# Finding a maximum independent set in time $O(2^{n/4})$?

J. M. Robson

January 2001

**Abstract**

We improve previous algorithms for finding the size of the largest independent set of a graph in three ways: firstly a more detailed analysis of the possible local structures around a vertex where the algorithm will branch recursively, much of the most detailed analysis being done by computer, secondly more use of auxiliary functions which consider independent sets subject to certain extra conditions and thirdly a better analysis of the improvements achieved by a dynamic programming variant. These improvements reduce the constant $c$ in the $O(2^{cn})$ time bound to less than 0.251.

## 1 Introduction

We consider the NP-hard problem of finding a maximum independent (or stable) set in a graph. As in previous papers ([4], [3]) we concentrate on determining the size of such a set, that is the *stability number* of the graph. Actually finding a set of this size cannot be much harder than finding the size and, in fact, our algorithms for the size can easily be modified to give algorithms of the same complexity for finding one set.

Previous papers have given algorithms with proven bounds on the run time for $n$ vertex graphs of $O(\alpha^n)$ with $\alpha < 2$) often expressed as $O(2^{cn})$, first for $c < 1/3$ ([4] and then $c \approx 0.304$ ([2]) and $c \approx 0.276$ ([3]. In this paper we reduce this bound to just under $c = 0.251$ ($\alpha \approx 1.189943$). [1] obtains a better bound for the case of graphs of degree bounded by 3.

### 1.1 Notation

$G$ is a graph $(V, E)$. We write $G - v$ or $G - S$ for the subgraph induced on $V - v$ or $V - S$ for $v \in V$ or $S \subset V$.
We write $N(v)$ for the set of neighbours of a vertex $v$ and $\overline{N}(v)$ for $N(v) + v$. $N^2(v) = N(N(v)) - v$ and $\overline{N}^2(v) = \overline{N}(\overline{N}(v))$. These notations are also used with a set of vertices rather than the single vertex $v$.

We write $d_G(v)$ for the degree of $v$ in $G$ and omit the $G$ when no ambiguity can arise.
We often abbreviate maximal independent set as m.i.s.

## 1.2 Structure

### 1.2.1 Degree less than 9 plus memorisation

We start by describing an algorithm *stab* computing the stability number of graphs of degree at most 9 and whose run time is $O(\alpha^n)$ for $\alpha \approx 1.204879$, that is $2^{cn}$ for $c \approx 0.268888$ on graphs of size $n$. Then we apply the technique of *memorisation* used in [3] to deduce an algorithm running in time $O({\alpha'}^n)$ for $\alpha' \approx 1.189980$ ($O(2^{c'n})$ for $c' \approx 0.250937$), still on graphs of degree at most 9. Finally we give an algorithm achieving the same bound $\alpha'$ on general graphs which has a simple structure as long as the degree is more than 9 and calls the second algorithm whenever it is applied to a subgraph of degree $\leq 9$.

### 1.2.2 Assumption that best case is chosen

Except for trivial cases, the algorithm considers the neighbourhood of a chosen vertex $v$, decides that there must be a maximum independent set obtained in one of $m$ ways by including certain vertices $IN_i$ in the independent set and excluding certain others $EX_i$, finds the size of the max obtained in each way $|IN_i| + stab(G - IN_i - EX_i)$ and takes the maximum of these as $i$ ranges from 1 to $m$. There are a large number of cases depending on the structure found in the neighbourhood of $v$. For each case, we claim that the time taken is bounded by $c_{case}2^{cn}$ for a constant $c_{case} \leq 1$ and prove this recursively by considering the cases which might arise in the recursive calls. The algorithm always chooses the case with the smallest value of $c_{case}$, that is in considering any case we can assume that no "better" cases are present in the graph although it is convenient to present the cases in an order different from that of the constants.

### 1.2.3 Constrained auxiliary functions

The detailed structure of cases distinguished by the algorithm is somewhat simplified by isolating themes which can be exploited in a number of places.

One situation which arises frequently is illustrated by the simple case of a vertex $v$ of degree 3 and having neighbours $w$, $x$ and $y$. If any independent set contained none of these four vertices or contained only $w$ or $x$ or $y$, it could be modified to give an independent set at least as large by adding $v$ and dropping $w$ or $x$ or $y$ if they were present. Hence we can say that there is a mis either containing $v$ (and hence not $w$, $x$ or $y$) or not containing $v$ but containing at least two of $\{w, x, y\}$. Hence in the recursive call on $G - v$ we are not concerned by the result unless every mis of $G - v$ contains at least two of $w$, $x$ and $y$. If this condition is satisfied our function must return the true stability number; otherwise we require only that the value returned is no greater than the true stability number. Hence we define a first auxiliary function $stab'$ which has as parameters a graph $G$ and a number $i \geq 1$ of pairs each consisting of a subset $S_i$ of the vertices of $G$ and an integer $n_i$; if every maximal independent set of $G$ contains at least $n_i$ vertices from $S_i$ for very $i$, the function returns the stability number of $G$; otherwise it returns a value less than or equal to the stability number.

Next we define a second auxiliary function $stab''$ for the even more constrained case where we are interested only in independent sets containing *exactly* one of two specified vertices. Precisely: $stab''(G, x, y)$ is only defined where $x$ and $y$ are two vertices of $G$ not connected by an edge; if every maximal independent set of $G$ contains either $x$ or $y$ but not both, the function returns the stability number of $G$; otherwise it returns a value less than or equal to the stability number.

