The Beacon Set Approach to Graph Isomorphism

Richard J. Lipton

Research Report #135

May 1978

This research was supported in part by NSF MCS 76-81486.

Abstract: The new concept of a "basis" for a graph makes it possible to obtain fast isomorphism tests and succinct certificates for many classes of graphs including random graphs and symmetric graphs.

Keywords: beacons, graphs, isomorphism, probabilistic algorithms, random graphs, succinct certificates, symmetric graphs

1. *Introduction*

While the *graph isomorphism problem* has many applications [3,6], it is fascinating to computer scientists (i.e. me) because unlike many other combinatorial problems it has not yet been proved intractable in the sense that it has not yet been proved to be NP-complete [1,15]. Yet no polynomial time algorithm is known for it.

Denote by $I_{\mathcal{G}}(n)$ the "best" running time of any algorithm for testing pairs of graphs with n vertices from the class $\mathcal{G}$ for isomorphism; I(n) with no subscript means all graphs. Since our current knowledge about isomorphism is incomplete -- and perhaps mine is even more so -- only upper bounds and not exact results can be obtained for $I_{\mathcal{G}}(n)$. I believe, however, that the new bounds given here on $I_{\mathcal{G}}(n)$ are interesting in that they supply further insights into the growth of this function.

In addition to $I_{\mathcal{G}}(n)$, it is important to study $C_{\mathcal{G}}(n)$, the "best" running time of any algorithm that computes certificates (sometimes called codes) of graphs with n vertices from the class $\mathcal{G}$. An algorithm computes certificates if it assigns an integer to each graph that is unique to that graph up to isomorphism. This concept arises naturally in many applications such as chemical compound identification. Let, for example, $G_1,\ldots,G_m$ be n-vertex graphs from $\mathcal{G}$. Then the isomorphic ones can be determined in

$$\binom{m}{2} \cdot I_{\mathcal{G}}(n) \text{ time}$$

by using the isomorphism test, but in

$$m \cdot C_{\mathcal{G}}(n) \text{ time}$$

by using certificates. Clearly, if m is very large and $C_{\mathcal{G}}(n)$ is about equal to $I_{\mathcal{G}}(n)$, then certificates afford a large savings.

Certificates are also related simply to isomorphism tests:

(1) $$C_{\mathcal{G}}(n) \geq I_{\mathcal{G}}(n).$$

In order to see this, let G and H be graphs in $\mathcal{G}$ with n vertices. Then by definition, G is isomorphic to H if and only if they have equal certificates. This almost demonstrates (1), but not quite. Actually, to be quite accurate (1) should be written as $c \cdot C_{\mathcal{G}}(n) \geq I_{\mathcal{G}}(n)$ for some constant $c > 0$, but in general such constants will be suppressed. An important open question is the partial converse of (1), i.e. for what function f does

(2) $$f(I_{\mathcal{G}}(n)) \geq C_{\mathcal{G}}(n)?$$

It would be interesting if f could be a polynomial. It is curious that all methods known -- at least known to me -- satisfy (2) with f indeed a polynomial.

In order to compare the new results obtained here with the previous results it may be useful first to review what is currently known. A brute force search yields $I(n) \leq n^2 \cdot n!$; with more care this can be improved to

(3) $$C(n) \leq n^2 \cdot n!.$$

The key to this result is to use the lexicographically first adjacency matrix as the certificate of the given graph. No better result is known for all graphs. It is widely believed, however, that (3) can be improved to

(4) $$I(n) \leq 2^{cn} \text{ or even to } I(n) \leq n^{c\log n}.$$

(Each c denotes a constant, but the two constants are not necessarily the same.) The latter improvement would be especially interesting since it would show that graph isomorphism is quite unlikely to be NP-complete, because if

$I(n) \le n^{c\ell ogn}$ and yet graph isomorphism is NP-complete, then all of NP would be do-able in time $n^{c'\ell ogn}$, which would be quite unexpected.

The literature [4,6,23] contains several methods of actually doing graph isomorphism. Most of these methods are based either on backtracking or heuristics and hence do not yields any bounds on $I_G(n)$ or $C_G(n)$. I will therefore exclude them from this discussion. Indeed, recent work [21] suggests that methods using heuristics that depend on such graph invariants as degrees and neighborhood structure will fail in general.

For certain classes of graphs, results are known that are better than brute force. For example, if G_1 is the class of cubic graphs, then [21]

(5) $$I_{G_1}(n) \le n^2 \cdot 2^n.$$

It appears that a similar result holds for certificates, i.e. for $C_{G_1}(n)$. More encouraging results are those on G_2, the class of trees, and G_3, the class of planar graphs [13,14]:

(6) $$C_{G_2}(n) = O(n) \text{ and } I_{G_3}(n) = O(n).$$

Again it appears that $C_{G_3}(n) = O(n)$, but I am unaware of any proof of this, although I have seen results that seem to imply it [14]. The separator theorem [18] implies a much weaker result for G_4, the class of graphs of any fixed genus:

(7) $$I_{G_4}(n) \le 2^{c\sqrt{n}}.$$

I conjecture but cannot prove that (7) can actually be improved to $I_{G_4}(n) \le n^c$.

Recently attention has focussed on the class of random graphs. Let $I(n,\lambda)$ denote the best running time of any algorithm that determines whether

$\underset{\sim}{G}$ and H are isomorphic, where $\underset{\sim}{G}$ is a random graph and H, which is selected by our worst enemy, is a function of $\underset{\sim}{G}$. Moreover, $I(n,\lambda) \le f(n)$ is valid if the algorithm fails to run in time $f(n)$ for at most $\lambda \cdot 2^{\binom{n}{2}}$ graphs on n vertices. (Here $\lambda = \lambda(n)$ is a function of n.) In a similar way, define $C(n,\lambda)$ for certificates. Then Babi and Erdös [2] show that

(8) $$C(n,\lambda) \le n^2 \text{ where } \lambda = n^{-\frac{1}{7}}.$$

Independently, Karp [16] shows that

(9) $$C(n,\lambda) \le n^c \text{ where } \lambda \to 0 \text{ exponentially fast}$$

and Theorem 5 in this paper shows that

(10) $$C(n,\lambda) \le n^c \text{ where } \lambda \to 0 \text{ exponentially fast.}$$

These results must be considered with the folklore of this area in mind. Most researchers seem to believe that random graphs are by no means the worst case.

The last result (10) on random graphs rests on a new approach to the graph isomorphism problem. The key to the approach is the concept of a *beacon set* (see section 3). Use $B_G(n)$ to denote the worst-case size of the beacon set of any n-vertex graph in G. Then

(11) $$C_G(n) \le k \cdot n^{k+2} \text{ where } k = B_G(n)$$

is the main observation behind the new results here. Clearly, if $B_G(n)$ is "small" then so is $C_G(n)$. For example, (10) is proved by showing that a random graph has a beacon set of size 4 with probability → 1 exponentially fast. In passing, note that there has been a definition of this concept without equation (11) and then only as applied to trees [12]. Curiously, while I will show in a moment that $B_G(n)$ is small for many classes of graphs, $B_{G_2}(n)$ -- recall that G_2 is the class of trees -- is $\frac{n}{2} + O(1)$.

Section 3 develops the beacon method; here I will confine myself to listing some of the applications of this method.

One application, which I have been working on with K. Booth, is

(12) $$C(n,\lambda) \leq n^{c\ell ogn} \text{ where } \lambda = \frac{1}{n!} .$$

Another way of saying this is that there is an algorithm for computing certificates that runs in time $n^{c\ell ogn}$ on the average. This is stronger than results (8), (9), and (10), since these results imply the existence of an algorithm that works well only on a large fraction of all graphs -- on the remaining graphs it may do so badly that its average behavior is n!.

Actually, the beacon set method implies a much stronger result. Let $C^*(n,\lambda)$ denote the running time of the best algorithm that computes the certificates of a random graph $\underset{\sim}{G}$ with the understanding that our worst enemy is allowed to add as many edges to $\underset{\sim}{G}$ or delete as many from $\underset{\sim}{G}$ as he wishes, provided only that he does not add or delete more than o(n) edges incident to any one vertex. Then (12) can be improved to

(13) $$C^*(n,\lambda) \leq n^{c\ell ogn} \text{ where } \lambda = \frac{1}{n!} .$$

Thus there is an algorithm that computes certificates in time $n^{c\ell ogn}$ on the average even if our enemy can change a large portion of the graph.

One of the folk beliefs of this area is that highly regular graphs, not random graphs, are the worst case for isomorphism. Miller [19] has addressed this issue by showing that G_5, the class of cubic symmetric graphs, have certificates that are computable in NP $\cap$ co-NP. But with the current state of knowledge about NP, this does not even improve the brute force bound of (5).

The beacon set method, however, shows that

(14) $$C_{G_5}(n) \le n^{c\ell ogn}$$

and

(15) $$C_{G_6}(n) \le n^{c\ell ogn}$$

where G_6 is the class of 2-symmetric or distance transitive graphs of a fixed degree. While this falls short of polynomial bounds, it does demonstrate that symmetric graphs may not be the worst case for isomorphism. Moreover, it is important to note that (13), (14), and (15) show that the beacon set method does not depend on the local irregularity of the degrees of vertices, in contrast to the results of Babi-Erdős and Karp. A striking way to see this is to observe that the ground rules of (13) allow enough edges to be added to a random graph (with probability → 1) to make it regular. This is why the beacon set is so powerful. Indeed, I conjecture that $C_{G_7}(n,\lambda) \le n^{c\ell ogn}$ where G_7 is the class of random regular graphs and $\lambda \to 0$ exponentially fast.

The remainder of this paper is the presentation of the details of the beacon set method. Several further applications of the method are obtained, the most interesting of which is perhaps

(16) $$C_G(n) \le 2^{cn(\ell og^2 n)/d}$$

where either (i) G contains only graphs of degree at least d and girth ≥ 5 or (ii) G contains only graphs of degree at least d and for any two vertices $x \ne y$ the number of vertices adjacent to only x or only y is at least εd for some $\varepsilon > 0$. In particular, this result has applications to cages and strongly regular graphs, again showing that these graphs are not the worst case.

2. *Definitions and Notation*

This paper follows the notation of Erdös and Spencer [9]. If V is a set, then

$$|V| = \text{cardinality of } V$$
$$[V]^2 = \{X\text{: } X \subseteq V \text{ and } |X| = 2\}$$
$$[n] = \{1,2,3,\ldots,n\}.$$

A *graph* G on a set V is a subset of $[V]^2$. The elements of G are the *edges*; the elements of V are the *vertices*. Two vertices x and y are *adjacent* if they are edges of a graph G on V, that is, if $\{x,y\} \in G$. The *degree* of a vertex is the number of vertices it is adjacent to. Two graphs G and H are *isomorphic* $(G \simeq H)$ if there is a function ϕ from the vertices of G to those of H such that $\{x,y\} \in G$ if and only if $\{\phi(x),\phi(y)\} \in H$.

We also need the notions of distance and walks in graphs. A *walk of length* m *from* x *to* y in graph G is a sequence of vertices $v_1,\ldots,v_{m+1}$ such that $x = v_1$, $y = v_{m+1}$, and, for $1 \le i \le m$, $\{v_i,v_{i+1}\} \in G$. Let $w_m(x,y)$ denote the *number of walks of length* m *from* x *to* y in G, and let $d(x,y)$ be the *distance from* x *to* y, which is defined as either the smallest m such that $w_m(x,y) > 0$ or ∞ if no such m exists.

