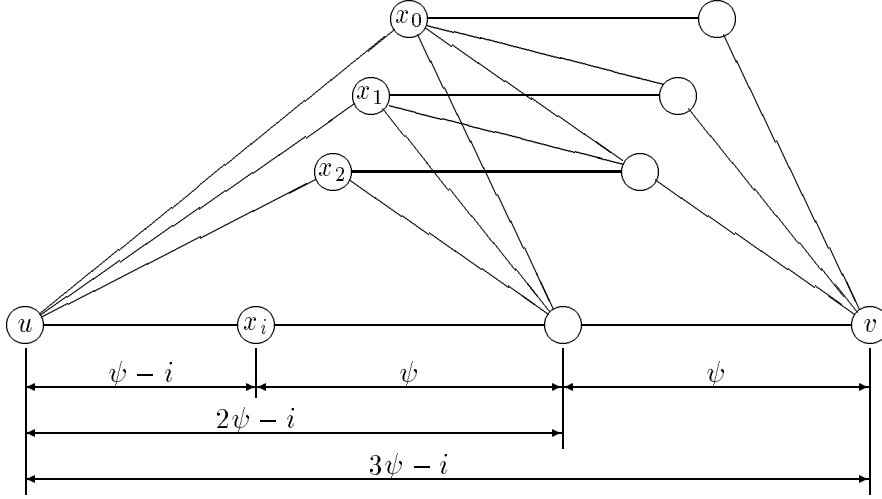
This establishes the bounds on $|V_1|$. Since $|V_2| \leq 2\psi - |V_1|$,

$$|V_2| < 2\psi - (1-\epsilon)p\psi < (1+\epsilon)(2-p)\psi$$

When $\psi < n/2$, Lemma 3.2 gives

$$|V_2| \geq (2\psi - \alpha) - |V_1| > (2 - (1+\epsilon)p)\psi - \alpha > (1-\epsilon)(2-p)\psi - \alpha \quad \square$$

Proof of Lemma 3.1. Immediate from Lemmas 3.4 and 3.5. \square

Figure 2: Definition of x_i s

3.1. More Technical Lemmas. The following lemmas are used in the proof of Theorem 3.2.

Lemma 3.6. *Let $\epsilon > 0$, $0 < p < 1$ be fixed, $(1 + \epsilon)\lambda(1 - p^2) \leq \psi < n$. For almost all $G = (V, E) \in \Psi_n(\psi, p)$, $u \in V \wedge |\{u - \psi, \dots, u + \psi\} \cap V_i| \geq (1 + \epsilon)\lambda(1 - p) \Rightarrow u \in V_i$.*

Proof. By Lemma 3.2, any pair of vertices u, v with $|u - v| \leq \psi$, is joined by a 2-path and hence $|d(1, u) - d(1, v)| \leq 2$. Thus, for each vertex u there are at most five sets V_i such that $|\{u - \psi, \dots, u + \psi\} \cap V_i| \geq 1$. Hence, the probability that for any $G \in \Psi_n(\psi, p)$, the assertion is not true is $\leq 5n(1 - p)^{(1 + \epsilon)\lambda(1 - p)} = 5n^{-\epsilon} \rightarrow 0$. \square

Lemma 3.7. *Let $\epsilon > 0$, $0 < p < 1$ be fixed, $\alpha = (1 + \epsilon)\lambda(1 - p^2) \leq \psi < n$. For almost all $G \in \Psi_n(\psi, p)$ there exists a path of length three between every pair of vertices u, v such that $|u - v| \leq 3\psi - \alpha$.*

Proof. Let $u, v \in V$ be such that $i = 3\psi - |u - v| \geq \alpha$. Let $x_j = u + \psi - j$ for $0 \leq j \leq i$, as illustrated in Figure 2. Clearly any 3-path connecting u and v must pass through one of x_0, \dots, x_i . The probability that no 3-path joins u and v is

$$\begin{aligned}
&= \text{P}(\text{no 3-path} \wedge u \not\sim x_0 \wedge \dots \wedge u \not\sim x_i) \\
&\quad + \sum_{j=0}^i \text{P}(\text{no 3-path} \wedge u \not\sim x_0 \wedge \dots \wedge u \not\sim x_{j-1} \wedge u \sim x_j) \\
&= (1 - p)^{i+1} \text{P}(\text{no 3-path} \mid u \not\sim x_0 \wedge \dots \wedge u \not\sim x_i) \\
&\quad + \sum_{j=0}^i p(1 - p)^j \text{P}(\text{no 3-path} \mid u \not\sim x_0 \wedge \dots \wedge u \not\sim x_{j-1} \wedge u \sim x_j) \\
&< (1 - p)^{i+1}
\end{aligned}$$

$$\begin{aligned}
& +p \sum_{j=0}^i (1-p)^j \mathbb{P}(\text{no 3-path} \mid u \not\sim x_0 \wedge \cdots \wedge u \not\sim x_{j-1} \wedge u \not\sim x_j \wedge u \not\sim x_{j+1} \wedge \cdots \wedge u \not\sim x_i) \\
= & (1-p)^{i+1} + p \sum_{j=0}^i (1-p)^j (1-p^2)^{i-j+1} \\
= & (1-p)^{i+1} + p(1+p)(1-p)^{i+1} \sum_{j=0}^i (1+p)^j \\
= & (1-p)^{i+1} + p(1+p)(1-p)^{i+1} \frac{(1+p)^{i+1} - 1}{p} \\
< & (1+p)(1-p^2)^{i+1}
\end{aligned}$$

Since for each value of i there are at most n vertex pairs u, v such that $|u - v| = i$, the probability that any pair u, v with $|u - v| \leq 3\psi - \alpha$ is not connected by a 3-path is

$$\leq \sum_{i=\lceil \alpha \rceil}^{3\psi-1} n(1+p)(1-p^2)^{i+1} < n(1+p)(1-p^2)^\alpha \sum_{i=0}^{\infty} (1-p^2)^i = \frac{1+p}{p^2} n^{-\epsilon} \rightarrow 0 \quad \square$$

Lemma 3.8. *Let $\epsilon > 0$, $0 < p < 1$ be fixed, $\alpha = (1 + \epsilon)\lambda(1 - p^2)$, $\beta = (1 + \epsilon)\lambda(1 - p)$ and $\max(\alpha, 2\beta) \leq \psi \leq (n - \beta)/2$. For almost all $G \in \Psi_n(\psi, p)$ $|V'_i| \leq \psi + \alpha$ for $i \geq 0$.*

Proof. The result follows from Lemma 3.4 for $i \geq 3$ and is immediate for $i = 0$. Before proving the theorem for $1 \leq i \leq 2$ we first need to show that $|V_3 \cap \{\psi + 2, \dots, 2\psi + \lceil \beta \rceil\}| \geq \beta$. Let $A = \{\psi + 2, \dots, 2\psi + 1\}$. By Lemmas 3.4 and 3.7 $A \subseteq V_2 \cup V_3$. Let $x = |A \cap V_3|$. Clearly if $x \geq \beta$ then we're done. Assume then that $x < \beta$ and let $B = \{2\psi + 2, \dots, 2\psi + (\lceil \beta \rceil - x) + 1\}$ and let $y = |B|$. If $u \in B$, then $|\{u - \psi, \dots, 2\psi + 1\} \cap V_2| \geq \psi - \beta$. Since $\psi \geq 2\beta$, $u \in B \Rightarrow |\{u - \psi, \dots, u + \psi\} \cap V_2| \geq \beta$. Thus by Lemma 3.6 $B \subseteq V_3$. Since $x + y \geq \beta$ we have that $|V_3 \cap \{\psi + 2, \dots, 2\psi + \lceil \beta \rceil\}| \geq \beta$.