Finally we remark that the gain from the memorisation technique of Section 4 depends on the fact that all graphs considered are induced subgraphs of the original graph but occasionally it is useful to consider a graph obtained by adding edges between one vertex and a number of others. Accordingly we define $stab'''(G, x)$ which returns the stability number of $G$ but does so in such a way that every call of $stab$, $stab'$ or $stab''$ has as parameter an induced subgraph of $G$ which does not contain the vertex $x$.

For each case of each of these auxiliary functions we define a constant $a_{case}$ and claim that the computation time is $a_{case}2^{cn}$. All the constants $a_{case} \leq 1$ except for those associated with the function $stab'''$.

We give equations for each of these constants. The form of these equations is that of C assignment statements because our latex file can be directly incorporated into a C program which evaluates the constants and finds the smallest value of $c$ for which the argument is valid. In this C program, $alpha$ is a pointer into a table containing powers of $\alpha$ and $max$ is a function such that $max(n, x_1, \cdots, x_n)$ is the maximum of $x_1, \cdots, x_n$ (for the values of $n$ used). The constants for the function $stab$ all have names beginning $c$ and those for the auxiliary functions begin $a$. The rest of the names are obtained in an obvious way from the labels of the enumerations in Sections 2 and 3.

Some of the printed values of the constants are followed by $***$. This is an indicator that they are *critical*, that is that reducing their value would allow an immediate decrease in the value of $\alpha$. We are far from having exhausted the possibilities of more detailed case analysis further improving the algorithm's performance.

#### 1.2.4 Basic idea of min degree and max neighbour degree and not regular

The normal action of the algorithm is to choose a pair $A$ and $B$ of neighbouring vertices so as to minimise $d(A)$, ensure that $d(B) > d(A)$ and lastly maximise the number of neighbours of $A$ with the same degree $d(A)$. For reasons discussed in [3], the analysis can ignore the case of disconnected and regular graphs, so we can always find such a pair.

## 2 Unconstrained

1. Dominance: if there are two vertices $A$ and $B$ such that $\overline{N}(A) \subset \overline{N}(B)$, then there is a maximum independent set not containing $B$. Calculate $stab(G) = stab(G - B)$.

   (a) $c1a = alpha[-1];= 0.829959$.

   (b) Note that a vertex of degree 1 dominates its neighbour and that, in this case the call $stab(G - B)$ has $A$ an isolated vertex giving $c1b = c9 * alpha[-1];= = 0.688832$.

   $c1 = c1a;= 0.829959$.

2. Degree 2: if a vertex $A$ has degree 2, consider its neighbours, say $B$ and $B'$. There is a m.i.s. containing $A$ (and neither $B$ nor $B'$) or both $B$ and $B'$. We write $x$ for $|\overline{N}^2(A) - \overline{N}(A)|$ and $y$ for $|\overline{N}^3(A) - \overline{N}^2(A)|$

   (a) if $B$ and $B'$ are connected, no independent set can contain both and we have $stab(G) = 1 + stab(G - \overline{N}(A)$
   $c2a = alpha[-3];= 0.571702$.
   Otherwise the independent set containing $A$ can be assumed to contain two elements of $|\overline{N}(\overline{N}(A)) - \overline{N}(A)|$ since, if not, it cannot be larger than all independent sets containing $B$ and $B'$. this gives us $stab(G) = max(1+stab'(G-\overline{N}(A), (\overline{N}(\overline{N}(A))-\overline{N}(A), 2)), 2+stab(G-\overline{N}(\overline{N}(A)))$.

   (b) if $x \leq 1$, the $stab'$ call is vacuous and we have
   $c2b = alpha[-2];= 0.688832$.

   (c) if $x = 2$, the $stab'$ call can be replaced by $2+stab(G-\overline{N}(\overline{N}(\overline{N}(A))))$,

      i. **if** $y = 1$
      the $stab$ call can be dropped giving
      $c2ci = alpha[-6];= 0.326843$.

      ii. **else**
      $c2cii = alpha[-7] + alpha[-5];= 0.665073$.

   $c2c = max(2, c2ci, c2cii);= 0.665073$.

   (d) if $x = 3$
   $c2d = a3 * alpha[-3] + alpha[-6];= 0.720650$

   (e) if $x = 4$
   $c2e = a4 * alpha[-3] + alpha[-7];= 0.714742$

   (f) if $x = 5$
   $c2f = a5 * alpha[-3] + alpha[-8];= 0.738691$

   (g) if $x = 6$
   $c2g = a6 * alpha[-3] + alpha[-9];= 0.728704$

   (h) if $x > 6$
   $c2h = alpha[-3] + alpha[-10];= 0.726786$

Hence $c2 = max(7, c2a, c2b, c2c, c2d, c2e, c2f, c2g);= 0.738691$.

3. **Degree 3**
Similarly to the previous case, if $d(A) = 3$, then there is a m.i.s. containing either $A$ or at least two neighbours of $A$. By the choice of $A$ and $B$, any neighbour of $A$ with degree 3 has a neighbour with degree no more than $d(B)$.

   (a) **In $N(A)$ (for some vertex $A$ with $d(A) = 3$) a vertex $v$ is connected to the other two vertices**
   $A$ dominates $v$ so $v$ can be deleted leaving $d(A) = 2$.
   $c3a = c2 * alpha[-1];= 0.613083$.