Random variables will be denoted $\underset{\sim}{G}$, $\underset{\sim}{X}$, etc. (The wavelet is the typewritten equivalent of boldface.) $\underset{\sim}{G}_n$ will be the random variable whose value is a graph on [n]. If $\{x,y\} \in [n]^2$, then

$$\text{Prob}[\{x,y\} \in \underset{\sim}{G}_n] = \frac{1}{2}$$

and these probabilities are independent for different edges.

The o,O notation is standard. In general, n will be used for the number of vertices in the graph under discussion. The random-access model of computation is standard [1].

3. *Beacons and the Basis of a Graph*

The notion of the basis of a graph is easy enough to grasp intuitively. Imagine that you are an amnesiac pilot who suddenly comes to and finds himself flying an airplane high over unknown, fogbound territory. Where are you? Where are you going? You can't figure out where you are from knowing where you took off and how long and in what direction you've been flying -- you have amnesia. You can't use landmarks to determine your location -- the view is featureless. But luckily there are several transmission stations, or beacons, transmitting signals that you can pick up with your instruments. You don't know where the beacons are, since they identify themselves only by call letters, but the one fact you remember in your amnesia is that all beacons transmit at one given strength. So by using your instruments you can determine accurately how far you are from any beacon. That means that as long as there are at least three beacons and they are not collinear you can determine the unique intersection of their signals at any time; hence you can navigate.

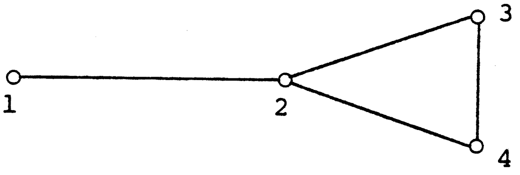
The concept of the basis of a graph is a generalization of this strategy for navigating. A set of vertices of a graph is a *basis* if each vertex can be uniquely named by computing its "distance" to a set of "beacons"; but the notion of distance is generalized -- f(x,y) is any function defined on pairs of vertices of a graph, with d(x,y) one case of f(x,y). Now the critical question becomes how many beacons you need, since smaller sets will imply faster isomorphism tests and smaller certificates.

Note in the precise definition of this concept that f(x,y) need not be a metric.

Definition: A set of vertices $p_1,\ldots,p_k$ is an *f-beacon set* for the graph G if for all vertices $x \neq y$ in G there is a p_i such that $f(x,p_i) \neq f(y,p_i)$.

In other words, $p_1,\ldots,p_k$ is an f-beacon set provided no two distinct vertices are assigned the same names where the *name* of a vertex x is defined as $(f(x,p_1),\ldots,f(x,p_k))$. The vertices $p_1,\ldots,p_k$ will be called *beacons*.

Perhaps at this point it would be worthwhile to present a simple example. Consider the graph



Now {1,2} is not a beacon set with respect to $f(x,y) = d(x,y)$, while {1,4} is. {1,2} fails since $d(3,1) = d(4,1)$ and $d(3,2) = d(4,2)$.

It is now possible to demonstrate how graphs with small beacon sets have fast isomorphism tests and succinct certificates. Assume for simplicity that all beacon sets are *distance beacon sets*, beacons with respect to $f(x,y) = d(x,y)$. The extension to other functions f is straightforward and will be done in section 4.

Lemma 1: $I_G(n) \leq c\cdot k\cdot n^{k+2}$ where $k = B_G(n)$.

Proof: Let G and H be graphs with n vertices and let G have a distance beacon set of size k. An algorithm first computes the distance functions $d_G(x,y)$ and $d_H(x,y)$. Next it searches all lists of k vertices of G until it finds a distance beacon set, say $p_1,\ldots,p_k$. At the same time, it finds the names of all the vertices of G; let $n_G(x)$ be the name of the vertex x. Then the algorithm operates on all the lists $q_1,\ldots,q_k$ of k vertices of H as follows:

1. First it determines whether $q_1,\ldots,q_k$ is a distance beacon set.

2a. If it is not, then the next list of k vertices of H is processed; if none remain, then G and H are not isomorphic.

2b. If it is, the names of all the vertices of H are computed; let $n_H(y)$ be the name of vertex y.

3. Now it checks whether $\{n_G(x): x \text{ vertex } G\} = \{n_H(y): y \text{ vertex } H\}$.

4a. If not, then the next list of k vertices of H is processed; again, if none remain, then G and H are not isomorphic.

4b. If so, then on to step 5.

5. Finally, the algorithm checks whether ϕ is an isomorphism, where $\phi(x)$ = the unique y in H with $n_G(x) = n_H(y)$.

6a. If it is not, it processes the next list of vertices, as before.

6b. If it is, then G is isomorphic to H.

Now the proof depends on whether this algorithm is correct. Clearly, if it ever answers that G is isomorphic to H then it must be correct. Assume therefore that it answers that G and H are not isomorphic when they actually are isomorphic. Let ϕ be the function that is the isomorphism from G to H, and let $p_1,\ldots,p_k$ be the distance beacon set selected for G by the algorithm. Then $q_1,\ldots,q_k$ is also a distance beacon set for H where $q_i = \phi(p_i)$ for $i = 1,\ldots,q_k$. This follows directly from the definition of a distance beacon set and the fact that the distance between vertices is invariant under isomorphism. Then $n_G(x) = n_H(\phi(x))$ and so the algorithm would answer that G is isomorphic to H; but the algorithm answered that they are not isomorphic. This contradiction proves that the algorithm is correct.

Now it is possible to estimate the running time of this algorithm on a random-access machine. The distance function $d_G(x,y)$ and $d_H(x,y)$ can be obtained in time $O(n^3)$ by a shortest-path algorithm [1]. Since the distances are all precomputed, the test to see whether $p_1,\ldots,p_k$ is a beacon set can be done in $O(kn + kn\ell ogn)$ time (ℓogn is the logarithm of n to base 2; ℓnn is the natural logarithm of n) -- the name of each vertex can be obtained in time $O(kn)$ and then uniqueness can be checked in time $O(kn\ell ogn)$. Then n^k cases are executed, and each requires three tasks. First, the algorithm has to check whether $q_1,\ldots,q_k$ is a beacon set and, if it is, compute all the names. Second, it must check whether the sets of names are equal. Third, it must verify that ϕ is indeed an isomorphism. As before, the first task can be done in $O(kn\ell ogn)$. The second clearly can be done in time $O(kn\ell ogn)$ by sorting. And the last task can be done in time $O(kn^2)$. Thus the entire algorithm can be done in time at most

$$O(n^3) + O(n^k kn\ell ogn) + O(n^k[kn\ell ogn + kn^2])$$

or at most $O(kn^{k+2})$ time. □

It should be clear that this lemma can be generalized to other "distances" pro ided (1) that they are graph invariants and (2) that they can be computed efficiently.

Now, the concept of a beacon set can actually lead to certificates for graphs. A function $F(G)$ on graphs is a *certificate* when

$$G \simeq H \text{ if and only if } F(G) = F(H).$$

As Miller points out [19], certificates are needed to answer questions of the form

"Is G isomorphic to any of the graphs $H_1,\ldots,H_m$?"

Such questions often arise, for instance, in the identification of chemical compounds. Certificates reduce this question to the standard search problem:

"Is $F(G)$ in the set $\{F(H_1),\ldots,F(H_m)\}$?"

Provided $F(G)$ and $F(H_1),\ldots,F(H_m)$ can all be efficiently computed, this question is much more desirable than the first one. Of course, this last constraint is the key: We must be able to compute $F(G)$ quickly. So in this paper, let's say that a class of graphs has *succinct certificates* if there is a certificate function for the class that can be computed quickly; this is a slight abuse of terminology, but a useful one.

Definition: The vertices of G are $[n]$ and, as usual, the *adjacency matrix* of G is the $n \times n$ matrix A whose (i,j) entry is 1 if $\{i,j\} \in G$ and 0 otherwise. Now the *distance certificate*, $F_d(G)$, can be defined as follows: Let k be the size of the smallest distance beacon set of G. Let $B_1,\ldots,B_m$ be all the distance beacon sets of G. For each B_i $(i = 1,\ldots,m)$, let A_i be the adjacency matrix of G_i where G_i is the graph obtained from G by reordering its vertices according to their names with respect to the beacon set B_i in, say, lexicographic order. Then $F_d(G)$ is defined as A_j where A_j is the lexicographically first $n \times n$ matrix in the set $\{A_1,\ldots,A_m\}$.

The key facts about the distance certificate are contained in the next lemma.

Lemma 2: $C_G(n) \le c \cdot k \cdot n^{k+2}$ where $k = B_G(n)$.

Proof: The algorithm is essentially the same as in Lemma 1. The size of the smallest beacon set, k, is found by trying all lists $p_1,\ldots,p_\ell$ of the vertices of G $(\ell = 1,2,\ldots)$ until a beacon set is found. The time bound for this is

$$\sum_{\ell=1}^{k} O(\ell n^{\ell+2}),$$

which is bounded by $O(kn^{k+2})$. The time to form the adjacency matrices $A_1,\ldots,A_m$ is bounded by $O(m(kn\ell ogn + kn^2))$ and is therefore $O(kn^{k+2})$, since $m \le n^k$. And the time required to find the lexicographically first adjacency matrix A_j is at most $O(m\ell ogm\ n^2)$, or $O(kn^{k+2}\ell ogn)$.

Clearly, if $F_d(G) = F_d(H)$, then G and H have the same adjacency matrix up to a permutation and so are isomorphic. Assume that G is isomorphic to H via the function ϕ but that $F_d(G) \ne F_d(H)$. Let $B_1,\ldots,B_m$ be all the size-k beacon sets of G and $B_1',\ldots,B_m'$ be all the size-k' beacon sets of H with k and k' minimal. Since G is isomorphic to H, it follows that m = m' and k = k'. Let A_i be the adjacency matrix of G sorted according to B_i and A_i' be the adjacency matrix of H sorted according to B_i' $(i = 1,\ldots,m)$. By renaming if necessary, we can assume that $\phi(b_i) = B_i'$ $(i = 1,\ldots,m)$. But then $A_i = A_i'$ since the names of vertices are preserved by ϕ. But then the lexicographically first matrix in $\{A_1,\ldots,A_m\}$ must equal the first in $\{A_1',\ldots,A_m'\}$. This contradiction shows that $F_d(G) = F_d(H)$. □

These two lemmas demonstrate the importance of determining whether a graph has a small beacon set. The next lemma makes it possible to prove the existence of small beacon sets for many classes of graphs. Again let $f(x,y)$ denote an arbitrary "distance" function.

Lemma 3: $B_{\mathcal{G}}(n) \le k$ provided for all $G \in \mathcal{G}$ and all x,y distinct vertices in G

$$(*) \quad |\{p\colon f(x,p) \neq f(y,p)\}| \geq r$$

where

$$\binom{n}{2}(1-\frac{r}{n})^k < 1.$$

In particular, $B_G(n) \leq k$ provided (*) and $k > \frac{2n\ell nn}{r}$.