Now, by Lemma 3.3, $l_3 \geq 2\psi - \alpha + 1$. This implies that $l'_2 \geq \psi - \alpha + 1$ and since $r'_2 \leq 2\psi + 1$, it follows that $|V'_2| \leq \psi + \alpha$ as claimed.

Finally, note that if $u \in A$ and $u \geq \psi + \beta$ then using Lemma 3.6, one can show that $u \not\sim V_3$ and hence $u \notin V'_1$. Thus $|V'_1| \leq \psi + \beta < \psi + \alpha$ as claimed. \square

Proof of Theorem 3.2. If $\psi > (n - 2\lambda(1 - p))/2$, then since $LEVEL'(G) < n$

$$LEVEL'(G) < 2\psi + 2\lambda(1 - p) = 2\psi + O(\log n) = 2\phi(G) + O(\log n)$$

If $\psi \leq (n - 2\lambda(1 - p))/2$ then we can apply Lemma 3.8 giving

$$LEVEL'(G) < 2\psi + 4\lambda(1 - p^2) = 2\psi + O(\log n) = 2\phi(G) + O(\log n) \quad \square$$

A similar analysis yields $level'(G) = \phi(G) + O(\log n)$.

4. Obtaining Nearly Optimal Layouts. In this section a specific modified level algorithm denoted *MLA1* is described and analyzed. It is shown that *MLA1* is capable of producing nearly optimal layouts for random graphs in $\Omega_n(\psi, p)$.

For each $u \in V$

Let τ be any layout that satisfies the following conditions for all $x, y \in V$.

- (a) $x \in V'_i(u) \wedge y \in V'_{i+1}(u) \Rightarrow \tau(x) < \tau(y)$
- (b) $1 \leq i \leq 2 \wedge x, y \in V'_i(u) \wedge |gc_u(x)| < |gc_u(y)| \Rightarrow \tau(x) < \tau(y)$
- (c) $i \geq 3 \wedge x, y \in V'_i(u) \wedge |gp_u(x)| > |gp_u(y)| \Rightarrow \tau(x) < \tau(y)$

Output the layout having minimum bandwidth.

Figure 3: Modified Level Algorithm 1

For a graph $G = (V, E)$, we define the *grandchildren* of v with respect to u by

$$gc_u(v) = V_2(v) \cap V'_{i+2}(u) \quad \forall v \in V'_i(u)$$

and the *grandparents* of v with respect to u by

$$gp_u(v) = V_2(v) \cap V'_{i-2}(u) \quad \forall v \in V'_i(u)$$

Also let $gc(v) = gc_1(v)$, $gp(v) = gp_1(v)$. The algorithm we will analyze is based on the observation that for $G \in \Psi_n(\psi, p)$ if $u, v \in V'_i$ and $v - u$ is not too small, then with high probability $|gc(u)| < |gc(v)|$ and $|gp(u)| > |gp(v)|$. The algorithm *MLA1* is described in Figure 3. Define $MLA1(G)$ as the bandwidth of the layout produced by *MLA1* on graph G .

Theorem 4.1. *Let $0 < p < 1$ be fixed, $\ln n = o(\psi)$, $\psi \leq n/4$. For almost all $G \in \Omega_n(\psi, p)$ $MLA1(G) = \phi(G) + O(\log n)$.*

The key fact used in the proof of Theorem 4.1 is contained in the following lemma.

Lemma 4.1. *Let $\epsilon > 0$, $0 < p < 1$ be fixed, $\alpha = (1 + \epsilon)\lambda(1 - p^2)$, $\ln n = o(\psi)$, $\psi \leq n/4$. For almost all $G \in \Psi_n(\psi, p)$, $|\tau(u) - u| \leq 4\alpha$, where $u \in V$ and τ is the layout produced by *MLA1* for which $\tau(1) = 1$.*

Proof of Theorem 4.1. By Lemma 4.1, *MLA1* will compute a layout in which no vertex is more than 4α from the ‘right position’. This implies that the bandwidth of the layout output by *MLA1* is at most $\psi + 8\alpha = \phi(G) + O(\log n)$. \square

The proof of Lemma 4.1 requires the following technical lemmas.

Lemma 4.2. *Let $\epsilon > 0$, $0 < p < 1$ be fixed, $\alpha = (1 + \epsilon)\lambda(1 - p^2)$, $2\alpha \leq \psi < n$. For almost all $G \in \Psi_n(\psi, p)$*

$$\begin{aligned} l'_1 &= 2 & \psi - \alpha &\leq r'_1 \leq \psi + \alpha \\ r'_1 - 2\alpha &< l'_2 &\leq r'_1 + 1 & \quad r'_1 + \psi - \alpha &\leq r'_2 &\leq r'_1 + \psi \\ r'_{i-1} - \alpha &< l'_i &\leq r'_{i-1} + 1 & \quad r'_{i-1} + \psi - \alpha &\leq r'_i &\leq r'_{i-1} + \psi \quad \text{for } i \geq 3 \end{aligned}$$

Proof. For $1 \leq i \leq 2$ the result is implicit in the proof of Lemma 3.8. For $i \geq 3$, Lemma 3.3 gives $l'_i > r'_{i-3} + 2\psi - \alpha$. Since $r'_{i-1} \leq r'_{i-3} + 2\psi$, $l'_i > r'_{i-1} - \alpha$. By Lemmas 3.2 and 3.7

$\{r'_{i-3} + \psi + 1, \dots, r'_{i-3} + 2\psi - \alpha\} \subseteq V'_{i-1}$ and $\{r'_{i-3} + 2\psi + 1, \dots, r'_{i-3} + 3\psi - \alpha\} \subseteq V'_i$ and $\{r'_{i-3} + 2\psi - \alpha + 1, \dots, r'_{i-3} + 2\psi\} \subseteq V'_{i-1} \cup V'_i$. This implies $r'_{i-1} + 1 \in V'_i$, hence $l'_i \leq r'_{i-1} + 1$. For $i \geq 3$, $r'_i \leq r'_{i-1} + \psi$ is immediate and $r'_i \geq r'_{i-1} + \psi - \alpha$ follows from Lemma 3.7 and $r'_{i-1} \leq r'_{i-3} + 2\psi$. \square

A consequence of Lemma 4.2 is that at least $\psi - \alpha$ of the vertices in V'_i are found in a region containing only vertices in V'_i . The regions associated with V'_i and V'_{i+1} are separated by a transition region containing at most 2α vertices.

Lemma 4.3. *Let $\epsilon > 0$, $0 < p < 1$ be fixed, $\alpha = (1 + \epsilon)\lambda(1 - p^2)$, $4\alpha \leq \psi \leq n/4$. For almost all $G \in \Psi_n(\psi, p)$ $1 \leq i \leq 2 \wedge u, v \in V'_i \wedge u - v \geq 4\alpha \Rightarrow |gc(u)| > |gc(v)|$.*

Proof. Lemmas 3.2 and 4.2 imply that if $u, v \in V'_i$ and $u - v \geq \alpha$ then $gc(v) \subseteq gc(u)$. It remains only to show that there is some vertex $x \in gc(u) - gc(v)$. Let $x = u + 2\psi - \lceil \alpha \rceil$. If $u - v \geq 4\alpha$, Lemma 4.2 yields,

$$r'_{i+1} \leq r'_{i-1} + 2\psi < l'_i + 2\psi + 2\alpha \leq v + 2\psi + 2\alpha < u + 2\psi - 2\alpha < x$$

Thus $x \notin V'_{i+1}$ and also by Lemma 4.2, $x \notin V'_j$ for any $j \leq i$. Since, by Lemma 3.2, there is a 2-path from u to x , $x \in gc(u)$. Since $x > v + 2\psi$, $x \notin gc(v)$. \square