   (b) **All vertices of $N(A)$ have degree at least 4**
   $stab(G) = max(1 + stab(G - \overline{N}(A)), stab'(G - A, (N(A), 2)))$, where in the second call all vertices of $G - A$ have degree at least 3
   $c3b = alpha[-4] + a3i * alpha[-1];= 0.837212$ *** .
   Or if there are two non-adjacent vertices of degree 3
   $c3bx = c3 * alpha[-4] + a3i * alpha[-1];= 0.759971$.

   (c) **Some vertex $A$ of degree 3 has neighbours $v, w, x$ with $d(v) > 3$ and $(w, x)$ an edge**
   $stab(G) = max(1 + stab(G - \overline{N}(A)), stab'(G - \overline{N}(v), (\{w, x\}, 1)))$
   $c3c = alpha[-4] + a1 * alpha[-5];= 0.801333$.

   (d) **$A$ has neighbours $v, w, x$ with $d(v) = d(w) = 3$**
   We must have $d(x) > 3$ by the choice of $A$. $stab(G) = max(1 + stab(G - \overline{N}(v)), stab'(G - v), (N(v), 2)))$ where, in the first call $d(w) \leq 2$ and in the second all vertices have degree $\geq 2$
   $c3d = max(3, c9, c1b, c2) * alpha[-4] + max(3, a3e, a3f, a3h) * alpha[-1];= 0.813306$.
   The situation now is symmetric and we change our notation: $A$ has a neighbour also of degree 3 and we call these two vertices $A_1$ and $A_2$; the two other neighbours of $A_i$ we call $B_{i,j}$ $(j = 1, 2)$ and the four vertices $B_{i,j}$ are distinct.

   (e) **At most one edge $(B_{j,1}, B_{j,2})$**
   W.l.o.g. we suppose that $(B_{2,1}, B_{2,2})$ is not an edge.

      i. **Some $B_{i,j}$ has degree $\leq 4$**
      $stab(G) = max(1 + stab(G - \overline{N}(A_{i'})), stab'(G - A_{i'}), (N(A_{i'}), 2)))$
      $c3ei = max(4, c9, c1b, c2, c3) * alpha[-4] + max(3, a3e, a3f, a3h) * alpha[-1];= 0.816748$.

      ii. **Some $B_{i,j}$ has degree $\geq 7$**
      $stab(G) = max(1 + stab(G - \overline{N}(B_{i,j})), stab(G - B_{i,j}))$
      $c3eii = max(3, c9, c1b, c2) * alpha[-8] + c2 * alpha[-1];= 0.799940$.
      In the remaining cases we write $e$ for the number of edges $(B_{1,1}, B_{2,j})$ assuming w.l.o.g. that this quantity is at least as great for $B_{1,1}$ as for $B_{1,2}$.

iii. $d(B_{1,1}) = 5$

A. $e = 0$
$stab(G) = max(1+stab(G-\overline{N}(A_2)), stab'(G-A_2, (\{A_1, B_{2,1}, B_{2,2}\}, 2)))$ where $d(B_{1,1}) < 5$ in $G - \overline{N}(A_2)$ and, in $G - A_2$, $d(A_1) = 2$ and $A_1$ has no common neighbours with $B_{2,1}$ or $B_{2,2}$.
$c3eiiiA = max(5, c9, c1b, c2, c3, c4)*alpha[-4]+a3hiy*alpha[-1];=$ 0.761906.

B. $e > 0$
$stab(G) = max(1+stab(G-\overline{N}(A_2)), stab'(G-A_2, (\{A_1, B_{2,1}, B_{2,2}\}, 2)))$. where $d(B_{1,1}) \leq 3$ in $G - \overline{N}(A_2)$
$c3eiiiB = max(4, c9, c1b, c2, c3)*alpha[-4]+max(2, a3f, a3h)*$ $alpha[-1];=$ 0.816748.

$c3eiii = max(2, c3eiiiA, c3eiiiB);=$ 0.816748.

iv. $d(B_{1,1}) = 6$

A. $e \leq 1$
$stab(G) = max(1+stab(G-\overline{N}(B_{1,1})), stab(G-B_{1,1}))$ where $1 \leq d(A_2) \leq 2$ in $G - \overline{N}(B_{1,1})$ and $d(A_1) = 2$ in $G - B_{1,1}$.
$c3eivA = max(2, c1b, c2)*alpha[-7]+c2*alpha[-1];=$ 0.813465.

B. $e = 2$
$stab(G) = max(1+stab(G-\overline{N}(A_2)), stab'(G-A_2, (\{A_1, B_{2,1}, B_{2,2}\}, 2)))$ where $d(B_{1,1}) = 3$ in $G - \overline{N}(A_2)$.
$c3eivB = c3 * alpha[-4] + max(2, a3f, a3h) * alpha[-1];=$ 0.816748.

$c3eiv = max(2, c3eivA, c3eivB);=$ 0.816748.

$c3e = max(4, c3ei, c3eii, c3eii, c3eiv);=$ 0.816748 *** .