Proof: The proof depends on a probabilistic argument:

$$\text{Prob}[\underline{p}_1,\ldots,\underline{p}_k \text{ randomly chosen vertices make a beacon set}] > 0$$

if $\binom{n}{2}(1-\frac{r}{n})^k > 1$. If this is true, of course, it establishes the first part of the lemma. To begin, given x,y vertices of G,

$$\begin{aligned} &\text{Prob}[p_1,\ldots,p_k \ \underline{\text{not}} \text{ a beacon set}] \\ &= \text{Prob}[\exists x \neq y,\ f(x,\underline{p}_i) = f(y,\underline{p}_i),\ i = 1,\ldots,k] \\ &\leq \sum_{x \neq y} \text{Prob}[f(x,\underline{p}_i) = f(y,\underline{p}_i),\ i = 1,\ldots,k]. \end{aligned}$$

But by independence,

$$\begin{aligned} &\text{Prob}[f(x,\underline{p}_i) = f(y,\underline{p}_i),\ i = 1,\ldots,k] \\ &= \text{Prob}[f(x,\underline{p}) = f(y,\underline{p})]^k. \end{aligned}$$

Now,

$$\begin{aligned} &\text{Prob}[f(x,\underline{p}) = f(y,\underline{p})] \\ &= \frac{|\{p\colon f(x,p) = f(y,p)\}|}{n} \end{aligned}$$

and so is at most $\frac{n-r}{n}$. Thus,

$$\begin{aligned}&\text{Prob}[\underline{p}_1,\ldots,\underline{p}_k \text{ not a beacon set}]\\ &\le \binom{n}{2}\left(\frac{n-r}{n}\right)^k\\ &< 1,\end{aligned}$$

which proves the existence of a beacon set of size k.

The second part of the lemma follows easily from the inequality $(0 < r \le n)$.

$$\left(1-\frac{r}{n}\right)^{n/r} \le e^{-1}.$$

For then

$$\binom{n}{2}\left(1-\frac{r}{n}\right)^k \le \binom{n}{2}e^{-rk/n}$$

and for $k > \frac{2n\ell nn}{r}$ it follows that

$$\binom{n}{2}\left(1-\frac{r}{n}\right)^k < 1. \qquad \square$$

The importance of this lemma is that it reduces a proof of the existence of a small beacon set to a proof that each set

$$\{p\colon f(x,p) = f(y,p)\}$$

is small. With x,y distinct, let's call this set an *f-hyperplane* of the given graph -- or simply a hyperplane when there can be no confusion. Then

all hyperplanes small

=> beacon sets small

=> certificates can be computed fast.

The key to this result is the realization that in many classes of graphs all hyperplanes _are_ small.

4. *Almost All Graphs Have Small Beacon Sets*

Now the task is to prove that almost all graphs have small beacon sets. It is better to prove this, not with respect to distance, but rather with respect to f defined by

$$f(x,y) = (w_1(x,y), w_2(x,y)),$$

$w_1(x,y)$ coming into play if and only if x is adjacent to y. In other words, "distance" is now determined by the number of paths of length 1 and length 2 between x and y. Note that f(x,y) is invariant under isomorphism and can be computed in time $O(n^3)$ for all x,y [11].

The first result in this direction concerns testing a random graph $\underset{\sim}{G}$ against an arbitrary graph H for isomorphism.

Theorem 4: There is an algorithm A that satisfies the following:

(1) $A(G,H)$ accepts if and only if $G \simeq H$.

(2) $\text{Prob}[A(\underset{\sim}{G},H)$ runs in time at most $O(n^6)] \geq 1 - e^{-cn^{1/4}}$ where H is any graph with n vertices.

In other words, $I(n,\lambda) = O(n^6)$ where $\lambda = e^{-cn^{1/4}}$.

This result is significantly stronger than many similar statements. It says that if $\underset{\sim}{G}$ is a random graph on n vertices then $A(\underset{\sim}{G},H)$ can be computed in time $O(n^6)$ with probability → 1 (exponentially fast) -- no matter how H is selected. Our worst enemy can choose H isomorphic to $\underset{\sim}{G}$; he can make H only slightly different from $\underset{\sim}{G}$; he can even make H a function of $\underset{\sim}{G}$. No matter what he does, for almost all $\underset{\sim}{G}$ the algorithm A will run in polynomial time.

Now, the folklore has it that random graphs may not be worst-case for isomorphism testing [4,23]. But these methods are not limited to random graphs;

section 6 applies them to many other classes of graphs. While the bound of theorem 4 is polynomial, it is much too high to be practical. But in section 5 it is improved to $O(n^{3.5})$.

As might be expected, there is also a theorem on certificates.

Theorem 5: There is a function $F_w(G)$ that satisfies the following:

(1) $F_w(G)$, as a bit string, has length n^2.

(2) $F_w(G) = F_w(H)$ if and only if $G \simeq H$.

(3) The run time of $F_w(G)$ equals that of $F_w(H)$ provided $G \simeq H$.

(4) For almost all $\underset{\sim}{G}$ on n vertices, $F_w(\underset{\sim}{G})$ runs in time bounded by $O(n^6 \ell og n)$, and the probability that $\underset{\sim}{G}$ does not satisfy this is at most $e^{-cn^{\frac{1}{4}}}$.

In other words, $C(n,\lambda) = O(n^6 \ell og n)$ where $\lambda = e^{-cn^{\frac{1}{4}}}$.

Another way of saying this is that almost all graphs have succinct certificates. And, except for the $\ell og n$ factor, theorem 5 supplies an alternative proof of theorem 4. In order to determine whether $G \simeq H$, first compute $F_w(\underset{\sim}{G})$. Almost all the time, this can be done in $O(n^6 \ell og n)$, so assume that it can be. Then begin to compute $F_w(H)$. There are two cases. First, the computation halts in time $O(n^6 \ell og n)$; then $G \simeq H$ if and only if $F_w(\underset{\sim}{G}) = F_w(H)$. Or second, the computation does not halt in $O(n^6 \ell og n)$; then by part (3) of theorem 5, $\underset{\sim}{G}$ cannot be isomorphic to H. This demonstrates how to test whether $\underset{\sim}{G}$ is isomorphic to H and, as a bonus, it shows why this method is independent of the choice of H.

The key to these results is the following lemma on the size of hyperplanes in random graphs.

Lemma 6: Let $\underset{\sim}{G}$ be a random graph on n vertices. Then for $\alpha > \frac{1}{\sqrt{\pi}}$,

$$\text{Prob}[\exists x \neq y \ni \text{at least } \alpha n^{\frac{1}{2}} \text{ bad vertices for } x,y] \leq e^{-cn^{\frac{1}{4}}}$$

where a vertex p is *bad* if $f(x,p) = f(y,p)$. (As usual, $c, c_1, c_2, \ldots$ are constants.)

Proof: Let

$$q = \text{Prob}[\exists \text{ at least } \alpha n^{\frac{1}{2}} \text{ bad vertices for } x,y]$$

where x,y are distinct fixed vertices of $\underset{\sim}{G}$. The strategy for proving the lemma is first to show that $q = O(e^{-cn^{\frac{1}{4}}})$ and then to observe that

$$\text{Prob}[\exists x \neq y \ni \text{at least } \alpha n^{\frac{1}{2}} \text{ bad vertices for } x,y] \leq \binom{n}{2} q.$$

Fix x,y as distinct vertices in $\underset{\sim}{G}$. Let $\underset{\sim}{A}$, $\underset{\sim}{B}$, $\underset{\sim}{C}$, $\underset{\sim}{D}$ be the random variables defined by

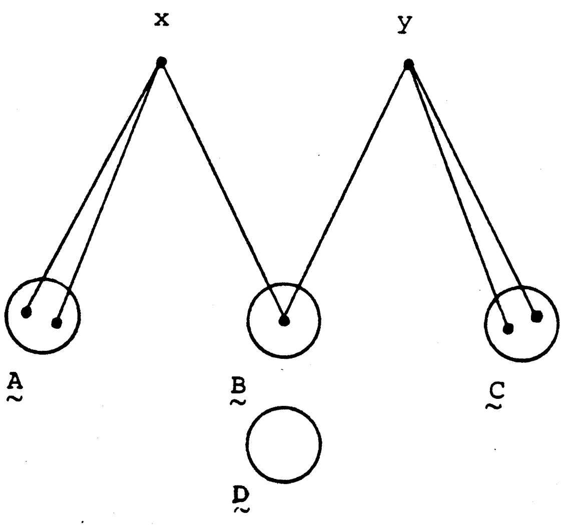
$$\underset{\sim}{A} = \{v\colon v \text{ adjacent to } x \text{ and not } y\}$$

$$\underset{\sim}{B} = \{v\colon v \text{ adjacent to } x \text{ and } y\}$$

$$\underset{\sim}{C} = \{v\colon v \text{ adjacent to } y \text{ and not } x\}$$

$$\underset{\sim}{D} = \{v\colon v \text{ adjacent to neither } x \text{ nor } y\}.$$

Clearly, $\underset{\sim}{A}$, $\underset{\sim}{B}$, $\underset{\sim}{C}$, $\underset{\sim}{D}$ form a partition of the vertices in G not including x and y:



And let $\underset{\sim}{Y}_p^{AC}$ be the random variable defined by

$$\underset{\sim}{Y}_p^{AC} = \begin{cases} 1 \text{ if p is adjacent to the same number of vertices in A as C} \\ 0 \text{ otherwise} \end{cases}$$

where p is a vertex and A and C are sets of vertices.

Next let $\Delta(A,B,C,D)$ mean "Each of the sets A,B,C,D has at least $\frac{n}{4} - n^{\frac{5}{8}}$ elements." Now we need to estimate

$$\text{Prob}[\sim\Delta(\underset{\sim}{A},\underset{\sim}{B},\underset{\sim}{C},\underset{\sim}{D})].$$

Intuitively one expects each of these sets to be about $\frac{n}{4}$ in size. To get the precise size, first observe that

$$\text{Prob}[\sim\Delta\underset{\sim}{A},\underset{\sim}{B},\underset{\sim}{C},\underset{\sim}{D}]$$

$$\leq \text{Prob}[|A| > \frac{n}{4} + \frac{1}{4}\, n^{\frac{5}{8}} \text{ or } \ldots \text{ or } \ldots \text{ or } |D| > \frac{n}{4} + \frac{1}{4}\, n^{\frac{5}{8}}]$$

since $|A| + |B| + |C| + |D| = n - 2$. Let

$$\underset{\sim}{X}_i = \begin{cases} 1 \text{ with probability } \frac{1}{4} \\ 0 \text{ with probability } \frac{3}{4} \end{cases}$$

be independent random variables. An application of the "central limit theorem" [10] shows that since $A = \underset{\sim}{X}_1 + \ldots + \underset{\sim}{X}_{n-2}$

$$\text{Prob}[|A| > \frac{n-2}{4} + \frac{\sqrt{3}}{4}\, x \cdot (n-2)^{\frac{1}{2}}] \qquad \frac{1}{\sqrt{2\pi}}\, e^{-\frac{1}{2}x^2}$$

as $n \to \infty$ and $x = \frac{2}{\sqrt{3}}\,(n-2)^{\frac{1}{8}}$. Thus

$$\text{Prob}[|\underset{\sim}{A}| > \frac{n}{4} + \frac{1}{4}\, n^{\frac{5}{8}}] \leq e^{-c_1 n^{\frac{1}{4}}}.$$