Lemma 4.4. *Let $\epsilon > 0$, $0 < p < 1$ be fixed, $\alpha = (1 + \epsilon)\lambda(1 - p^2)$, $4\alpha \leq \psi \leq n/4$. For almost all $G \in \Psi_n(\psi, p)$ $i \geq 3 \wedge u, v \in V'_i \wedge u - v \geq 4\alpha \Rightarrow |gp(u)| < |gp(v)|$.*

Proof. Lemmas 3.2 and 4.2 imply that if $u, v \in V'_i$ and $u - v \geq \alpha$ then $gp(u) \subseteq gp(v)$. It remains to show that there exists some vertex x in $gp(v) - gp(u)$. Let $x = v - 2\psi + \lceil \alpha \rceil$. If $u - v \geq 4\alpha$, Lemma 4.2 yields

$$l'_{i-1} > r'_{i-2} - 2\alpha \geq r'_i - 2\psi - 2\alpha \geq u - 2\psi - 2\alpha \geq v - 2\psi + 2\alpha > x$$

Thus $x \notin V'_{i-1}$ and also by Lemma 4.2, $x \notin V'_j$ for any $j \geq i$. Since, by Lemma 3.2, there is a 2-path from v to x , $x \in gp(v)$. Since $x < u - 2\psi$, $x \notin gp(u)$. \square

Proof of Lemma 4.1. By Lemmas 4.2 to 4.4, if $u - v > 4\alpha$ then $\tau(v) < \tau(u)$. Consequently, for any u there can be at most 4α vertices v such that $u > v$ and $\tau(u) < \tau(v)$. Similarly, there can be at most 4α vertices w such that $u < w$ and $\tau(u) > \tau(w)$. Hence, $|\tau(u) - u| \leq 4\alpha$. \square

5. Pragmatics. This section reports on the results of empirical studies of several modified level algorithms, including *MLA1*, described in the previous section. It also contains some implementation details and analyses of the algorithms' running times.

Four modified level algorithms were studied. They are denoted here as *MLA1* through *MLA4*. An implementation of *MLA1* is shown in Figure 4. This procedure returns a layout τ , with u as the starting vertex. The strategy for ordering the vertices within levels is the one described in the previous section. The procedure shown calls several others. *Make_mod_levels*($G, u, V'_0, \dots, V'_{n-1}$) computes $V'_i(u)$ and returns it in the list V'_i for $0 \leq i \leq n - 1$, using breadth-first-search. The procedures *count_gc* and *count_gp* count the number of 'grandchildren' and 'grandparents' for vertices in the levels specified by the last two arguments. (For example, the call to *count_gc* in line (4) counts the grandchildren of all vertices in the first two levels and returns the results in the array *ngc*.) The procedure

```

(1) procedure MLA1( $G = (V, E), u, \tau$ )
(2)    $n := |V|$ ;
(3)   make_mod_levels( $G, u, V', \dots, V'_{n-1}$ );
(4)   count_gc( $G, u, V'_0, \dots, V'_{n-1}, ngc, 1, 2$ );
(5)   count_gp( $G, u, V'_0, \dots, V'_{n-1}, ngp, 3, n - 1$ );
(6)   for  $i \in [1, 2] \rightarrow \text{sort}(V'_i, ngc(x) < ngc(y))$  rof;
(7)   for  $i \in [3, n - 1] \rightarrow \text{sort}(V'_i, ngp(x) > ngp(y))$  rof;
(8)    $next := 1$ ;      { next position in layout }
(9)   for  $i \in [0, n - 1] \rightarrow$ 
(10)      for  $x \in V'_i \rightarrow \tau(x) := next; next := next + 1$  rof;
(11)  rof;
(12)  return;
(13) end;

```

Figure 4: Implementation Details for *MLA1*

$\text{sort}(L, R(x, y))$ sorts the list L so that x precedes y in the sorted list if and only if x is related to y under R . For example, $\text{sort}(V'_i, ngc(x) < ngc(y))$ sorts V'_i so that if $ngc(x) < ngc(y)$ then x precedes y in V'_i . The running time of *MLA1* is dominated by the *count_gc* and *count_gp* functions. A straightforward implementation of these gives a running time of $O(n\psi^2)$. The procedure *make_mod_levels* can be implemented to run in $O(|E|) = O(n\psi)$ time, and the sorting steps in lines [6] and [7] require at most $O(n \log n)$.

There are other possible strategies for arranging the vertices within each level. Cuthill and McKee [5], who first suggested the level algorithms, arranged the vertices within levels according to the order in which they were visited by a breadth-first search algorithm. This results in an arbitrary ordering of the first level and arranges each vertex in subsequent levels based on the position of its ‘leftmost’ neighbor. Cheng [2,3] refined this strategy by ordering the vertices in the first level in increasing order of the number of neighbors in the next level. Adapting this algorithm to the modified level strategy gives the algorithm *MLA2*, which is shown in Figure 5. *MLA2* calls the procedure *count_ch*, which counts the number of neighbors each vertex has in the ‘next level’. As with *count_gc* and *count_gp*, the calculation is done only for those levels specified by the last two arguments (in this case, just the first level). This can be done in $O(\psi^2)$ time, while the remainder of *MLA2* can be done in $O(n\psi)$ time.

The procedure *MLA3*, shown in Figure 6 is a cross between *MLA1* and *MLA2*. It uses the strategy of *MLA1* to order the vertices in the first level, then reverts to the strategy of *MLA2* for all subsequent levels. Computing *ngc* for the vertices in the first level requires $O(\psi^3)$ time. The remainder of *MLA3* can be done in $O(n\psi)$.

MLA4 is a refinement of *MLA3* designed to improve the running time when the bandwidth is fairly large. Instead of using the number of ‘grandchildren’ to order the vertices in

```

(1) procedure MLA2( $G = (V, E), u, \tau$ )
(2)    $n := |V|$ ;
(3)   make_mod_levels( $G, u, V'_0, \dots, V'_{n-1}$ );
(4)   count_ch( $G, u, V'_0, \dots, V'_{n-1}, nch, 1, 1$ );
(5)   sort( $V'_1, nch(x) < nch(y)$ );
(6)   for  $x \in V \rightarrow \tau(x) := 0$  rof;      { 0 denotes undefined }
(7)    $\tau(u) := 1$ ;
(8)    $next := 2$ ;      { next position in layout }
(9)   for  $x \in V'_1 \rightarrow \tau(x) := next; next := next + 1$  rof;
(10)   $left := 2$ ;      { left end of  $V'_1$  in layout }
(11)   $right := next - 1$ ;      { right end of  $V'_1$  in layout }
(12)  for  $i \in [2, n - 1] \rightarrow$ 
(13)    do  $left \leq right \rightarrow$ 
(14)       $x := \tau^{-1}(left)$ ;
(15)      for  $\{x, y\} \in E \rightarrow$ 
(16)        if  $\tau(y) = 0 \rightarrow \tau(y) := next; next := next + 1$  fi
(17)      rof;
(18)       $left := left + 1$ ;
(19)    od;
(20)     $right := next - 1$ ;      { right end of  $V'_i$  }
(21)  rof;
(22)  return
(23) end

```

Figure 5: Implementation Details for *MLA2*

the first level, it uses the number of paths to grandchildren. This can be computed more quickly, since it eliminates the necessity of throwing out duplicates. The total running time of *MLA4* is $O(n\psi)$.

MLA2 through *MLA4* are more difficult to analyze than *MLA1* because decisions made in ordering each level affect the ordering of subsequent levels. Consequently, one might expect that errors made in ordering the early levels could accumulate and cause large errors further on. Experimental results suggest that in fact this does not happen, that the process is self-limiting. However, straightforward analytical techniques for bounding the error give unsatisfactory results.