(f) **Two edges** $(B_{1,1}, B_{1,2})$ **and** $(B_{2,1}, B_{2,2})$ For each $B_{i,j}$ we define $d'(B_{i,j})$ as the degree of $B_{i,j}$ in the graph $G - \overline{N}(A_{i'})$ $(\prime \neq i)$ and then we define $m$ as the minimum of the four $d'(B_{i,j})$. In the next three cases $i$ and $j$ refer to (one of) the $B_{i,j}$ giving this minimum.

i. $m \leq 3$
$stab(G) = max(1+stab(G-\overline{N}(A_{i'})), 1+stab'(G-A_{i'}, (N(A_{i'}), 2)))$
$c3fi = c3 * alpha[-4] + a3e * alpha[-1];=$ 0.777472.

ii. $m = 4$ In this and the following subcase we use the same calculation:
$stab(G) = max(1 + stab(G - \overline{N}(A_{i'})), 1 + +stab'(G - A_{i'} - \overline{N}(A_i), (B^{i',1}, B^{i',2}), 1)))$
$c3fii = c4 * alpha[-4] + a1e * alpha[-4];=$ 0.799940.

iii. $m > 4$ $c3fiii = alpha[-4] + a1ex * alpha[-4];=$ 0.784657.

$c3f = max(3, c3fi, c3fii, c3fii);=$ 0.799940.

$c3 = max(6, c3a, c3b, c3c, c3d, c3e, c3f);=$ 0.837212 *** . Or if there are two non-adjacent vertices of degree 3
$c3x = max(6, c3a, c3bx, c3c, c3d, c3e, c3f);=$ 0.816748 *** .

4. **Degree** 4

For degrees 4 to 7, apart from a few special cases, we always use the same method. Choose a vertex of minimum degree $d$ and having the minimum possible number of neighbours of degree $d$. Call its neighbours $B_1$ to $B_d$ where the order is chosen in a way to be described below. We use refinements on the simple calculation:
$stab(G) = 1 + max(stab(G - \overline{N}(A)), max_{i=1}^{d-1}(stab'(G - \overline{N}(B_i) - \cup_{j=1}^{i-1} B_j), (\cup_{j=i+1}^{d} B_j, 1)))$.
A range of refinements may be used depending on the structure of the graph around $A$, more precisely on the subgraph $G_B$ induced on $\{B_1, \cdots B_d\}$, on the degrees of all $B_i$ and, in some critical cases, the numbers of common neighbours of pairs and triples of elements of $B$.

   (a) **The minimum degree in** $G - \overline{N}(A)$
   If $m > 1$ vertices $B_i$ have degree equal to $d$, then , by the choice of $A$, each of them has at least $m - 1$ neighbours different from $A$ and of the same degree $d$. If any of these $B_i$ does not have $m - 1$ neighbours in $G_B$ whose degree in $G$ was $d$, then $G - \overline{N}(A)$ has a vertex of degree at most $d - 1$.

   (b) **The minimum degree in** $G - \overline{N}(B_i) - \cup_{j=1}^{i-1} B_j$
   In $G - \overline{N}(B_i) - \cup_{j=1}^{i-1} B_j$ we have an upper bound on the degree of each $B_j$ $(j > i)$ excluding any neighbours of $B_i$. This is $d_G(B_j)$ minus one for $A$ and for each $B_k$ which is a neighbour of $B_j$ and either has $k < i$ or is also a neighbour of $B_i$.
   The $c_m$ $(6 \leq m \leq 8)$ which can be justified by this argument is only used if it is smaller than the $a_m$ suggested by the $stab'(\ \ , (\cup_{j=i+1}^{d} B_j, 1))$ option.

   (c) **Neighbours of previous** $B_j$
   For $i > 1$, if no maximum independent set is included in those considered previously, then any maximum independent set includes at least two neighbours of each $B_j$ for $j < i$. We have an upper bound on the number of such neighbours in the graph $G - \overline{N}(B_i) - \cup_{j=1}^{i-1} B_j$. If $B_i$ is not itself a neighbour of $B_j$ this gives the possibility of using a constant. Again we use the smallest such constant if it is smaller than one already justified.

The bound provable by the use of these techniques on each of the recursive calls may depend on the order of the $B_i$. For a given structure around $A$, we evaluate the bound for each of the $d!$ orders and the algorithm chooses one of the orders giving the lowest result.

The worst case bound on the performance is computed by carrying out this choice for each possible structure around $A$. The possible structures are precomputed and one example of each isomorphism class is inserted into a list and the function *evaluate*() calculates the constant for each element of the appropriate list and returns the maximum. In all cases we take into account only structures where the number of common neighbours of two elements of $B$ (excluding any in $\{A\} \cup B$) is at most 3 since those with four or more common neighbours are easily handled by $stab(G) = max(stab(G-B_1-B_2), stab'(G-N(B_1)\cap N(B_2), (\{B_1, B_2\}, 1)))$ if $B_1$ and $B_2$ are the vertices of $B$ with at least five common neighbours (including $A$). The number of distinct structures is greatest at 38125 for $d = 7$. The second parameter to $evaluate()$ is a default parameter $def$ such that a call of $stab(G')$ is guaranteed to take time at most $k * def * 2^{c|G|}$ for any $G'$ containing $G - \overline{N}(A)$.