And then finally, the same argument applied to $\underset{\sim}{B}$, $\underset{\sim}{C}$, and $\underset{\sim}{D}$ yields as desired

$$\mathrm{Prob}[\sim\Delta(\underset{\sim}{A},\underset{\sim}{B},\underset{\sim}{C},\underset{\sim}{D})] \le 4e^{-c_1 n^{\frac{1}{4}}} \le e^{-c_1 n^{\frac{1}{4}}}.$$

Now we can return to the estimation of q. Clearly,

$$\begin{aligned} q \le\ & \mathrm{Prob}[\exists \text{ at least } \alpha n^{\frac{1}{2}} \text{ bad vertices for } x,y \wedge \Delta(\underset{\sim}{A},\underset{\sim}{B},\underset{\sim}{C},\underset{\sim}{D})] \\ & + \mathrm{Prob}[\exists \text{ at least } \alpha n^{\frac{1}{2}} \text{ bad vertices for } x,y \wedge \sim\Delta(\underset{\sim}{A},\underset{\sim}{B},\underset{\sim}{C},\underset{\sim}{D})] \end{aligned}$$

By our last estimate, it is sufficient to bound the first term on the right-hand side. Since no bad vertex p can lie in $\underset{\sim}{A}$ or $\underset{\sim}{C}$, it suffices to show that

$$q_1 = \mathrm{Prob}[\exists \text{ at least } \frac{\alpha n^{\frac{1}{2}}}{2} \text{ bad vertices for } x,y \text{ in } \underset{\sim}{B} \wedge \Delta]$$

and

$$q_2 = \mathrm{Prob}[\exists \text{ at least } \frac{\alpha n^{\frac{1}{2}}}{2} \text{ bad vertices for } x,y \text{ in } \underset{\sim}{D} \wedge \Delta]$$

have the required upper bounds where $\Delta = \Delta(\underset{\sim}{A},\underset{\sim}{B},\underset{\sim}{C},\underset{\sim}{D})$.

First we need to estimate q_1. Assume that $p \in \underset{\sim}{B}$ is bad for x,y. Then $w_2(x,p) = w_2(y,p)$. But

$$w_2(y,p) = \sum_{\substack{z \text{ adjacent} \\ y \text{ and } p}} 1 = \sum_{\substack{z \in \underset{\sim}{C} \\ z \text{ adjacent } p}} 1 + \sum_{\substack{z \in \underset{\sim}{B} \\ z \text{ adjacent } p}} 1 .$$

Thus,

$$\sum_{\substack{z \in \underset{\sim}{A} \\ z \text{ adjacent } p}} 1 = \sum_{\substack{z \in \underset{\sim}{C} \\ z \text{ adjacent } p}} 1$$

or, in other words, $\underset{\sim}{Y}_{p}^{AC} = 1$ where $A = \underset{\sim}{A}$ and $C = \underset{\sim}{C}$. Therefore,

$$q_1 \le \sum_{\substack{A,B,C,D \\ \text{with } \Delta(A,B,C,D) \text{ true}}} \mathrm{Prob}[\sum_{p \in B} \underset{\sim}{Y}_{p}^{AC} \ge \frac{\alpha n^{\frac{1}{2}}}{2} \wedge A = \underset{\sim}{A} \wedge \ldots \wedge D = \underset{\sim}{D} \wedge \Delta(A,B,C,D)]$$

where the sum is over all A,B,C,D partitions of the vertices of G minus x and y. Thus,

$$q_1 \le \sum_{\substack{A,B,C,D \\ \text{with } \Delta(A,B,C,D) \text{ true}}} \text{Prob}[\ \sum_{p \in B} \underset{\sim}{Y}_p^{AC} \ge \frac{\alpha n^{\frac{1}{2}}}{2} \wedge A = \underset{\sim}{A} \wedge \ldots \wedge D = \underset{\sim}{D}].$$

The key is that

$$\sum_{p \in B} \underset{\sim}{Y}_p^{AC} \ge \frac{\alpha n^{\frac{1}{2}}}{2}$$

and

$$A = \underset{\sim}{A} \ldots D = \underset{\sim}{D}$$

are independent events: The first uses only edges with <u>no</u> x,y endpoints, and the second uses only edges <u>with</u> an x or y endpoint. Hence

$$q_1 \le \sum_{\substack{A,B,C,D \\ \text{with } \Delta(A,B,C,D) \text{ true}}} \text{Prob}[\ \sum_{p \in B} \underset{\sim}{Y}_p^{AC} \ \frac{\alpha n^{\frac{1}{2}}}{2}] \cdot \text{Prob}[A = \underset{\sim}{A} \wedge \ldots \wedge D = \underset{\sim}{D}].$$

Therefore, it will follow that

$$q_1 \le \sum_{\substack{A,B,C,D \\ \text{with } \Delta(A,B,C,D) \text{ true}}} O(e^{-c_3 n^{\frac{1}{4}}}) \cdot \text{Prob}[A = \underset{\sim}{A} \wedge \ldots \wedge D = \underset{\sim}{D}] = O(e^{-c_3 n^{\frac{1}{4}}})$$

provided

$$\text{Prob}[\ \sum_{p \in B} \underset{\sim}{Y}_p^{AC} \ge \frac{\alpha n}{2}] = O(e^{-c_3 n^{\frac{1}{4}}})$$

where A,B,C,D satisfy Δ(A,B,C,D).

Let's therefore assume that Δ(A,B,C,D) is true. Now, $\underset{\sim}{Y}_p^{AC}$ for $p \in B$ are all independent random variables -- each one uses only edges with an endpoint p.

And

$$\mathrm{Prob}[\underset{\sim}{Y}_p^{AC} = 1] \;=\; \sum_i \frac{\binom{|A|}{i}\binom{|C|}{i}}{2^{|A|+|C|}} \;=\; \frac{\binom{|A|+|C|}{|C|}}{2^{|A|+|C|}} .$$

By Stirling's approximation and the fact that the central binomial term is the largest,

$$\mathrm{Prob}[\underset{\sim}{Y}_p^{AC} = 1] \;=\; \frac{\lambda}{n^{\frac{1}{2}}}$$

where

$$\lambda \le \frac{2}{\sqrt{\pi}}\,(1 + o(1)).$$

Intuitively we feel that $\sum_{p\epsilon B} \underset{\sim}{Y}_p^{AC}$ will exceed its expectation only rarely -- indeed the Chebyshev inequality demonstrates this. But in order to get tight bounds on the sum we cannot just apply the inequality. Nor can we use the "central limit theorem," since the distribution of the random variables depends on n. But we can use the inequality of Chernoff [9]. This implies that

$$\mathrm{Prob}[\sum_{p\epsilon B} \underset{\sim}{Y}_p^{AC} \ge \frac{n}{2}] \;\le\; \sum_{t\ge\frac{\alpha n^{\frac{1}{2}}}{2}} \binom{m}{t} p_0^t\, q_0^{m-t} \;\le\; e^{w_1+w_2}$$

where $p_0 = \frac{\lambda}{n^{\frac{1}{2}}}$, $q_0 = 1 - p_0$, $m = \frac{n}{4} + \Theta n^{\frac{5}{8}}$ $(0 \le \Theta \le 1)$, $k = \frac{\alpha n^{\frac{1}{2}}}{2}$,

$$w_1 \le \left(\frac{n}{4} + \Theta n^{\frac{5}{8}} - \frac{\alpha n^{\frac{1}{2}}}{2}\right) \ell n \left| \frac{\left(\frac{n}{4} + \Theta n^{\frac{5}{8}}\right)\left(\frac{1-\lambda}{n^{\frac{1}{2}}}\right)}{\frac{n}{4} + \Theta n^{\frac{5}{8}} - \frac{\alpha n^{\frac{1}{2}}}{2}} \right|$$

and

$$w_2 \le \frac{\alpha n^{\frac{1}{2}}}{2} \ln \left| \frac{\lambda}{n^{\frac{1}{2}}} \frac{(\frac{n}{4} + \Theta n^{\frac{5}{8}})}{\frac{\alpha n^{\frac{1}{2}}}{2}} \right|$$

provided also that $k \ge p_0 m$, m being the cardinality of B.

In order to simplify these expressions, we need to note that for any $\epsilon > 0$ small enough and $x \ge 0$ small enough

$$\ln \frac{1}{1-x} \le (i+\epsilon)x$$

and

$$\ln(1-x) \le -(1-\epsilon)x.$$

Then, after fixing $\epsilon > 0$ small and n large,

$$w_1 \le \left(\frac{n}{4} + \Theta n^{\frac{5}{8}} - \frac{\alpha n^{\frac{1}{2}}}{2}\right) \ln \left(1 - \frac{\lambda}{n^{\frac{1}{2}}}\right) + \left(\frac{n}{4} + \Theta n^{\frac{5}{8}} - \frac{\alpha n^{\frac{1}{2}}}{2}\right) \ln \frac{1}{1 - \frac{\frac{\alpha n^{\frac{1}{2}}}{2}}{\frac{n}{4} + \Theta n^{\frac{5}{8}}}}$$

and so

$$w_1 \le -\left(\frac{n}{4} + \Theta n^{\frac{5}{8}} - \frac{\alpha n^{\frac{1}{2}}}{2}\right)(1-\epsilon) \frac{\lambda}{n^{\frac{1}{2}}} + \left(\frac{n}{4} + \Theta n^{\frac{5}{8}} - \frac{\alpha n^{\frac{1}{2}}}{2}\right)(1+\epsilon) \frac{\frac{\alpha n^{\frac{1}{2}}}{2}}{\frac{n}{4} + \Theta n^{\frac{5}{8}}}.$$

Hence,

$$w_1 \le -\frac{\lambda}{4}(1-\epsilon)n^{\frac{1}{2}} + \frac{(1+\epsilon)\alpha}{2} n^{\frac{1}{2}} + o(n^{\frac{1}{2}}).$$

Also, turning our attention now to w_2,

$$w_2 \le \frac{\alpha n^{\frac{1}{2}}}{2} \ell n \frac{2\lambda(\frac{n}{4} + \Theta n^{\frac{5}{8}})}{\alpha n} \le \frac{\alpha n^{\frac{1}{2}}}{2} \ell n \left| \frac{\lambda}{2\alpha} (1 + O(n^{-\frac{3}{8}})) \right|$$

and so

$$w_2 \le - \frac{\alpha n^{\frac{1}{2}}}{2} \ell n \left| \frac{2\alpha}{\lambda} (1 + o(1)) \right| .$$

Finally,

$$w_1 + w_2 \le (- \frac{\lambda}{4} + \frac{\alpha}{2} - \frac{\alpha}{2} \ell n \frac{2\alpha}{\lambda}) n^{\frac{1}{2}} + \delta n^{\frac{1}{2}}$$

where $\delta > 0$ is arbitrarily small provided n is large enough and $\epsilon > 0$ is small enough. Of course, $k \ge p_0 m$ must be true, but it is true provided

$$\frac{\alpha n^{\frac{1}{2}}}{2} \ge \frac{\lambda}{n^{\frac{1}{2}}} (\frac{n}{4} + \Theta n^{\frac{5}{8}})$$

or $\alpha > \frac{\lambda}{2}$ and n is large enough. Since α is any quantity larger than $\frac{1}{\sqrt{\pi}}$ it follows that $\alpha > \frac{\lambda}{2}$ for large n. And an elementary argument shows that

$$- \frac{\lambda}{4} + \frac{\alpha}{2} - \frac{\alpha}{2} \ell n \frac{2\alpha}{\lambda} < -c_4$$

where $c_4 > 0$ is a constant. This concludes the estimation of q_1, since it is bounded by $e^{w_1+w_2}$ or $e^{-c_4 n^{\frac{1}{2}}}$ for n large.