Figures 7 through 9 summarize the results of a series of experiments that were undertaken to verify the theoretical performance bounds described in the previous sections for *MLA1*, provide tighter bounds for graphs of moderate size and compare *MLA1* to the other modified level algorithms. For each of the data points shown in Figure 7, ten random graphs in $\Psi_n(n/4, 1/2)$ were generated and each of the algorithms was run. For each algorithm, these ten results were averaged and the difference between these averages and $n/4$ were

```

(1) procedure MLA3( $G = (V, E), u, \tau$ )
(2)    $n := |V|$ ;
(3)   make_mod_levels( $G, u, V'_0, \dots, V'_{n-1}$ );
(4)   count_gc( $G, u, V'_0, \dots, V'_{n-1}, ngc, 1, 1$ );
(5)   sort( $V'_1, ngc(x) < ngc(y)$ );
(6)   for  $x \in V \rightarrow \tau(x) := 0$  rof;      { 0 denotes undefined }
(7)    $\tau(u) := 1$ ;
(8)    $next := 2$ ;                          { next position in layout }
(9)   for  $x \in V'_1 \rightarrow \tau(x) := next; next := next + 1$  rof;
(10)   $left := 2$ ;                            { left end of  $V'_1$  in layout }
(11)   $right := next - 1$ ;                    { right end of  $V'_1$  in layout }
(12)  for  $i \in [2, n - 1] \rightarrow$ 
(13)    do  $left \leq right \rightarrow$ 
(14)       $x := \tau^{-1}(left)$ ;
(15)      for  $\{x, y\} \in E \rightarrow$ 
(16)        if  $\tau(y) = 0 \rightarrow \tau(y) := next; next := next + 1$  fi
(17)      rof
(18)       $left := left + 1$ ;
(19)    od;
(20)     $right := next - 1$ ;                  { right end of  $V'_i$  }
(21)  rof;
(22)  return
(23) end

```

Figure 6: Implementation Details for *MLA3*

plotted. The results show that all the algorithms produce good layouts. All of the results are within 20% of $n/4$ and the best are within 2%.

Figure 8 shows the measured execution times for these runs. (The algorithms were coded in the C programming language and run on a VAX 11/750 under Unix¹.) Here, *MLA2* and *MLA4* enjoy a substantial advantage. Of course, this speed advantage is directly related to the large value of ψ relative to n . For smaller values of ψ the differences would be less.

One last set of results is shown in Figure 9. This shows how the performance of the algorithms deteriorates as ψ becomes large relative to n . *MLA1* deteriorates first, when $\psi \approx n/4$. This is because, the strategy used to order the levels becomes less effective when V'_4 becomes much smaller than ψ . *MLA3* is not affected by this phenomenon until $\psi \approx n/3$ since the ‘grandchildren’ strategy is used only to order the first level. *MLA2* and *MLA4* are more robust, maintaining their good performance until $\psi \approx n/2$. At this point all four degenerate from modified level algorithms to level algorithms.

¹Unix is a trademark of AT&T.

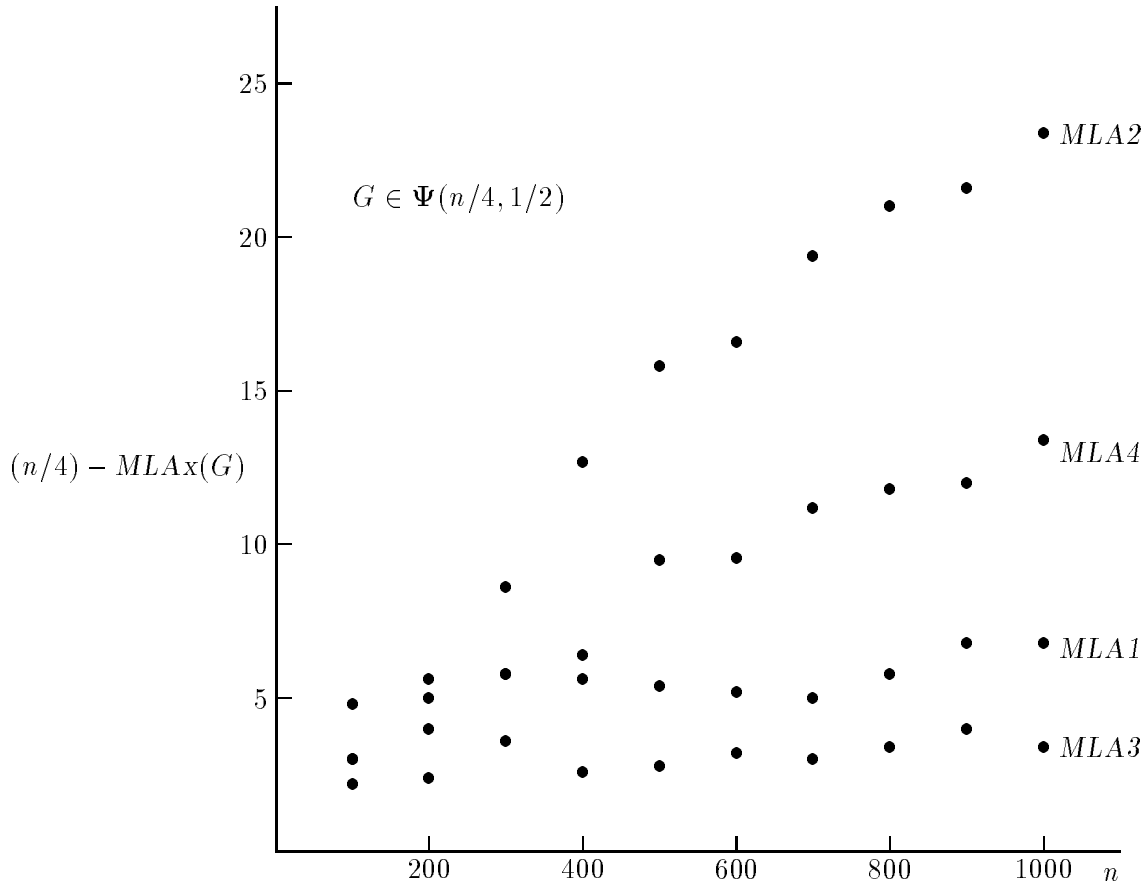


Figure 7: Performance of Modified Level Algorithms

6. Selection of Starting Vertices. Up until this point we have largely ignored the question of how one selects a good starting vertex in the modified level algorithm. Of course, the brute force solution is simply to try all possibilities and pick the best result. This adds a factor of n to the running times quoted in the previous sections, but does ensure the best possible choice. In this section, we consider strategies that permit us to select small sets of candidate starting vertices, that with high probability, contain a good choice.

The most obvious strategy (suggested by Cuthill and McKee) is to concentrate on vertices with small degree. For $G \in \Psi_n(\psi, p)$ it's reasonable to expect the degree of vertex 1 will be smaller than the degree of most other vertices. The following lemma puts a probable upper bound on the number of low degree vertices that need to be tried to obtain near optimal performance. For $G = (V, E)$, define the set of *low degree vertices* by $ld(G) = \{v \in V \mid d(v) \leq d(1)\}$.