(a) **A vertex $B$ in $N(A)$ has $d(B) \geq 8$**
$stab(G) = max(stab(G - B), 1 + stab'(G - \overline{N}(B), (N(A) - B, 1)))$
$c4a = alpha[-1] * c3 + alpha[-9] * a2;= 0.869495.$

(b) **Subdominance**
If $N(A) = \{B_1, B_2, B_3, B_4\}$ and there are edges $(B_1, B_2)$ and $(B_1, B_3)$, there is a m.i.s. containing either $B_4$ and one other $B_i$ or neither $B_1$ nor $B_4$.
$stab(G) = max(1+stab'(G-\overline{N}(B_4), (\{B_1, B_2, B_3\}, 1)), stab(G-B_1-B_4))$, where $d(A) = 2$ in the call of $stab$.
$c4b = alpha[-5] * a2 + alpha[-2] * c2;= 0.876901.$

(c) **Otherwise**
$c4c = evaluate(degree4, 1.0);= 0.898284$ *** .

$c4 = max(3, c4a, c4b, c4c);= 0.898284$ *** .
Where it is known that there are two non-connected vertices of degree 4 we improve this to
$c4x = max(3, c4a, c4b, evaluate(degree4, c4));= 0.889737$ *** .

5. Degree 5

(a) **A vertex $B$ in $N(A)$ has $d(B) \geq 8$**
$stab(G) = max(stab(G - B), 1 + stab(G - \overline{N}(B)))$
$c5a = alpha[-1] * c4 + alpha[-9];= 0.932396.$

(b) **Vertices $B$ and $B'$ in $N(A)$ have $d(B) \geq 7$ and $d_{G-\overline{N}(B)}(B') \leq 4$**
Similar to case $c6c$
$c5b = (alpha[-8] + alpha[-1]) * c4;= 0.947778$ *** .

(c) **Subdominance** (edges $(B_1, B_i)$ for $i = 2, 3, 4$)
$stab(G) = max(1 + stab(G - \overline{N}(B_5)), stab(G - B_1 - B_5))$, where $d(A) = 3$ in the second call of $stab$.
$c5c = alpha[-6] + alpha[-2] * c3;= 0.903542.$

(d) **Otherwise**
$c5d = evaluate(degree5, 1.0);= 0.932109.$

$c5 = max(4, c5a, c5b, c5c, c5d);= 0.947778$ *** .

6. Degree 6

(a) **A vertex $B$ in $N(A)$ has $d(B) \geq 8$**
$stab(G) = max(1 + stab(G - \overline{N}(B)), stab(G - B))$
$c6a = alpha[-9] + alpha[-1] * c5; = 0.973474.$

(b) **Subdominance** (edges $(B_1, B_i)$ for $i = 2, 3, 4, 5$; similar to case $c4b$)
$c6b = alpha[-7] + alpha[-2] * c4; = 0.890033.$

(c) **Vertices $B$ and $B'$ in $N(A)$ have $d(B) \geq 7$ and $d_{G-\overline{N}(B)}(B') \leq 5$**
This will happen if $N(A)$ contains two vertices of degree 6 and $\geq 7$ not adjacent or three vertices of degree 7 with exactly two edges between them. $stab(G) = max(1 + stab(G - \overline{N}(B)), stab(G - B))$
$c6c = (alpha[-8] + alpha[-1]) * c5; = 1.000000$ *** .

(d) **$N(A)$ has no edges**
All vertices in $N(A)$ have degree $\geq 7$.

i. The last two vertices in $N(A)$ have at least one common neighbour $v$ apart from $A$
Unless $N(B_6) \subset N(B_5)$ which allows $stab(G) = max(1 + stab(G - \overline{N}(B_6)), stab(G - B_5 - B_6))$, we have
$stab(G) = max(1 + stab(G - \overline{N}(A)), 1 + stab(G - \overline{N}(B_1)), 1 + stab(G - B_1 - \overline{N}(B_2)), 1 + stab'(G - B_1 - B_2 - \overline{N}(B_3), ((B_4, B_5, B_6), 1)), 1 + stab'(G - B_1 - B_2 - B_3 - v - \overline{N}(B_4), ((B_5, B_6), 1)), 2 + stab(G - B_1 - B_2 - B_3 - B_4 - \overline{N}(B_5, B_6)))$
$c6di = alpha[-7] + alpha[-8] + alpha[-9] + alpha[-10] * a2 + alpha[-11] * alpha[-1] * a1 + alpha[-14]; = 0.990458.$

ii. Otherwise
The same calculation except for the omission of $-v$ but now $|\overline{N}(B_5, B_6)| = 14$.
$c6dii = alpha[-7] + alpha[-8] + alpha[-9] + alpha[-10] * a2 + alpha[-11] * a1 + alpha[-18]; = 0.969953.$

$c6d = max(2, c6di, c6dii); = 0.990458.$

(e) **Otherwise**
$c6e = evaluate(degree6, 1.0); = 0.922053.$

$c6 = max(5, c6a, c6b, c6c, c6d, c6e); = 1.000000$ *** .

7. **Degree** 7: $c7 = evaluate(degree7, 1.0); = 0.980268$

8. **Degree** 8: $A$ has a neighbour $B$ with degree 9.
$stab(G) = max(1 + stab(G - \overline{N}(B)), stab(G - B))$
$c8 = alpha[-10] + alpha[-1]; = 0.985043$

9. **Unconnected**: if $G$ is not connected, we simply call $stab$ on each connected component and sum the results.

(a) if there is a connected component consisting of a single isolated vertex $v$, $stab(G) = 1 + stab(G - v)$
$c9a = alpha[-1]; = 0.829959.$

(b) otherwise time $\leq kmax_{i=2}^{n-2}(\alpha^i + \alpha^{n-i}) \leq \alpha^n(\alpha^{-2} + \alpha^{2-n})$, giving $c9b = alpha[-2] + alpha[-12]; = 0.795658$ since we can assume $n \geq 14$.