Now we can complete the proof of the lemma by estimating q_2. If $p \in \underset{\sim}{P}$ is a bad vertex for x and y, then $w_2(x,p) = w_2(y,p)$ as before, and so

$$\sum_{\substack{z \text{ adjacent} \\ x \text{ and } p}} 1 = \sum_{\substack{z \text{ adjacent} \\ y \text{ and } p}} 1$$

or

$$\sum_{\substack{z \in \underset{\sim}{A} \\ z \text{ adjacent } p}} 1 = \sum_{\substack{z \in \underset{\sim}{C} \\ z \text{ adjacent } p}} 1$$

and so $\underset{\sim}{Y}_p^{AC} = 1$, where $A = \underset{\sim}{A}$ and $C = \underset{\sim}{C}$. The rest of the estimation can proceed as before, and this completes the proof of the lemma. □

Now we can complete the proof of theorems 4 and 5. The key to both of them is the following observation: Let $\underset{\sim}{G}$ be a random graph on n vertices. Then by lemma 6 ($\alpha = 1$) with probability exponentially close to 1 we can assume that for all distinct vertices x,y in $\underset{\sim}{G}$

$$|\{p: f(x,p) = f(y,p)\}| \leq n^{\frac{1}{2}}.$$

And then by lemma 3 it follows that $\underset{\sim}{G}$ has a beacon set of size 4. We can then follow the same strategy as in lemmas 1 and 2 to prove theorems 4 and 5; only theorem 5 need be done in some detail.

Proof of Theorem 5: $F_w(G)$ is exactly the same as $F_d(G)$ except of course that distance beacon sets are replaced by f-beacon sets.

1) ($F_w(G)$, as a bit string, has length n^2.) This follows immediately from the construction.

2) ($F_w(G) = F_w(H)$ if and only if $G \simeq H$.) This follows as in lemma 2. The key is that f(x,y) is clearly invariant under isomorphism.

3) (The run time of $F_w(G)$ equals that of $F_w(H)$ provided $G \simeq H$.) Suppose that $G \simeq H$ via the function ϕ. Then the determination of the beacon sets $B_1,\ldots,B_m$ of minimal size k is do-able in $O(kn^{k+2})$ in both. And the rest of the two computations, the sorting of the adjacency matrices, runs in the same bounds in both.

4) (For almost all $\underset{\sim}{G}$ on n vertices, $F_w(\underset{\sim}{G})$ runs in time bounded by $O(n^6 \ell og n)$, and the probability that $\underset{\sim}{G}$ does not satisfy this is at most $e^{-cn^{\frac{1}{2}}}$.) The last claim follows immediately from lemma 6 with $\alpha = 1$ and lemma 3 since they imply that $\underset{\sim}{G}$ has a beacon set of size 4 with the required probability. Note that f(x,y) can be precomputed by one matrix product, and so can be done in $O(n^3)$. □

Now you can see why it's so important to be careful in the estimations in lemma 6. Without that lemma we could conclude the existence of a basis only of size O(1), not 4. Some initial experiments on random graphs of various sizes indicate good agreement with the results of lemma 6:

size of n (number of vertices in $\underset{\sim}{G}$)	average size of hyperplane in $\underset{\sim}{G}$
50	2.8
100	3.9
150	4.7
200	5.6

For these values of n the average size of a hyperplane is roughly $.4n^{\frac{1}{2}}$. Lemma 6 predicts that as $n \to \infty$ it will approach $\alpha n^{\frac{1}{2}}$ where $\alpha > \frac{1}{\sqrt{\pi}} \approx .318$. Thus these experiments suggest that even random graphs of modest size will tend to have beacon sets of 4.

5. *An Improved Isomorphism Test*

This section shows how to improve the isomorphism test result of theorem 4 and, more generally, lemma 1. There are two important points to make about this improvement. First, it appears to yield only a faster isomorphism test, not a better certificate method. Second, it requires that we allow probabilistic algorithms [20]. That is, our algorithms must be permitted to make probabilistic choices. They have to be correct for any of these choices, but their running time may depend on the choices they make.

Theorem 7: There is a probabilistic algorithm A that satisfies the following:

(1) $A(G,H)$ accepts if and only if $G \approx H$.

(2) $A(G,H)$ runs in time $O(n^{3.5})$ with probability at least $1 - e^{-cn^{\frac{1}{2}}}$.

As before, H is <u>any</u> graph with n vertices, i.e., H can actually be a function of $\underset{\sim}{G}$. In other words, $I(n,\lambda) = O(n^{3.5})$ where $\lambda = e^{-cn^{\frac{1}{2}}}$. As with theorem 4, the exact statement of this theorem is noteworthy. With $\underset{\sim}{G}$ a random graph on n vertices, $A(\underset{\sim}{G},H)$ can be computed in time $O(n^{3.5})$ with probability 1 (exponentially) no matter how H is selected. Again, our worst enemy can choose H isomorphic to $\underset{\sim}{G}$ or not. No matter what he does, for almost all $\underset{\sim}{G}$ the algorithm is very likely to make probabilistic choices that allow it to run in the required time bound.

We can view statement (2) of theorem 7 alternatively as a sort of game: First $\underset{\sim}{G}$ is randomly selected. Then our enemy chooses H. Finally the algorithm A randomly makes certain choices. No matter what H our enemy chooses, we win -- that is, A runs in the required time bound -- almost all the time.

Proof: Let $\underset{\sim}{G}$ be a random graph on n vertices and let H be any graph on n vertices -- it may depend on $\underset{\sim}{G}$ in any way at all. Then, as in theorem 4, we can assume that

$$|\{p: f(x,p) = f(y,p)\}| \le n^{\frac{1}{2}}$$

for all distinct vertices x,y where $f(a,b) = f(w_1(a,b),w_2(a,b))$. And also as in theorem 4, we can precompute all the values of f(a,b) for both $\underset{\sim}{G}$ and H.

The key to the algorithm is the notion of a good beacon set. Let p_1,p_2 be vertices in $\underset{\sim}{G}$. Even though p_1 need not form a beacon set by itself, we can define the value of $f(x,p_1)$ as the name of x modulo p_1. In a similar way we can define the value of $(f(x,p_1),f(x,p_2))$ as the name of x modulo p_1,p_2. Then $p_1,\ldots,p_4$ is a *good* beacon set provided

1) $p_1,\ldots,p_4$ is a beacon set for $\underset{\sim}{G}$
2) under the equivalence relation "has the same name modulo p_1" p_2's equivalence class is smaller than $c_2n^{\frac{1}{2}}$
3) under the equivalence relation "has the same name modulo p_1,p_2" p_3's equivalence class is smaller than c_3
4) under the equivalence relation "has the same name modulo p_1,p_2" p_4's equivalence class is smaller than c_4.

Suppose for a moment that $p_1,\ldots,p_4$ is a good beacon set for $\underset{\sim}{G}$. Then in order to test whether $\underset{\sim}{G}$ is isomorphic to H in time $O(n^{3.5})$:

1) Try all q_1 in H and see whether

$$\{f(x,p_1): x \text{ in } G\} = \{f(y,q_1): y \text{ in } H\}$$

as multisets [17]. If q_1 does not satisfy this, then try the next q_1. If none are left, then $\underset{\sim}{G}$ is not isomorphic to H.

2) Now try all q_2 in H such that $f(p_2,p_1) = f(q_2,q_1)$ -- that is, q_2 has the same name modulo q_1 as p_2 does modulo p_1. For each q_2 then check to see whether the multisets of names modulo p_1,p_2 and q_1,q_2 are equal. If not, then try another q_2. If none are left, then as before $\underset{\sim}{G}$ is not isomorphic to H.

3) And now, try all q_3,q_2 in H such that they have the same names modulo q_1,q_2 as p_3,p_4 do modulo p_1,p_2. For each such pair, see whether $q_1,\ldots,q_4$ is a beacon set. If it is, check as in theorem 4 to see whether it induces an isomorphism. If it does, then $\underset{\sim}{G}$ is isomorphic to H. If all q_3,q_4 fail, then $\underset{\sim}{G}$ is not isomorphic to H.

Now, provided $p_1,\ldots,p_4$ is a good beacon set, this procedure will always be correct and will run in $O(n^{3.5})$. Clearly, if it outputs that $\underset{\sim}{G}$ is isomorphic to H, then it has actually constructed the isomorphism and so is correct. So let's assume that it has output that $\underset{\sim}{G}$ is not isomorphic to H. If this output comes in step (1) or (2), then there is no "counterpart" of certain vertices of $\underset{\sim}{G}$ in H, and so it is correct. If the output comes in step (3), then the procedure has acted just like the one in theorem 4 except that it pruned its search by avoiding impossible candidate lists $q_1,\ldots,q_4$. Therefore the procedure is always correct.

And does the procedure run in $O(n^{3.5})$? Let Q_i $(i = 1,\ldots,4)$ be the number of values assumed by q_i during this algorithm's execution. Then it runs in at most

$$(Q_1n^2 + Q_1Q_2n^2 + Q_1Q_2Q_3Q_4n^2) \quad = \quad O(Q_1Q_2Q_3Q_4n^2)$$

time. Clearly, $Q_1 = n$. Since $p_1,\ldots,p_4$ is a good beacon set, it follows that q_2 takes on at most $c_2n^{\frac{1}{2}}$ distinct values. To see this, note that q_2's equivalence class under the relation "has the same name modulo q " is smaller than $c_2n^{\frac{1}{2}}$, for the names assigned by p_1 and q_1 are the same. Thus $Q_2 \le c_2n^{\frac{1}{2}}$. The same argument shows that $Q_3 \le c_3$ and $Q_4 \le c_4$. Thus the procedure runs in $O(c_2c_3c_4n^{3.5})$.

To complete the proof of the theorem, it remains only to show how to find a good beacon set quickly. The key is that the argument of lemma 3 proves more than the existence of beacon sets; it proves that any randomly selected vertices $\underline{p}_1,\ldots,\underline{p}_4$ are likely to be a beacon set. We can use this to prove further that

$$\text{Prob}[p_1,\ldots,p_4 \text{ is a good beacon set}] \ge \delta > 0$$

where $\delta > 0$ is an absolute constant. Then the result follows by randomly selecting $p_1,\ldots,p_4$ until we get a good beacon set. Clearly the probability that this succeeds in at most m selections is at least $1 - \delta^m$. The theorem will then follow once we observe that each candidate beacon set can be checked in $O(n^2)$.

How likely is $\underline{p}_1,\ldots,\underline{p}_4$ to be a good beacon set? Let $\underline{Q}_2,\underline{Q}_3,\underline{Q}_4$ be the following random variables:

1) $\underline{Q}_2$ is the size of the number of elements in the same equivalence class as $\underline{p}_2$ and $\underline{p}_1$.