Lemma 6.1. *Let $\epsilon > 0$, $0 < p < 1$ be fixed, $12(1 + \epsilon)(1/p) \ln n \leq \psi < n$. For almost all $G \in \Psi_n(\psi, p)$, $|ld(G)| < 4\sqrt{(3/p)(1 + \epsilon)\psi \ln n}$.*

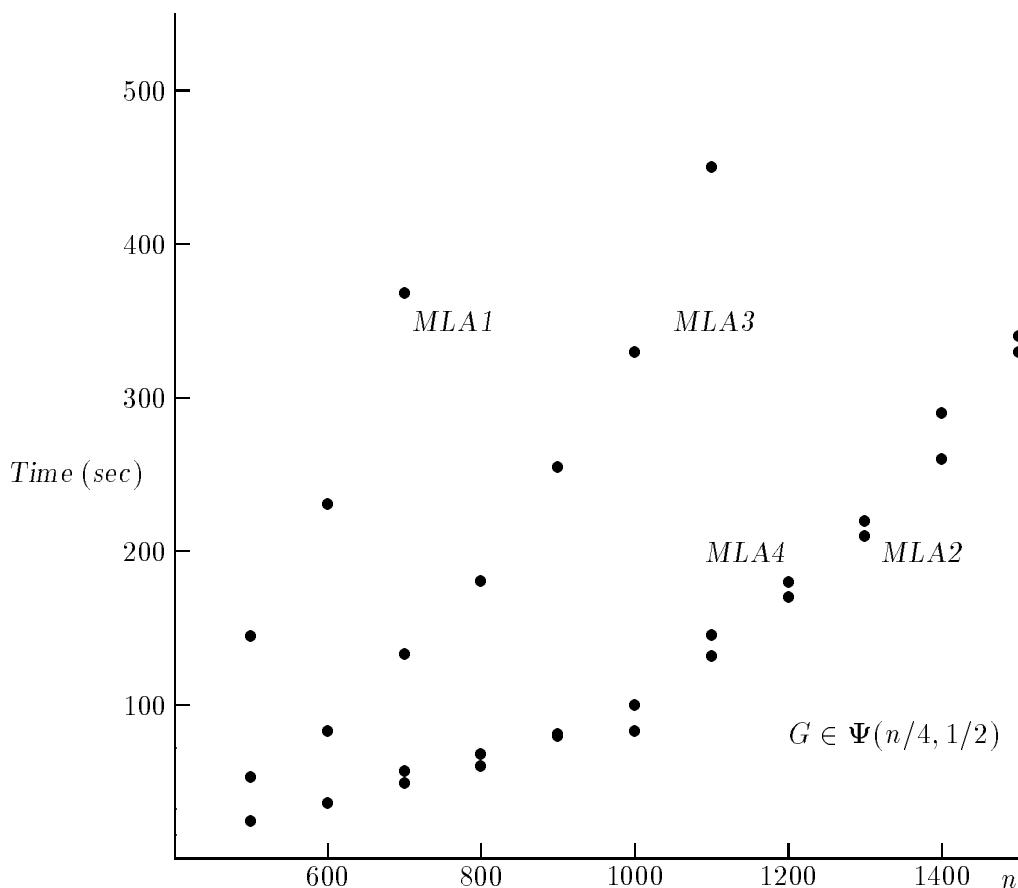


Figure 8: Running Time of Modified Level Algorithms

Proof. Let $0 < \alpha \leq \psi p/2$. By Proposition 3.1

$$P(d(1) \geq \psi p + \alpha) < e^{-\alpha^2/3\psi p}$$

For $v \in V$ such that $(2\alpha/p) < v < n - (2\alpha/p)$

$$P(d(v) \leq \psi p + \alpha) < e^{-\alpha^2/2\psi p}$$

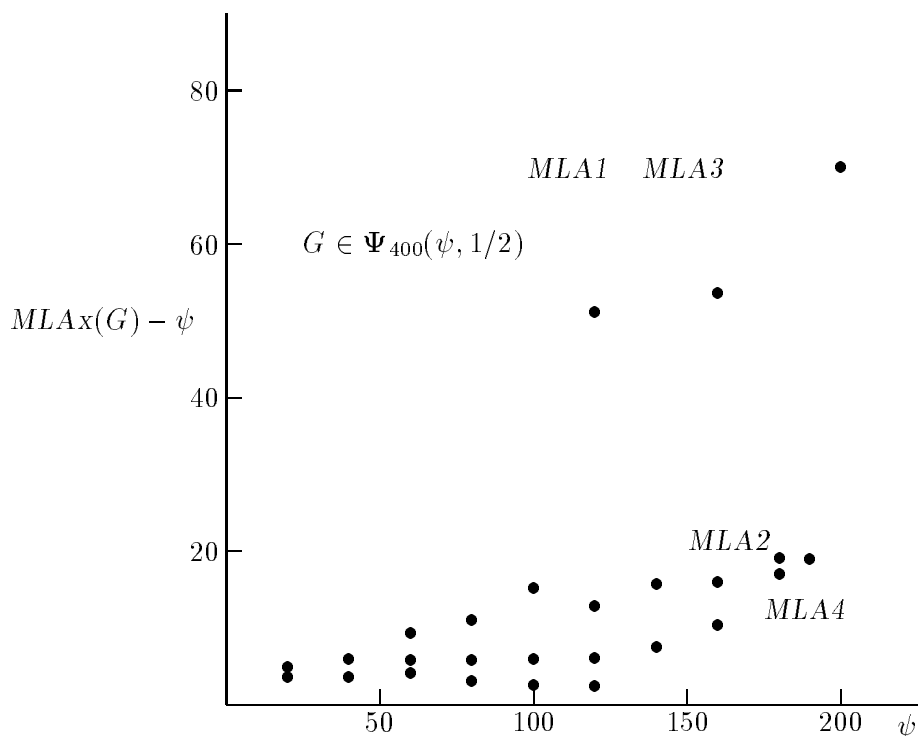
Letting $\alpha = \sqrt{3(1+\epsilon)\psi p \ln n}$ yields

$$P(d(v) < d(1)) < 2e^{-\alpha^2/3\psi p} = 2n^{-(1+\epsilon)}$$

Since there are $< n$ such vertices v ,

$$P(\exists v \mid (2\alpha/p) < v < n - (2\alpha/p) \wedge d(v) < d(1)) < 2n^{-\epsilon} \rightarrow 0$$

Consequently there are at most $4\alpha/p$ vertices in $ld(G)$. \square

Figure 9: Deterioration of Modified Level Algorithms as ψ Grows

Lemma 6.1 gives us a way of ensuring a good starting vertex. The cost is an added factor of $O(\sqrt{\psi \ln n})$ in the running time.

The next theorem suggests another method for identifying a good starting vertex. Let $L_x(G)$ be the layout τ of G produced by *MLA1* for which $\tau(x) = 1$ and let $MLA1_x(G)$ be the bandwidth of G with respect to $L_x(G)$.

Theorem 6.1. *Let $0 < p < 1$ be fixed, $\ln n = o(\psi)$, $\psi \leq n/16$. For almost all $G \in \Psi_n(\psi, p)$ ($x \in V \wedge \tau = L_x(G) \wedge y = \tau^{-1}(n)$) $\Rightarrow MLA1_y(G) = \phi(G) + O(\log n)$.*

The procedure suggested by Theorem 6.1 is this. Pick an arbitrary vertex x and run *MLA1* with x as the starting vertex. Let y be the ‘rightmost vertex’ in the resulting layout. Now, re-run *MLA1* with y as the starting vertex. Theorem 6.1 states that the resulting layout is close to optimal. The proof of Theorem 6.1 requires the following lemmas.

Lemma 6.2. *Let $\epsilon > 0$, $0 < p < 1$ be fixed, $\alpha = (1 + \epsilon)\lambda(1 - p^2)$, $\ln n = o(\psi)$, $\psi \leq n/16$. For almost all $G \in \Psi_n(\psi, p)$ ($x \in V \wedge \tau = L_x(G) \wedge y = \tau^{-1}(n)$) $\Rightarrow (y < 4\alpha \vee y > n - 4\alpha)$.*

Proof. Let $x \in V$, $\tau = L_x(G)$ and $y = \tau^{-1}(n)$. Also let G_l be the subgraph induced by $\{1, 2, \dots, x\}$ and let G_r be the subgraph induced by $\{x, \dots, n\}$. Note that $G_l \in \Psi_x(\psi, p)$ and $G_r \in \Psi_{n-x+1}(\psi, p)$. Next, let $\tau_l = L_x(G_l)$, $\tau_r = L_x(G_r)$ and let $y_l = \tau_l^{-1}(x)$, $y_r = \tau_r^{-1}(n - x + 1)$. The analysis now divides into several cases.