$c9 = max(2, c9a, c9b); = 0.829959$.

10. **Miscellaneous cases** If the graph is regular we can simply choose any vertex $v$ and use $stab(G) = max(1 + stab(G - \overline{N}(v)), stab(G - v))$. [3] explains why these cases can be ignored in the timing analysis.

    If any pair of vertices $v$ and $w$ are not connected but $|N(v) - N(w)| \leq 1$, there is a m.i.s. either containing both $v$ and $w$ or not containing $w$. Thus we can use $stab(G) = max(2 + stab(G - \overline{N}(v, w)), stab(G - w)$ if appropriate. If in addition $|N(w) - N(v)| \leq 1$, we have the better possibility $stab(G) = max(2 + stab(G - \overline{N}(v, w)), stab(G - v - w))$. In each case we can improve the $stab(G - v - w)$ to $stab'(G - v - w, (N(v) \cup N(w), 3))$.

# 3 Constrained

1. **One out of two** Suppose that the set $S$ is $\{A, B\}$ with $d(A) \leq d(B)$.

   (a) $d(A) = 0$: $stab'(G, (S, 1)) = 1 + stab(G - A)$ $a1a = alpha[-1]; = 0.829959$.

   (b) $d(A) = 1$: $stab'(G, (S, 1)) = 1 + stab(G - \overline{N}(A))$ $a1b = alpha[-2]; = 0.688832$.

   (c) $(A, B)$ **is an edge and** $d(A) = 2$
   Let $x$ be the other neighbour of $A$ and $G'$ be $G - A - B$ with added edges between $x$ and each remaining element of $N(B)$.
   $stab'(G, (S, 1)) = 1 + stab'''(G', x)$.
   $a1c = a8 * alpha[-2]; = 0.796842$.

   (d) $(A, B)$ **is an edge and** $d(A) = 3$
   We assume that $A$ and $B$ have no common neighbour (case $a1f$).

      i. $d(B) > 3$
      $stab'(G, ((A, B), 1)) = 1 + max(stab(G - \overline{N}(A)), stab'(G - \overline{N}(B), (N(A) - B, 1)))$
      $a1di = alpha[-4] + alpha[-5] * a1; = 0.801333$.

      ii. $d(B) = 3$ Of the four neighbours of $A$ and $B$, let $C$ be the one with minimum degree in $G - \overline{N}(B)$ or $G - \overline{N}(A)$. Assume wlog. that $C$ is in fact in $N(A)$ and that $d(C)$ in $G - \overline{N}(B)$ is $d$; let the other two neighbours of $B$ be $D$ and $E$.

         A. $d \leq 3$
         $stab'(G, ((A, B), 1) = 1 + max(stab(G - \overline{N}(A)), stab(G - \overline{N}(B)))$
         $a1diiA = alpha[-4] * a1 + alpha[-4] * max(4, c1, c2, c3, c9); = 0.791055$.

B. $d = 4$
$stab'(G, ((A, B), 1)) = max(1 + stab(G - \overline{N}(B)), 2 + stab(G - \overline{N}(A, D)), 2 + stab(G - \overline{N}(A, E))$
$a1diiB = alpha[-4]*c4+alpha[-4]*2*alpha[-5]; = 0.799940.$

C. $d > 4$
We use the same calculation as in $a1diiB$. We no longer have the $c4$ from the low degree of $C$ but this is compensated by the fact that $|\overline{N}(A, D)| \geq 10$ and similarly for $E$.
$a1diiC = alpha[-4] + alpha[-4] * 2 * alpha[-6]; = 0.784657.$

$a1dii = max(3, a1diiA, a1diiB, a1diiC); = 0.799940.$

$a1d = max(2, a1di, a1dii); = 0.801333.$

(e) $(A, B)$ **is an edge and** $d(A) > 3$
$stab'(G, (S, 1)) = max(stab(G - \overline{N}(A)), stab(G - \overline{N}(B))) + 1.$
$a1e = 2 * alpha[-5]; = 0.787613.$
or if in fact $d(A) > 4$
$a1ex = 2 * alpha[-6]; = 0.653687.$

(f) $A$ **and** $B$ **have a common neighbour** $v$
$stab'(G, (S, 1)) = stab'(G - v, (S, 1)).$
$a1f = a1 * alpha[-1]; = 0.688832.$

(g) $d(B) = 2$ **(and so also** $d(A) = 2$**)**
Let the neighbours of $A$ be $C$ and $D$. The independent sets to be considered contain either $A$ or $B$, $C$ and $D$.

i. $(C, D)$ is an edge
$stab'(G, (S, 1)) = 1 + stab(G - \overline{N}(A))$. $a1gi = alpha[-3]; = 0.571702.$

ii. **All neighbours of** $C$ **and** $D$ **apart from** $A$ **are also neighbours of** $B$
Any independent set containing $A$ can be transformed into one at least as large by removing $A$ and any neighbours of $B$ and adding $\{B, C, D\}$. $stab'(G, (S, 1)) = 3 + stab(G - \overline{N}(\{A, B\}))$
$a1gii = alpha[-6]; = 0.326843.$

iii. **Otherwise**
$stab'(G, (S, 1)) = max(1 + stab(G - \overline{N}(A)), 3 + stab(G - \overline{N}(\{B, C, D\})))$
where, in the first call the graph has a vertex $(B)$ of degree 2.
$a1giii = c2 * alpha[-3] + alpha[-7]; = 0.693578.$