2) $\underline{Q}_3$ is the size of the number of elements in the same equivalence class as $\underline{p}_3$ modulo $\underline{p}_1,\underline{p}_2$.

3) $\underline{Q}_4$ is the size of the number of elements in the same equivalence class as $\underline{p}_4$ modulo $\underline{p}_1,\underline{p}_2$.

Then we must see whether $s \ge \delta$ where $\delta > 0$ is a constant and

$$s = \text{Prob}[\underline{Q}_2 \le c_2n^{\frac{1}{2}} \ \wedge\ \underline{Q}_3 \le c_3 \ \wedge\ \underline{Q}_4 \le c_4 \ \wedge\ \underline{p}_1,\ldots,\underline{p}_4 \text{ is a beacon set}].$$

If so,

$$s \;\geq\; 1 - \text{Prob}[\underset{\sim}{Q}_2 > c_2 n^{\frac{1}{2}} \;\wedge\; \underset{\sim}{Q}_3 > c_3 \;\wedge\; \underset{\sim}{Q}_4 > c_4 \;\wedge\; \underset{\sim}{p}_1,\ldots,\underset{\sim}{p}_4 \text{ not a beacon set} \;.$$

And it follows by the usual argument that

$$\begin{aligned} s \;\geq\; 1 &- \text{Prob}[\underset{\sim}{Q}_2 > c_2 n^{\frac{1}{2}}] \\ &- \text{Prob}[\underset{\sim}{Q}_3 > c_3] \\ &- \text{Prob}[\underset{\sim}{Q}_4 > c_4] \\ &- \text{Prob}[\underset{\sim}{p}_1,\ldots,\underset{\sim}{p}_4 \text{ not a beacon set}]. \end{aligned}$$

Since the hyperplanes of $\underset{\sim}{G}$ are less than or equal to $n^{\frac{1}{2}}$ in size, the argument of lemma 3 shows that

$$\text{Prob}[\underset{\sim}{p}_1,\ldots,\underset{\sim}{p}_4 \text{ is not a beacon set}] \;\leq\; \binom{n}{2}\left(\frac{1}{n^{\frac{1}{2}}}\right)^4 \;\leq\; .5.$$

Therefore, it is sufficient to show that $\text{Prob}[\underset{\sim}{Q}_2 > c_2 n^{\frac{1}{2}}]$, $\text{Prob}[\underset{\sim}{Q}_3 > c_3]$, and $\text{Prob}[\underset{\sim}{Q}_4 > c_4]$ are all small for c_2, c_3, and c_4 large enough.

Let's work first with $\text{Prob}[\underset{\sim}{Q}_2 > c_2 n^{\frac{1}{2}}]$. Examine the names modulo $\underset{\sim}{p}_1$. Let $\underset{\sim}{N}_i$ be a random variable $(i = 1,\ldots,n)$ that is the *ith* largest equivalence class with respect to the equivalence relation "have the same name modulo $\underset{\sim}{p}_1$." Clearly $\underset{\sim}{N}_1,\ldots,\underset{\sim}{N}_n$ partitions $[n]$. Let $\underset{\sim}{Y}$ be the number of unordered pairs of vertices with the same name, i.e.

$$\underset{\sim}{Y} \;=\; \sum_i \binom{|N_i|}{2}.$$

Then, as in lemma 3, where $E[X]$ is the <u>*expectation*</u> of X,

$$E[\underset{\sim}{Y}] \;\leq\; \sum_{\substack{x,y \\ \text{distinct}}} \text{Prob}[x,y \text{ have the same name modulo } p_1] \;\leq\; \binom{n}{2}\left(\frac{1}{n^{\frac{1}{2}}}\right) \;\leq\; \frac{n^{\frac{3}{2}}}{2}.$$

By the definition of expectation,

$$\mathrm{Prob}[\underset{\sim}{Y} \geq t] \;\leq\; \frac{E[\underset{\sim}{Y}]}{t} \;\leq\; \frac{n^{\frac{3}{2}}}{2t} \,.$$

Hence, $\mathrm{Prob}[\underset{\sim}{Y} \geq cn^{\frac{3}{2}}] \leq \frac{1}{2c}$. Now it follows that

$$\mathrm{Prob}[\underset{\sim}{Q}_2 > c_2 n^{\frac{1}{2}}] \;\leq\; \mathrm{Prob}[\underset{\sim}{Q}_2 > c_2 n^{\frac{1}{2}} \;\wedge\; \underset{\sim}{Y} < c_n^{\frac{3}{2}}] \;+\; \frac{1}{2c} \,.$$

Recall that $\underset{\sim}{Q}_2$ is the same as the number of elements in the equivalence class of $\underset{\sim}{p}_2$ and so

$$\begin{aligned}&\mathrm{Prob}[\underset{\sim}{Q}_2 > c_2 n^{\frac{1}{2}}]\\ \leq\;& \mathrm{Prob}[\underset{\sim}{p}_2 \text{ lies in } \underset{\sim}{N}_i \text{ with } |\underset{\sim}{N}_i| > c_2 n^{\frac{1}{2}} \;\wedge\; \underset{\sim}{Y} < cn^{\frac{3}{2}}] \;+\; \frac{1}{2c} \,.\end{aligned}$$

The first term on the right-hand side is bounded by

$$\sum_{\substack{N_1,\ldots,N_n \\ \text{partition } [n]}} \mathrm{Prob}[\Gamma_1 \wedge \Gamma_2]$$

where Γ_1 is the event "p_2 lies in N_i with $|N_i| > c_2 n^{\frac{1}{2}}$" and Γ_2 is the event "$N_1 = \underset{\sim}{N}_1, \;\ldots,\; \underset{\sim}{N}_n = \underset{\sim}{N}_n$ and $\underset{\sim}{Y} < cn^{\frac{3}{2}}$." Because these events are independent, it follows that $\mathrm{Prob}[\underset{\sim}{Q}_2 > c_2 n^{\frac{1}{2}}]$ is bounded by

$$\frac{1}{2c} \;+\; \sum_{\substack{N_1,\ldots,N_n \\ \text{partition } [n]}} \mathrm{Prob}[\Gamma_1]\cdot\mathrm{Prob}[\Gamma_2]$$

and hence also by

$$\frac{1}{2c} + \sum_{\substack{N_1,\dots,N_n \text{ partition } [n] \\ \binom{|N_1|}{2} + \dots + \binom{|N_n|}{2} < cn^{\frac{3}{2}}}} \text{Prob}[\Gamma_1]\cdot\text{Prob}[N_1 = \underset{\sim}{N}_1 \wedge \dots \wedge N_n = \underset{\sim}{N}_n].$$

Now let $N_1,\dots,N_n$ be a partition of $[n]$ with $\binom{|N_1|}{2} + \dots + \binom{|N_n|}{2} < cn^{\frac{3}{2}}$. Then $\text{Prob}[\Gamma_1]$ is at most

$$\frac{\sum\limits_{|N_i| > c_2 n^{\frac{1}{2}}} |N_i|}{n} \leq \frac{\sum\limits_{|N_i| > c_2 n^{\frac{1}{2}}} |N_i|^2}{c_2 n^{\frac{1}{2}} \cdot n}$$

$$\leq \frac{4 \sum\limits_i \binom{|N_i|}{2}}{c_2 n^{\frac{1}{2}} \cdot n}$$

$$\leq \frac{4c}{c_2}$$

Putting this all together yields

$$\text{Prob}[\underset{\sim}{Q}_2 > c_2 n^{\frac{1}{2}}] \leq \frac{1}{2c} + \sum_{\substack{N_1,\dots,N_n \\ \text{partition } [n]}} \frac{4c}{c_2} \text{Prob}[N_1 = \underset{\sim}{N}_1 \wedge \dots \wedge N_n = \underset{\sim}{N}_n]$$

and so $\text{Prob}[\underset{\sim}{Q}_2 > c_2 n^{\frac{1}{2}} \leq \frac{1}{2c} + \frac{4c}{c_2}$. Clearly, for c and c_2 large we can force this probability to be arbitrarily small. This completes the estimation of $\text{Prob}[\underset{\sim}{Q}_2 > c_2 n^{\frac{1}{2}}]$.

Now, to estimate $\text{Prob}[\underset{\sim}{Q}_3 > c_3]$, the argument is essentially the same as before. This time $\underset{\sim}{N}_i$ refers to the equivalence relation "have the same name modulo $\underset{\sim}{p}_1, \underset{\sim}{p}_2$." Then

$$E[\underset{\sim}{Y}] \;\le\; \binom{n}{2}\left(\frac{1}{n^{1/2}}\right)^2 \;\le\; \frac{n}{2}\,.$$

Therefore, as before,

$$\mathrm{Prob}[\underset{\sim}{Y} \ge cn] \;\le\; \frac{1}{2c}\,.$$

We then argue as before that

$$\mathrm{Prob}[\underset{\sim}{Q}_3 > c_3] \;\le\; \frac{1}{2c} \;+\; \mathrm{Prob}[\underset{\sim}{P}_3 \text{ lies in } \underset{\sim}{N}_i \text{ with } |\underset{\sim}{N}_i| > c_3 \;\wedge\; \underset{\sim}{Y} < cn].$$

Again this is bounded by

$$\frac{1}{2c} + \sum_{\substack{N_1,\dots,N_n \text{ partition } [n] \\ \binom{|N_1|}{2} + \dots + \binom{|N_n|}{2} \;<\; cn}} \mathrm{Prob}[\Gamma_1]\cdot\mathrm{Prob}[N_1 = \underset{\sim}{N}_1 \;\wedge\; \dots \;\wedge\; N_n = \underset{\sim}{N}_n].$$

where Γ_1 is the event "$\underset{\sim}{P}_3$ lies in N_i with $|N_i| > c_3$." It then follows that $\mathrm{Prob}[\Gamma_1]$ is bounded by

$$\frac{\sum_{|N_i|>c_3} |N_i|}{n} \;\le\; \frac{\sum_{|N_i|>c_3} |N_i|}{c_3 n}$$

$$\le\; \frac{4\sum_i \binom{|N_i|}{2}}{c_3 n}$$

$$\le\; \frac{4c}{c_3}\,.$$

Putting this all together yields $\mathrm{Prob}[\underset{\sim}{Q}_3 > c_3] \le \frac{1}{2c} + \frac{4c}{c_3}$, which as before yields the desired estimate provided c and c_3 are large.

Finally, the estimation of $\mathrm{Prob}[\underset{\sim}{Q}_4 > c_4]$ is exactly the same as the last one and can therefore be omitted.

This completes the proof of the theorem. □

The key to theorem 7 is the ability to prune our search by constantly checking G against H. This method can be generalized, so that the time bound of lemma 1 becomes $c^k k! n^{\frac{k}{2}+2}$ (c some constant) provided $\binom{n}{2}(1 - \frac{r}{n})^k < 1$ where $|\{p: d(x,p) \neq d(y,p)\}| \geq r$. Thus for k small the isomorphism test of lemma 1 is improved from $O(n^{k+2})$ to $O(n^{\frac{k}{2}+2})$.