Case 1. $x < n/4$. For any $z \in \{1, \dots, x\}$, Lemma 3.2 implies

$$d(x, z) \leq 2\frac{x}{\psi} < \frac{n}{2\psi} \quad \text{and} \quad d(x, n) \geq \frac{n-x}{\psi} > \frac{3n}{4\psi}$$

Thus, $d(x, z) < d(x, n)$ for any $z < x$. This implies that $y \in \{x, \dots, n\}$. In fact, $y = y_r$. To see this, first note that for $i \geq 0$, $V'_i(x, G_r) \subseteq V'_i(x, G)$ (the notation for the vertices belonging to each level has been extended to distinguish between the two graphs). Furthermore, if $z \in V'_i(x, G_r)$ where $i \geq 4$ then $gp_x(z, G_r) = gp_x(z, G)$ (the notation for the grandparents of z has been similarly extended). Consequently, $y = y_r$. Applying Lemma 4.1 to G_r yields $y > n - 4\alpha$.

Case 2. $n/4 \leq x \leq 3n/4 \wedge y \in \{x, \dots, n\}$. By the same argument used in case 1, $y = y_r$ and applying Lemma 4.1 to G_r yields $y > n - 4\alpha$.

Cases 3,4 are symmetric with 1,2. \square

Proof of Theorem 6.1. Let $x \in V$, $\tau = L_x(G)$ and $y = \tau^{-1}(n)$. By Lemma 6.2, either $y < 4\alpha$ or $y > n - 4\alpha$. Since the two cases are symmetric, we will only discuss the former. Let $\sigma = L_y(G)$, G_r be the subgraph induced by $\{y, \dots, n\}$ and $\sigma_r = L_y(G_r)$. Note that the restriction of σ to $\{y, \dots, n\}$ is the same as σ_r . By Lemma 3.2, every vertex in $\{1, \dots, y-1\}$ is connected to y by a 2-path and no vertex in $\{1, \dots, y-1\}$ is adjacent to any vertex in $V'_3(y)$. Consequently, $\{1, \dots, y-1\} \subseteq V'_1(y)$. Let $\{u, v\} \in E$, and consider the following three cases.

Case 1. $\{u, v\} \subseteq \{y, \dots, n\}$. By Lemma 4.1, $|\sigma_r(u) - \sigma_r(v)| \leq \psi + 8\alpha$. Consequently, $|\sigma(u) - \sigma(v)| \leq \psi + 12\alpha$.

Case 2. $\{u, v\} \subseteq \{1, \dots, y-1\}$. Since $\{u, v\} \subseteq V'_1(y)$ and by Lemma 3.8, $|V'_1(y)| \leq 4\alpha + (\psi + \alpha) = \psi + 5\alpha$, it follows that $|\sigma(u) - \sigma(v)| \leq \psi + 5\alpha$.

Case 3. $u \in \{1, \dots, y-1\}, v \in \{y, \dots, n\}$. Because $u < y, v < y + \psi$. By Lemma 4.1, $|\sigma(v) - v| \leq 4\alpha$, giving $\sigma(v) < y + \psi + 4\alpha < \psi + 8\alpha$. Since $u \in V'_1(y)$, $\sigma(u) \leq \psi + 5\alpha$. Thus, $|\sigma(u) - \sigma(v)| \leq \psi + 8\alpha$.

In all three cases above, we conclude that $|\sigma(u) - \sigma(v)| \leq \psi + 12\alpha = \psi + O(\log n) = \phi(G) + O(\log n)$. \square

The method for selecting a starting vertex outlined above can be refined in several directions. One way is to run *MLA1* several times, each time using the rightmost vertex from the previous run as the starting vertex for the next run. This extends the applicability of the method to larger values of ψ . Another refinement is to run *MLA1* several times as just described, but then take the 4α rightmost vertices from the last run and use these as a set of candidate starting vertices. With high probability, either vertex 1 or vertex n is in this set. The results obtained in this way may be somewhat closer to optimal, but the cost is an extra $O(\log n)$ factor in the running time.

7. Properties of Random Graphs. This section is largely independent and examines several properties of random graphs, particularly graphs in $\Psi_n(\psi, p)$. The following theorem is a special case of a result proved by Erdős and Renyi in [6].

Theorem 7.1. *Let $-1 < \epsilon < 1$ be fixed, $p = (1 + \epsilon)(\ln n)/n$, $G \in \Gamma_n(p)$. If $\epsilon > 0$, G is almost always connected. If $\epsilon < 0$, G is almost always disconnected.*

The following is a similar result for random graphs with small bandwidth.

Theorem 7.2. *Let $-1 < \epsilon < 1$ be fixed, $0 < p < 1$, $\psi = (1 + \epsilon)\lambda(1 - p)/2$, $\psi \rightarrow \infty$. If $\epsilon > 0$ then almost all $G \in \Psi_n(2\psi, p)$ are connected. If $\epsilon < 0$ then almost all $G \in \Psi_n(\psi, p)$ are disconnected.*

To prove Theorem 7.2 we need to introduce another probability distribution and prove two lemmas. Let n and ψ be positive integers, $\psi < n$, $0 < p < 1$, and let $G = (V, E)$ be a random variable defined by the following experiment.

- Let $V = \{1, 2, \dots, n\}$.
- For each pair u, v $1 \leq u < v \leq n$ and $|u - v| \leq \psi \vee |u - v| \geq n - \psi$ include the edge $\{u, v\}$ in E with probability p .

The probability distribution defined by this experiment is denoted $\Psi_n^c(\psi, p)$.

Lemma 7.1. *Let $-1 < \epsilon < 1$ be fixed, $0 < p < 1$, $\psi = (1 + \epsilon)\lambda(1 - p)/2$, $1 \leq \psi \leq n/2$, $G \in \Psi_n^c(\psi, p)$. If $\epsilon > 0$ then G almost always contains no isolated vertex. If $\epsilon < 0$ then G almost always contains at least one isolated vertex.*

Proof. First part $-\epsilon > 0$. Let

$$\begin{aligned} X_v &= \begin{cases} 1 & \text{if } v \text{ is isolated} \\ 0 & \text{if } v \text{ is not isolated} \end{cases} \\ X &= X_1 + X_2 + \dots + X_n \\ \mu &= E(X) = \sum_{v=1}^n E(X_v) = n(1 - p)^{2\psi} \end{aligned}$$

Then,

$$P(X \geq 1) \leq \mu = n(1 - p)^{2\psi} = n^{-\epsilon} \rightarrow 0$$

This completes the proof of the first part.

Second part $-\epsilon < 0$. Let X, X_1, \dots, X_n be defined as before.

$$\begin{aligned} E(X^2) &= \sum_{u=1}^n \sum_{v=1}^n E(X_u X_v) = \sum_{u=1}^n \sum_{v=1}^n P(u \text{ and } v \text{ are both isolated}) \\ &= n(1 - p)^{2\psi} + 2\psi n(1 - p)^{4\psi-1} + n(n - 2\psi - 1)(1 - p)^{4\psi} \end{aligned}$$

By Chebyshev's inequality,

$$P(X = 0) \leq \frac{\sigma^2}{\mu^2} = \frac{E(X^2) - \mu^2}{\mu^2} = \frac{1}{\mu} + \frac{2\psi p}{n(1 - p)} - \frac{1}{n} < n^\epsilon + (1 + \epsilon) \frac{p \ln n}{n(1 - p) \ln(1/(1 - p))}$$

The function $\frac{p}{(1-p)\ln(1/(1-p))}$ gets large as $p \rightarrow 1$. However, $1 \leq \psi < (1+\epsilon)\lambda(1-p)$
 $\Rightarrow p \leq 1 - n^{-(1+\epsilon)}$. Hence,

$$P(X = 0) < n^\epsilon + (1+\epsilon)\frac{\ln n}{n^{-\epsilon} \ln n^{1+\epsilon}} = 2n^\epsilon \rightarrow 0 \quad \square$$

Let $D(G)$ denote the diameter of G .