$a1g = max(3, a1gi, a1gii, a1giii); = 0.693578.$

(h) Otherwise
$stab'(G, (S, 1)) = max(stab(G - \overline{N}(B)), stab'(G - \overline{N}(A) - B, (N(B), 2))) + 1.$

i. $d(A), d(B) = 2, 3$: $a1hi = c2 * alpha[-4] + a3 * alpha[-4]; = 0.677344.$

ii. $d(A), d(B) = 2, 4$: $a1hii = c2 * alpha[-5] + a4 * alpha[-4]; = 0.658968.$

iii. $d(A), d(B) = 2, 5$: $a1hiii = c2 * alpha[-6] + a5 * alpha[-4];= 0.667662$.

iv. $d(A), d(B) = 2, > 5$: $a1hiv = c2 * alpha[-7] + alpha[-4];= 0.674871$.

v. $d(A), d(B) = 3, 3$: $a1hv = c3 * alpha[-4] + a3 * alpha[-5];= 0.668515$.

vi. $d(A), d(B) = 3, 4$: $a1hvi = c3 * alpha[-5] + a4 * alpha[-5];= 0.635180$.

vii. $d(A), d(B) = 3, > 4$: $a1hvii = c3 * alpha[-6] + alpha[-5];= 0.667444$.

viii. $d(A), d(B) = 4, 4$: $a1hviii = c4 * alpha[-5] + a4 * alpha[-6];= 0.607286$.

ix. $d(A), d(B) \geq 4, > 4$: $a1hix = alpha[-6] + alpha[-6];= 0.653687$.

$a1h = max(9, a1hi, a1hii, a1hiii, a1hiv, a1hv, a1hvi, a1hvii, a1hviii, a1hix);= 0.677344$.

$a1 = max(8, a1a, a1b, a1c, a1d, a1e, a1f, a1g, a1h);= 0.829959$.
or when we know $d(A) > 0$
$a1x = max(7, a1b, a1c, a1d, a1e, a1f, a1g, a1h);= 0.801333$.

2. **One out of three**

(a) **some element of $S$ has degree $4$ or less** $stab'(G, (S, 1)) = stab(G)$
$a2a = max(5, c9, c1b, c2, c3, c4);= 0.898284$.

(b) **all elements of $S$ have degree at least $5$**

i. **Some elements $s_1$ and $s_2$ are not adjacent**
$stab'(G, (S, 1)) = 1 + max(stab(G - \overline{N}(s_1)), stab(G - \overline{N}(G - s_2) - s_1), stab(G - \overline{N}(s_3)))$
$a2b1 = 2 * alpha[-6] + alpha[-7];= 0.924953$.

ii. **some element $s_1$ of $S$ has degree $6$ or more**
$stab'(G, (S, 1)) = max_i(1 + stab(G - \overline{N}(s - i)))$
$a2b2 = alpha[-7] + 2 * alpha[-6];= 0.924953$.

iii. **two elements $s_1$ and $s_2$ of $S$ have a common neighbour other than $s_3$**
$stab'(G, (S, 1)) = 1 + max(stab(G - \overline{N}(s_1)), stab(G - \overline{N}(s_3)), stab'(G - \overline{N}(s_2), (N(s_1) - s_2 - s_3 - N(s_1) \cap N(s_2))))$
$a2b3 = alpha[-6] + alpha[-6] * a2 + alpha[-6] * a1;= 0.903590$.

iv. **otherwise** $stab'(G, (S, 1)) = 1 + max(stab(G - \overline{N}(s_1)), stab'(G - \overline{N}(s_2), (N(s_1) - s_2 - s_3, 1)), stab'(G - \overline{N}(s_3), (N(s_1) - s_2 - s_3, 1), (N(s_2) - s_1 - s_3, 1)))$
$a2b4 = alpha[-6] + alpha[-6] * (a2 + a2x);= 0.934638$ *** .
Or, if there are two "one out of three" conditions with disjoint sets $S$,
$a2b4x = alpha[-6] * a2 + alpha[-6] * (a2 + a2x);= 0.913275$.

$a2b = max(4, a2b1, a2b2, a2b3, a2b4);= 0.934638$ *** .
Or, if there are two "one out of three" conditions with disjoint sets $S$,
$a2bx = max(4, a2b1, a2b2, a2b3, a2b4x);= 0.924953$.

$a2 = max(2, a2a, a2b);= 0.934638$ *** .
Or, if there are two "one out of three" conditions with disjoint sets $S$,
$a2x = max(2, a2a, a2bx);= 0.924953$.

3. **Two out of three**

   (a) There is a vertex $A \in S$ of degree 0:
   $stab'(G,(S,2)) = 1 + stab'(G - A,(S - A, 1))$ $a3a = a1 * alpha[-1];=$ $0.688832$.
   or if there are two "two out of three" conditions with disjoint sets $S$
   $a3ax = a3 * alpha[-1];= 0.571702$.

   (b) There is a vertex $A \in V(G) - S$ of degree 1: $stab'(G,(S,2)) = stab'(G-\overline{N}(A),(S-N(A),2))$ $a3b = max(2, alpha[-2], a3)*alpha[-2];=$ $0.474489$.