6. *Further Directions*

This section will explore further applications of the beacon set methods to various classes of graphs, reducing these results to the graph-theory question of whether or not a small beacon set exists. Throughout this section we will work only with distance beacon sets. The key of course is that we must discover classes of graphs that have small hyperplanes -- that is, for x,y distinct vertices

$$|\{p: d(x,p) = d(y,p)\}|$$

is small. Obviously, many classes of graphs fail in this regard. For example, in any tree T if x and y are leaves and they are both adjacent to another vertex, then $|\{p: d(x,p) = d(y,p)\}| = n - 2$, where n is the number of vertices of T. (I note in passing, however, that any such x and y are the "same"; technically there is an automorphism from x to y [5]. Perhaps these methods can be extended to handle this situation.)

Theorem 8: $B_{\mathcal{G}}(n) \leq \frac{2n\ell nn}{r}$ provided for all $G \in \mathcal{G}$ any two distinct vertices x,y have at least r noncommon neighborhoods (z is a noncommon neighbor if it is adjacent to exactly one of x and y).

Proof: Immediate from lemma 3. □

Even omitting any reference to lemma 1, or better yet to section 5's improvement to lemma 1, it should be clear that the larger r is, the better the time bound on the isomorphism test. An interesting variant of this theorem is based on the concept of girth in a graph, *girth* being the size of the smallest circuit (if any) in G.

Corollary 9: $B_G(n) \le \frac{2n\ell nn}{2d-1}$ provided all $G \in \mathcal{G}$ have minimum degree at least d and girth $g \ge 5$.

Proof: This follows directly from theorem 8 and the simple observation that any distinct vertices x and y can have at most one vertex that they are both adjacent to; otherwise $g \le 4$. □

Thus if $g \ge 5$ and d is large, then G has a nontrivial improvement over the n! isomorphism test. It's also interesting to note that this girth condition can be replaced by a weaker one based on the number of cubes (cycles of length 4) in G.

The next set of results depends on a further refinement of our notion of a beacon set.

Definition: The vertices $p_1,\ldots,p_k$ are an *eulerian beacon set* for the graph G if for all distinct x and y with a common neighborhood (that is, with a vertex adjacent to both x and y) $d(x,p_i) \ne d(y,p_i)$ for some p_i. Let $E_{\mathcal{G}}(n)$ denote the maximum-size eulerian beacon set required for any n-vertex graph in $\mathcal{G}$.

Lemma 10:

1) $C_{\mathcal{G}}(n) \le k \cdot n^{k+2}$ where $k = E_{\mathcal{G}}(n)$.

2) If for all distinct x,y with a common neighbor in G, G any graph in $\mathcal{G}$, $|\{p: d(x,p) \ne d(y,p)\}| \ge r$, then $E_{\mathcal{G}}(n) \le k$ provided $\binom{n}{2}(1 - \frac{r}{n})^k < 1$.

Proof:

1) Let $p_1,\ldots,p_k$ be an eulerian beacon set in G and let $q_1,\ldots,q_k$ be an eulerian beacon set in H. First we need to see whether there is a way in $O(n^2)$ time to determine whether there is an isomorphism of G to H such that p_i maps to q_i $(i = 1,\ldots,k)$. If so, then as in lemma 1 we can obtain the required time bound.

Let G' be the multigraph [11] that corresponds to adding another copy of each edge of G and H' the same kind of multigraph for H. Then it is well known that G' and H' both have eulerian circuits [11]. Let $w_1,\ldots,w_m$ be any such eulerian circuit in G' with $p_1 = w_1$. We can obtain this circuit in time at most $O(n^2)$. Now we must construct the corresponding circuit in H' -- or at least try to construct it. Let $q_1 = w_1'$. Now assume that we have already constructed $w_1',\ldots,w_i'$, with $i < m$. Let w_{i+1}' be the neighbor of w_i' with the same name modulo $q_1,\ldots,q_k$ as w_i modulo $p_1,\ldots,p_k$. Since $p_1,\ldots,p_k$ and $q_1,\ldots,q_k$ are both eulerian beacon sets, it follows that w_{i+1}' must be unique if it exists. If it does not exist, then no isomorphism exists that maps p_i to q_i $(i = 1,\ldots,k)$ this follows directly from the definition of beacon sets. Therefore, assume that $w_1',\ldots,w_m'$ is successfully constructed. Define ϕ, the function from the vertices of G to those of H, as follows: $\phi(x) = y$ provided x is first visited on the circuit $w_1,\ldots,w_m$ at the same time that y is first visited on the circuit $w_1',\ldots,w_m'$. This is clearly well defined. Then we check whether ϕ is an isomorphism. And finally, if the desired isomorphism exists, then it must be ϕ.

2) This follows exactly as in lemma 3. □

As in section 5, we can improve this result by using a probabilistic algorithm and, as before, we can obtain certificates. Of course, the importance of the concept of an eulerian beacon set is that it places a potentially weaker restriction on the set of beacons.

Lemma 11: Let G be a graph with minimum degree d and girth g; then it has an eulerian beacon set of size k provided

$$\binom{n}{2}(1 - \frac{r}{n})^k < 1$$

where

$$r \geq \begin{cases} 2(d-1)^{\lfloor g/2 \rfloor - 1}, & g \text{ odd}, \\ (d-1)^{(g/2)-1} + (d-1)^{(g/2)-2}, & g \text{ even}. \end{cases}$$

Proof: Let x,y be distinct vertices both adjacent to another vertex z. Then it suffices to show that

$$|\{p\colon d(x,p) \neq d(y,p)\}| \geq r.$$

Let

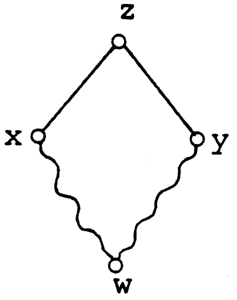
$$A = \{p\colon \text{there is a path from } x \text{ to } p \text{ of length} \leq h_1\}$$

and

$$B = \{p\colon \text{there is a path from } y \text{ to } p \text{ of length} \leq h_2\}$$

where h_1 and h_2 are defined later. Then provided $2h_1 < g$ and $2h_2 < g$, the induced graphs of A and B are trees. Let's prove this for A; the proof for B is similar.

The key is to look at all the paths out of x of length h_1. None of these paths can meet except at x; otherwise a circuit of length at most $2h_1 < g$ is formed. If A and B have no vertex in common that can be reached via paths that avoid z, with $h_1 + h_2 < g - 2$, the lemma is proved. So suppose that they do have such a common vertex, and call it w. Then x to w to y to z and back to x,



is a cycle provided we choose w to be the minimal distance from x. But then the cycle is of length at most g - 1, and this is a contradiction.

Finally, a calculation based on the parity of g and the fact that $r \geq (d-1)^{h_1} + (d-1)^{h_2}$ yields the proof of the lemma. □

Now recall that a graph G is a <u>*(d,g) cage*</u> if it has degree d everywhere and girth g and if it is the smallest such graph. Cages are interesting because of their uniformity. (Recall that cages need not be symmetric.)

<u>*Corollary 12:*</u> Let $\mathcal{G}$ be the class of (d,g) cages with d fixed. Then $c_{\mathcal{G}}(n) \leq 2^{cn^{\frac{1}{2}}\ell ogn}$.

<u>*Proof:*</u> Tutte [22] shows that any such graph G has n vertices where

$$n \leq \left(\frac{d-1}{d-2}\right) \cdot \left((d-1)^{g-1} + (d-2)^{g-2} + (d-4)\right)$$

and Biggs [4] shows that

$$n \geq \begin{cases} 1 + d + d(d-1) + \ldots + d(d-1)^{\frac{1}{2}(g-3)}, & g \text{ odd,} \\ 1 + d + d(d-1) + \ldots + d(d-1)^{\frac{1}{2}(g-1)}, & g \text{ even.} \end{cases}$$

At this point, the proof of the corollary is reduced to nothing more than a calculation and an application of lemma 11. □

It might be interesting to digress for a moment to look at a few examples of these results. In each case we can get a lower bound on r where

$$|\{p\colon d(x,p) \neq d(y,p)\}| \geq r$$

where x,y are distinct vertices with a common neighbor.

The first set of examples is based on graphs defined in Biggs [5]:

number of vertices of G	degree of G	girth of G	r
19	4	5	6
30	7	5	12
126	3	12	48
728	4	12	324

Of course, all these bounds on r are only lower bounds. But these estimates and the probabilistic isomorphism algorithm of section 5 suggest that it may be feasible to test even the largest of these graphs for isomorphism.

A second set of examples is based on the <u>*m-cubes*</u> Q_m: The vertices of these graphs are the 2^m vectors $v_1,\ldots,v_m)$ where $v_i = 0$ or 1 $(1 \leq i \leq m)$ and two vertices are adjacent when their vectors differ in exactly one coordinate. An easy argument demonstrates that r here is at least 2^{m-1}. This supplies an $n^{\ell ogn}$ isomorphism test $(n = 2^m)$ for m-cubes; again section 5 improves this to $O(n^{\frac{1}{2}\ell ogn + o(\ell ogn)})$.

Many other classes of graphs can be shown to satisfy the requirement that r be large. The rest of this section will explore some more general examples of such classes. But first I want to point out that one of the great merits of the beacon method is that it is predictable. That is, we can rapidly determine whether r is large by a simple sampling procedure that runs in $O(n^3)$. Thus we can estimate the running times of our isomorphism testing methods quite accurately without ever running them. This ability should be of some practical importance.

The motivation for corollary 12 and also for the next two results is an insight mentioned earlier that is due to Miller [19]. He suggests that graph isomorphism may be easier rather than harder in the case of symmetric graphs. Locally, parts of a symmetric graph look the same, so testing isomorphism for these graphs appears intuitively to be hard. Yet Miller's results suggest otherwise: He shows that isomorphism for these graphs is in both NP and the complement of NP.

The classes of symmetric graphs we study are 1-symmetric, 2-symmetric, and distance transitive [5]: Recall that an _automorphism_ of G is an isomorphism from G to G. G is _1-symmetric_ provided for each x,y adjacent and x',y' adjacent there is an automorphism ϕ such that $\phi(x) = x'$ and $\phi(y) = y'$. G is _2-symmetric_ if for each two paths x,y,z and x',y',z' both of length 2 there is an automorphism ϕ such that $\phi(x) = x'$, $\phi(y) = y'$, and $\phi(z) = z'$. And G is _distance transitive_ if for all x,y and x',y' with $d(x,y) = d(x',y')$ there is an automorphism ϕ such that $\phi(x) = x'$ and $\phi(y) = y'$.

Theorem 13: Let $\mathcal{G}$ be either the class of 2-symmetric graphs or the class of distance-transitive graphs of degree d, with d fixed. Then

$$B_{\mathcal{G}}(n) \le \frac{2n\ell nn}{r}$$

where $r \ge \frac{2}{d(d-1)} \cdot (n - d - 1)$.