Lemma 7.2. *Let $0 < \epsilon \leq p < 1$ where ϵ is fixed, and let $G \in \Gamma_n(p)$. Then*

$$P(D(G) > 2) \leq \binom{n}{2}(1-\epsilon^2)^{n-2}.$$

Proof. Let u and v be any two vertices in G . The number of possible 2-paths between them is $n-2$ and the probability that any one of them is absent is $1-p^2$. Hence the probability that u and v are not connected by a 2-path is $(1-p^2)^{n-2}$. Consequently, the probability that any pair of vertices is not connected by a 2-path is

$$\leq \binom{n}{2}(1-p^2)^{n-2} \leq \binom{n}{2}(1-\epsilon^2)^{n-2} \quad \square$$

Proof of Theorem 7.2. First part $-\epsilon > 0$. G is connected if the first 2ψ vertices induce a connected subgraph and all other vertices have at least one edge to a lower numbered vertex. By Lemma 7.2, if $p > \alpha$ for some $\alpha > 0$ then the probability that the first 2ψ vertices induce a subgraph of diameter greater than two is

$$\leq \binom{2\psi}{2}(1-\alpha^2)^{2\psi-2} \rightarrow 0$$

Hence, if p is bounded below, the first 2ψ vertices almost always induce a connected subgraph. If on the other hand $p \rightarrow 0$ we must use Theorem 7.1 to establish that the first 2ψ vertices induce a connected subgraph. This requires that we show that there exists some $\gamma > 0$ such that $p \geq (1+\gamma)(\ln(2\psi))/(2\psi)$. From the hypothesis of the theorem

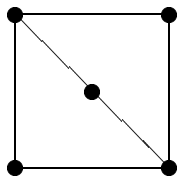
$$\frac{p(2\psi)}{\ln(2\psi)} = (1+\epsilon)\frac{p \ln n}{\ln(1/(1-p)) \ln(2\psi)} > (1+\epsilon/2)$$

for large enough n since $p \rightarrow \ln(1/(1-p))$ as $p \rightarrow 0$ and $n > 2\psi$. Now, the probability that any of the remaining vertices have no edges to lower numbered vertices is $< n(1-p)^{2\psi} = n^{-\epsilon} \rightarrow 0$. This completes the proof of the first part of Theorem 7.2.

Now let $\epsilon < 0$ and let $G' \in \Psi_n^c(\psi, p)$. Clearly, $P(G \text{ is connected}) \leq P(G' \text{ is connected})$ and since by Lemma 7.1, G' is almost always disconnected, it follows that G is almost always disconnected. \square

A simple lower bound for the bandwidth of any connected graph is given by

$$\phi(G) \geq \omega(G) = \left\lceil \frac{n-1}{D(G)} \right\rceil$$



$$\phi(G) = 3 \quad \omega^*(G) = 2$$

Figure 10: Graph Showing $\omega^*(G) \neq \phi(G)$

since the first and last vertices in any optimal layout are connected by a path of length at most $D(G)$ and hence at least one edge in this path has length $g \leq \omega(G)$. Chvatal [4] was apparently the first to notice this. A more general lower bound is given by

$$\phi(G) \geq \omega^*(G) = \max_{G'} \omega(G')$$

where G' ranges over all connected subgraphs of G . The graph shown in Figure 10 shows that $\omega^*(G) \neq \phi(G)$ in general. It is natural to ask if there is any constant c such that for all connected graphs $\phi(G) \leq c\omega^*(G)$. Ronald Graham has pointed out that this is not the case. The argument is given in [11]. In spite of this result however, we can show that if $\ln n = o(\psi)$, then for almost all $G \in \Psi_n(\psi, p)$,

$$D(G) < (1 + \epsilon) \frac{n}{\phi(G)} + 5.$$

Theorem 7.3. *Let $\epsilon > 0$, $0 < p < 1$ be fixed, $\alpha = (1 + \epsilon)\lambda(1 - p^2) \leq \psi < n$. For almost all $G \in \Psi_n(\psi, p)$ $D(G) < \frac{n}{\psi - \alpha/3} + 5$.*

Proof. By Lemma 3.7, there exists a path $Q = (v_0, \dots, v_{3r})$ that satisfies

$$v_0 = 1 \quad v_{3r} \geq n - 3\psi \quad v_{3(i+1)} \geq v_{3i} + 3\psi - \alpha \quad 0 \leq i \leq r$$

Since $n \geq v_{3r}$ and $v_{3r} > 3\psi r - \alpha r$, $3r < \frac{n}{\psi - \alpha/3}$.

By Lemma 3.2, any vertex $u < v_{3r}$ is connected by a 2-path to some vertex in $\{v_1, \dots, v_{3r}\}$, and any vertex $u > v_{3r}$ is connected by a 4-path to v_{3r} . Thus, every pair of vertices is joined by a path of length at most $3r + 5$. \square

By Lemma 7.2, if $p \geq \epsilon > 0$ then for almost all $G \in \Gamma_n(p)$, $D(G) = 2$. When p is allowed to approach zero as n gets large the diameter can become larger. By Theorem 7.1, when p is much less than $(\ln n)/n$ the graph is likely to be disconnected. We now consider the probable diameter of random graphs in $\Gamma_n(p)$ when $p = (c \ln n)/n$ and c is a constant. We do this by examining the probable size of V_1, V_2, \dots . Let $n_i = |V_i|$. Clearly,

$$n_0 = 1$$

k	\hat{n}_k	$(np)^k$
0	1	1
1	28	28
2	763	763
3	20,850	21,096
4	428,450	582,890
5	549,910	—

Figure 11: Comparison of \hat{n}_k with $(np)^k$ for $n = 10^6$, $p = (2 \ln n)/n$

$$\begin{aligned}
n_2 &\in \mathbf{B}(n-1, p) \\
n_3 &\in \mathbf{B}(n-(n_1+1), 1-(1-p)^{n_1}) \\
&\dots \\
n_{k+1} &\in \mathbf{B}(n-s_k, 1-(1-p)^{n_k})
\end{aligned}$$

where $s_k = \sum_{j=0}^k n_j$. Define $\hat{n}_0 = 1$, $\hat{n}_{k+1} = (n - \hat{s}_k)(1 - (1-p)^{\hat{n}_k})$, where $\hat{s}_k = \sum_{j=0}^k \hat{n}_j$. We can use \hat{n}_k as an estimator for n_k . Figure 11 gives values of \hat{n}_k for particular values of n and p . The sequence grows very rapidly until a large fraction of the vertices in the graph has been ‘captured’. Then the remaining vertices are taken in the last step. The figure also gives values of the function $(np)^k$. For $k \leq 3$, $(np)^k$ gives an excellent estimate for \hat{n}_k .

Let k^* be such that $s_{k^*} = n$. In the following we show that for $k \leq k^* - 2$, $n_k > (np/8)^k$ with high probability. We can use this to get a probabilistic upper bound on k^* and hence on $D(G)$. The main results are

Theorem 7.4. *Let $c > 8$ be fixed, $p = (c \ln n)/n$, $\gamma = np/8$. For almost all $G \in \Gamma_n(p)$, $1 \leq k \leq k^* - 2 \Rightarrow n_k > \gamma^k$.*

Theorem 7.5. *Let $c > 8$ be fixed, $p = (c \ln n)/n$, $\gamma = np/8$. For almost all $G \in \Gamma_n(p)$,*

$$D(G) \leq 2 \left(\left\lceil \frac{\lceil \ln(1/p) \rceil}{\ln \gamma} \right\rceil + 2 \right).$$

The proof of Theorem 7.4 is contained in the following lemmas.