Proof: Let x_0 and y_0 be two distinct vertices both adjacent to some vertex. We have to show that $|\{p: d(x_0,p) \neq d(y_0,p)\}| \geq r$, since this will then by lemma 10 imply our result. Define

$$\Gamma(x) = \{a: \exists \text{ shortest path from } y_0 \text{ to } a \text{ via } x\}.$$

Let $d(x_0,y_0) = \ell$; ℓ is clearly either 1 or 2. Now, for any x with $d(x,y_0) = \ell$, $|\Gamma(x_0)| \leq |\Gamma(x)|$. Let $\Gamma(x_0)$ contain the distinct vertices $a_1,\ldots,a_m$ and let $v_0^i,\ldots,v_{k_i}^i$ be the required path from y_0 to a_i via x_0. By the definition of Γ, $v_j^i = x_0$ for each i and some j. Clearly, since $d(x_0,y_0) = \ell$, $j \geq \ell$. But this path is a shortest path and so $j \leq \ell$; hence $v_\ell^i = x_0$ for all i. By the definitions of 2-symmetric and distance-transitive, there is an automorphism α such that $\alpha(y_0) = y_0$ and $\alpha(x_0) = x$. Then $\alpha(a_1),\ldots,\alpha(a_m)$ are all distinct, and the theorem will follow if we can show that they all lie in $\Gamma(x)$. Since α is an isomorphism, $\alpha(v_0^i),\ldots,\alpha(v_{k_i}^i)$ is a path from y_0 to $\alpha(a_{k_i}) = \alpha(a_i)$. And it is a shortest path since $d(y_0,a_i) = d(y_0,\alpha(a_i)) = k_i$. Finally, it goes via x since $\alpha(v_\ell^i) = \alpha(x_0) = x$. Therefore $\alpha(a_1),\ldots,\alpha(a_m)$ all lie in $\Gamma(x)$; hence our claim is correct.

The next point is that

$$|\Gamma(x_0)| \geq \frac{1}{d(d-1)} \cdot (n - d - 1).$$

To see this, let z be any vertex with $d(y_0,z) \geq \ell$. Then z must be in

$$\bigcup_{d(y_0,x)=\ell} \Gamma(x).$$

For if $d(y_0,z) \geq \ell$ then there is a shortest path, say, $w_1,\ldots,w_m$, from y_0 to z. Then by definition z lies in $\Gamma(w_\ell)$ and $d(y_0,w_\ell) = \ell$. Thus

$$|\{z\colon d(y_0,z) \geq \ell\}| \;\leq\; |\Gamma(x_0)| \cdot |\{x\colon d(y_0,x) = \ell\}|,$$

since the cardinality of all the $\Gamma(x)$ with $d(y_0,x) = \ell$ are the same. Clearly $|\{x\colon d(y_0,x) = \ell\}| \leq d(d-1)$ since $\ell \leq 2$. Also $|\{z\colon d(y_0,z) \geq \ell\}| \geq n - d - 1$. So the second claim is proved.

The key to the proof now is the observation that any element of $\Gamma(x_0)$ is closer to x_0 than to y_0. The same argument with the roles of x_0 and y_0 reversed then completes the proof. □

Corollary 14: Let G be the class of graphs with degree d that are 2-symmetric or distance-transitive (d fixed). Then

$$C_G(n) \leq n^{c\ell ogn}.$$

Theorem 15: Let G be the class of cubic symmetric graphs. Then

$$B_G(n) \;\leq\; 12\ell nn + o(1).$$

Proof: Let a, b, and c be the three neighbors of a vertex x. Then since G is cubic and 1-symmetric, there is an automorphism, say α, that leaves x fixed and leaves none of a,b,c fixed. Otherwise the stabilizer of x, the set of automorphisms that fix x, would be of order 2, which is impossible.

Let $H(u,v) = |\{p\colon d(u,p) \neq d(v,p)\}|$. Then in order to prove this theorem we must show that for each x with neighbors a,b,c

$$H(a,b) \;\geq\; \frac{1}{6}(n-4),$$

$$H(b,c) \;\geq\; \frac{1}{6}(n-4),$$

and

$$H(a,c) \geq \frac{1}{6}(n-4).$$

Actually, in order to establish this it is sufficient to prove only one of these inequalities. Assume that $H(a,b) \geq \frac{1}{6}(n-4)$. Then let α be the automorphism that fixes x and leaves none of a,b,c fixed. Without loss of generality, we can assume that $\alpha(a) = c$, since it's clear that either $\alpha(a)$ or $\alpha(b)$ must be c. Now $\alpha(b)$ must equal a, and so $\alpha(c)$ must equal b:

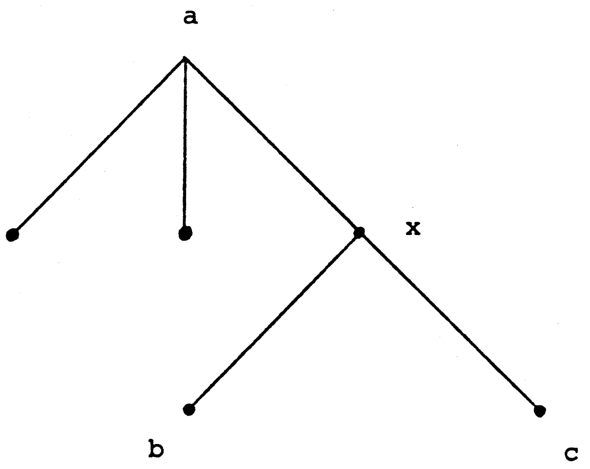
$$\alpha(a) = c, \ \alpha(b) = a, \text{ and } \alpha(c) = b.$$

Thus $H(a,b) \geq \frac{1}{6}(n-4)$, and so $H(\alpha(a),\alpha(b)) = H(c,a) \geq \frac{1}{6}(n-4)$ and, in the same way, $H(\alpha(c),\alpha(a)) = H(b,c) \geq \frac{1}{6}(n-4)$. This clearly follows since H(u,v) is invariant under isomorphism.

Thus it remains only to prove that one of these inequalities does hold. First we note that we can assume that $G \neq K_4$ and therefore that a, b, and c are not adjacent to each other. Otherwise we can use the automorphism α to get that G is K_4. As in theorem 13, define

$$\Gamma(w) = \{y\colon \exists \text{ shortest path from a to y via w}\}.$$

Then, again as in theorem 13, we can show that $]\Gamma(x)] \geq \frac{1}{3}(n-1)$:



Note that this uses only the fact that G is 1-symmetric. Then

$$|\Gamma(b)| + |\Gamma(c)| \geq |\Gamma(x)| - 1.$$

This follows because if $v \in \Gamma(x)$ and $v \neq x$ then $v \in \Gamma(b) \cup \Gamma(c)$. Then either

$$|\Gamma(b)| \geq \frac{1}{2}(|\Gamma(x)| - 1)$$

or

$$|\Gamma(c)| \geq \frac{1}{2}(|\Gamma(x)| - 1).$$

Since $\frac{1}{2}(|\Gamma(x)| - 1) = \frac{1}{6}(n-4)$, we can assume without loss of generality that

$$|\Gamma(b)| \geq \frac{1}{6}(n-4).$$

This completes the proof of the theorem; for $H(a,b) \geq |\Gamma(b)|$. □

Corollary 16: Let G be the class of cubic symmetric graphs. Then

$$C_G(n) \leq n^{c\ell ogn}.$$

These results support Miller's insight that isomorphism for symmetric graphs is perhaps easier than the general case.

The final result concerns the isomorphism of points in general position in euclidean space E^d of dimension d. As usual, we use $d(x,y)$ to denote the distance between the points x and y. A set of n points in E^d are in *general position* if no d+1 of them lie on any common hyperplane [8]. We say that they are *weakly isomorphic* is there is a permutation of the indices ϕ such that $d(x_i,x_j) = d(y_{\phi(i)},y_{\phi(j)})$ for all i and j.

Theorem 17: Weak isomorphism of sets of n points in general position in E^d can be done in $O(n^{k+2})$ provided $n \geq d^\sigma$ where $k \geq \frac{2\sigma}{\sigma-1}$.

Proof: Let $x_1,\ldots,x_n$ and $y_1,\ldots,y_n$ be two sets of points. Weak isomorphism is essentially equivalent to isomorphism of the underlying complete graphs with the weight d(a,b) attached to the edge {a,b}. The methods of lemma 1 and lemma 3 can be generalized to this setting to provide an $O(n^{k+2})$ algorithm for isomorphism testing if there is a k-beacon set. Now

$$|\{x_\ell: d(x_i,x_\ell) = d(x_j,x_\ell)\}| \leq d$$

since the set of points x_ℓ equidistant from x_i and x_j $(i \neq j)$ is a hyperplane. The result then follows as in lemma 3 since

$$\binom{n}{2} (\frac{d}{n})^k < 1$$

provided $n \geq d^\sigma$ and $k \geq \frac{2\sigma}{\sigma-1}$. □

As in section 5, this result can be improved to an algorithm that runs in $O(n^{3.34})$ when n is larger than, say, d^3. And the time bound of this theorem is not n^d but a fixed polynomial in n provided only that n is larger than some nontrivial power of d.

Acknowledgements

I wish to thank John M. Lewis and Richard Ladner for a number of very helpful discussions about graph isomorphism, Mary-Claire van Leunen for her help in editing, and Rich DeMillo for his ensights and encouragement while I worked on these results.

References

1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1976.

2] L. Babi and P. Erdős. Random graph isomorphism. Unpublished manuscript, 1978.

3] H. G. Barrow, A. P. Ambler, and R. M. Burstall. Some techniques for recognising structures in pictures. In S. Watanabe, editor, *Frontiers of Pattern Recognition*, pages 1-29. Academic Press, 1972.

4] A. T. Berztiss. A backtracking procedure for isomorphism of directed graphs. *JACM* 20(3):365-377, July 1973.

5] N. Biggs. *Algebraic Graph Theory*, pages 154, 163. Cambridge University Press, 1974.

6] D. G. Corneil and C. C. Gotlieb. An efficient algorithm for graph isomorphism. *JACM* 17(1):51-64, January 1970.

7] D. G. Corneil and R. Mathon. Algorithm techniques for the generation and analysis of strongly regular graphs. To appear.

8] C. W. Dodge. *Euclidean Geometry and Transformations*. Addison-Wesley, 1977.

9] P. Erdős and J. Spencer. *Probabilistic Methods in Combinatorics*, page 18. Academic Press, 1974.

10] W. Feller. *An Introduction to Probability Theory and Its Applications*, Volume I, page 178. John Wiley & Sons, second edition 1957.

11] F. Harary. *Graph Theory*. Addison-Wesley, 1969.

12] F. Harary and J. Meter. On the metric dimension of a graph. *Ars Combinatoria*, Volume 2, pages 191-195, 1976.

13] J. E. Hopcroft and R. E. Tarjan. A $v\ell ogv$ algorithm for the isomorphism of triconnected planar graphs. *JCSS* 7(3):323-331, 1973.

14] J. E. Hopcroft and J. K. Wong. Linear time algorithm for isomorphism of planar graphs. *Proceedings of the 6th Annual ACM Symposium on Theory of Computing*, pages 172-184, 1974.

15] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85-103. Plenum Press, 1972.

16] R. M. Karp. Private communication.

17] D. E. Knuth. *Fundamental Algorithms*, Volume I. Addison-Wesley, 1968.

18] R. J. Lipton and R. E. Tarjan. A separator theorem for graphs of arbitrary genus. In preparation.

19] G. Miller. Graph isomorphism: General remarks. *Proceedings of the Ninth Annual ACM Symposium on Theory of Computing*, pages 143-150, 1977.

20] M. O. Rabin. Probabilistic algorithms. To appear.

21] R. E. Tarjan. Private communication.

22] W. T. Tutte. *Connectivity in Graphs*. University of Toronto Press, 1966.

23] J. P. Ullman. An algorithm for subgraph isomorphism. *JACM* 23(1):31-42, January 1976.