Lemma 7.3. *Let $c > 8$ be fixed, $p = (c \ln n)/n$, $\gamma = np/8$. For almost all $G \in \Gamma_n(p)$ $1 \leq k \leq k^* - 2 \wedge n_{k-1} < 1/p \wedge s_{k-1} \leq n/2 \Rightarrow n_k > \gamma n_{k-1}$.*

Proof. Since $n_k \in \mathbf{B}(n - s_{k-1}, 1 - (1-p)^{n_{k-1}})$,

$$\bar{n}_k = E(n_k) = (n - s_{k-1})(1 - (1-p)^{n_{k-1}}) \geq \frac{n}{2} p n_{k-1} (1 - p n_{k-1}/2) > \frac{np}{4} n_{k-1} = 2\gamma n_{k-1}$$

By Proposition 3.1

$$P(n_k \leq \gamma n_{k-1}) \leq P(n_k \leq \bar{n}_k/2) < e^{-\bar{n}_k/8} < e^{-\gamma n_{k-1}/4}$$

Let A_k denote the event $n_k \leq \gamma n_{k-1}$. The probability that there exists a k satisfying the hypothesis of the lemma, such that A_k holds is

$$\leq P(A_1) + P(A_2 | \bar{A}_1) + \cdots + P(A_{k^*-2} | \bar{A}_1 \cdots \bar{A}_{k^*-3}) \quad (3)$$

$$\leq e^{-\gamma/4} + e^{-\gamma^2/4} + \cdots + e^{-\gamma^{k^*-2}/4} \rightarrow 0 \quad (4)$$

□

Lemma 7.4. *Let $c > 8$ be fixed, $p = (c \ln n)/n$. For almost all $G \in \Gamma_n(p)$ $1 \leq k \leq k^* - 2 \wedge s_{k-1} \leq n/2 \Rightarrow n_{k-1} < 1/p$.*

Proof. Assume that $n_{k-1} \geq 1/p$. Then since $n_k \in \mathbf{B}(n - s_{k-1}, 1 - (1-p)^{n_{k-1}})$,

$$\bar{n}_k = E(n_k) = (n - s_{k-1})(1 - (1-p)^{n_{k-1}})g \leq \frac{n}{2}(1 - 1/e) > n/4$$

By Proposition 3.1

$$P(n_k \leq n/8) \leq P(n_k \leq \bar{n}_k/2) < e^{-\bar{n}_k/8} < e^{-n/32} \rightarrow 0$$

Hence, assume $n_k > n/8$. Then the probability that any of the remaining vertices is not adjacent to something in V_k is

$$\leq (n - s_k)(1-p)^{n_k} < ne^{-np/8} = n^{1-c/8} \rightarrow 0$$

This implies that $k^* \leq k + 1$ which is a contradiction. □

Lemma 7.5. *Let $c > 8$ be fixed, $p = (c \ln n)/n$. For almost all $G \in \Gamma_n(p)$, $1 \leq k \leq k^* - 2 \Rightarrow s_{k-1} \leq n/2$.*

Proof. Assume that $s_{k-1} > n/2$ and let k' be the smallest integer such that $s_{k'} > n/2$. By Lemma 7.4, $n_{k'-1} < 1/p$ and by Lemma 7.3, for all $k \leq k'$, $n_k > \gamma n_{k-1}$, where $\gamma = np/8$. Since for large n , $\gamma > 2$, we have $s_k > 2s_{k-1}$ for $k \leq k'$. Consequently $n_{k'} = s_{k'} - s_{k'-1} > s_{k'}/2 > n/4$. Now, the probability that any of the vertices in $V - (V_0 \cup V_1 \cup \cdots \cup V_{k'})$ is not adjacent to some vertex in $V_{k'}$ is

$$< (n - s_{k'})(1-p)^{n_{k'}} < ne^{-np/4} = n^{1-c/4} \rightarrow 0$$

This implies that $k^* \leq k' + 1$ which is a contradiction. □

This establishes Theorem 7.4.

Proof of Theorem 7.5. Note that $D(G) \leq 2k^*$. Let k' be the smallest integer such that $\gamma^{k'} \geq 1/p$. Clearly $k' = \left\lceil \frac{\ln(1/p)}{\ln \gamma} \right\rceil$. If $k' > k^* - 2$, we're done. If $k' \leq k^* - 2$ we can apply Theorem 7.4 giving $n_{k'} \geq 1/p$. By the argument used in the proof of Lemma 7.4, this implies $k^* \leq k' + 2$. □

8. Conclusions. The work reported here is part of an ongoing research effort aimed at developing better methods for evaluating the performance of heuristic algorithms for hard combinatorial problems. This is an area where the usual analytical tools often fail us,

and the available results are unsatisfying. To be useful, a performance evaluation method must satisfy two basic criteria. First, it must be able to explain the practical success of popular algorithms and the differences observed between competing algorithms. Second, it should provide insight suggesting new and better algorithms, and supply a basis for making predictions about their success in practice. The ultimate utility of such a method depends on how accurately it predicts the performance of algorithms in real applications.

Worst-case analysis is inadequate for evaluating the performance of heuristics for bandwidth minimization, precisely because it fails to satisfy the criteria given above. As shown in Theorems 2.1 and 2.2, even probabilistic analysis can be of little use if one is naive in choosing the probability distribution. The key to the work reported here is in the choice of distribution. Because $\Psi_n(\psi, p)$ generates only graphs having bandwidth $\leq \psi$, we can explore properties that are common to most such graphs, even though they may be rare among unrestricted graphs. The success of heuristics like the level algorithms is due to the fact that they exploit these properties.

The methods used in this paper at least partially satisfy the criteria outlined above. They provide the first satisfactory analytical explanation of the practical success of the level algorithms and they provide insight leading to methods, which at least in theory are better. If the modified level algorithms fare as well in practice as they do on paper, the utility of these methods will have been demonstrated.

Acknowledgements. I want to thank my good friend and Ph.D. advisor, Hal Sudborough, who supervised the research reported here. I also wish to thank two anonymous referees whose careful reading and thoughtful comments uncovered several errors in an early draft and led to significant improvements in the presentation.

References

- [1] Angluin, D., L. G. Valiant. "Fast Probabilistic Algorithms for Hamiltonian Circuits and Matchings". In *Journal of Computer and System Sciences* 18, 155-193, 1979.
- [2] Cheng, K. Y. "Minimizing the Bandwidth of Sparse Symmetric Matrices". In *Computing* 11, 103-110, 1973.
- [3] Cheng, K. Y. "Note on Minimizing the Bandwidth of Sparse Symmetric Matrices". In *Computing* 11, 27-30, 1973.
- [4] Chvatal, V. "A Remark on a Problem of Harary". In *Czechoslovak Math Journal* 20, 95, 1970.
- [5] Cuthill, E., J. McKee. "Reducing the Bandwidth of Sparse Symmetric Matrices". In *ACM National Conference Proceedings* 24, 157-172, 1969.
- [6] Erdos, P., A. Renyi. "On Random Graphs I.". In *Publicationes Mathematicae*, 290-297, 1959.

- [7] Garey, Michael R., R. L. Graham, David S. Johnson, D. E. Knuth. "Complexity Results for Bandwidth Minimization". In *SIAM Journal of Applied Mathematics* 34 , 477-495, 5/78.
- [8] Monien, B., I. H. Sudborough. "Bandwidth Problems in Graphs". In *Proceedings 18th Annual Allerton Conference on Communication, Control, and Computing*, 650-659, 1980.
- [9] Papadimitriou, Christos H. "The NP-Completeness of the Bandwidth Minimization Problem". In *Computing* 16, 263-270, 1976.
- [10] Saxe, James B. "Dynamic Programming Algorithms for Recognizing Small-Bandwidth Graphs in Polynomial Time", Carnegie-Mellon University Technical Report, 1980.
- [11] Turner, Jonathan S. "Bandwidth and Probabilistic Complexity", Northwestern University, Ph.D. thesis, 6/82.