   (c) $S$ is a 3-clique: $stab'(G,(S,2)) = 0$

   (d) $S$ has two edges $(A,B)$ and $(A,C)$ (but excluding case $g$): $stab'(G,(S,2)) = 2 + stab(G - \overline{N}(B) - \overline{N}(C))$ $a3d = alpha[-4];= 0.474489$.

   (e) $S$ has one edge $(A,B)$: $stab'(G,(S,2)) = 1 + stab'(G-\overline{N}(C),(\{A,B\},1))$
   $a3e = max(4, a1b, a1c, a1d, a1e) * alpha[-3];= 0.458124$.

   (f) $S$ has two vertices $A$ and $B$ with a common neighbour $v$: $stab'(G,(S,2)) = stab'(G-v,(S,2))$. $a3f = max(6, a3b, a3d, a3e, a3f, a3g, a3h)*alpha[-1];=$ $0.474489$.

   (g) $S$ has a vertex $A$ of degree 1: $stab'(G,(S,2)) = 1 + stab'(G - \overline{N}(A),(S - A, 1))$. $a3g = a1 * alpha[-2];= 0.571702$.

   (h) None of the above: Let $A$ be an element of $S$ with minimum degree. $stab'(G,(S,2)) = max(1 + stab'(G - \overline{N}(A),(S - A, 1)), 2 + stab'(G - A - \overline{N}(S - A),(N(A), 2)))$, where the first call has $S - A$ a set of two vertices each of degree at least $d(A)$ and with no edge between them or a common neighbour.

      i. $d(A) = 2$:
      (Better calculation of $stab'$) $stab'(G,(S,2)) = max(1 + stab'''(G-\overline{N}(A), S - A), 2 + stab(G - \overline{N}(S - A)))$.
      Many of the worst cases of $stab'''$ can only occur when the size of $\overline{N}(S - A)$ is large:
      $a3hi = max(14, /*a7d\, cannot\, occur\, because\, no\, common\, neighbours* /a7e*alpha[-3]+c2*alpha[-6], /*switching\, on\, C\, instead\, of\, A* /a7e*alpha[-4]+c3*alpha[-6], a7g*alpha[-3]+c2*alpha[-8], a7h* alpha[-3]+c2*alpha[-9], a7i*alpha[-3]+c2*alpha[-10], a7j* alpha[-3]+c2*alpha[-11], a7k*alpha[-3]+c2*alpha[-8], a7l*$

$alpha[-3]+c2*alpha[-9], a7m*alpha[-3]+c2*alpha[-10], a7n*alpha[-3]+c2*alpha[-10], a7o*alpha[-3]+c2*alpha[-11], a7p*alpha[-3]+c2*alpha[-12], a7q*alpha[-3]+c2*alpha[-12], a7r*alpha[-3]+c2*alpha[-13]) ;= 0.505447.$

If we know only that $A$ has no common neighbours with $B$ and $C$ the other elements of $S$ (but not that $B$ and $C$ have no common neighbours) and that the minimum $m$ of $d(B)$, $d(C)$ is greater than 2, we can do better. Either $B$ and $C$ have a common neighbour $v$ giving $stab'(G,(S,2)) = stab'(G-v,(S,2))$ where $d(A) = 2$ in $G-v$ and only cases $a3h$ and $a3f$ can occur or $stab'(G,(S,2)) = max(1+stab'''(G-\overline{N}(A), \{B,C\}, 2+stab(G-\overline{N}(\{B,C\})))$ and, depending on the subcase of $stab'''$ which occurs we have lower bounds on $|\overline{N}(\{B,C\})|$ and $d(A) = 2$ in $G-\overline{N}(\{B,C\})$

If $m = 3$, $a3hix = max(2, max(2, a3h, a3f)*alpha[-1], max(3, a7k*alpha[-3]+c2*alpha[-8], a7l*alpha[-3]+c2*alpha[-9], a7m*alpha[-3]+c2*alpha[-10]))$;$= 0.465778,$

If $m \geq 4$, $a3hiy = max(2, a3hix*alpha[-1], max(5, a7n*alpha[-3]+c2*alpha[-10], a7o*alpha[-3]+c2*alpha[-11], a7p*alpha[-3]+c2*alpha[-12], a7q*alpha[-3]+c2*alpha[-12], a7r*alpha[-3]+c2*alpha[-13]))$;$= 0.404453.$

ii. $d(A) = 3$: We use the same method as in the no-common-neighbour part of cases $a3hix$ and $a3hiy$:
$a3hii = max(3, c3*alpha[-8]+max(6, a7k, a7n, a7o, a7p, a7q, a7r)*alpha[-4], c3*alpha[-9]+a7l*alpha[-4], c3*alpha[-10]+a7m*alpha[-4])$;$= 0.437037.$

iii. $d(A) \geq 4$:
$stab'(G,((A,B,C),2)) = max(1+stab'(G-\overline{N}(A),((B,C),1)), 2+stab(G-A-\overline{N}(B,C)))$
$a3hiii = a1h*alpha[-5]+alpha[-11]$;$= 0.395456.$

$a3h = max(3, a3hi, a3hii, a3hiii)$;$= 0.505447.$

(i) **Degrees of all vertices in $S$ are at least 3**

i. **Two elements $A$ and $B$ of $S$ are neighbours**
$stab'(G,(\{A,B,C\},2)) = 1+stab'(G-\overline{N}(C),(\{A,B\},1))$
$a3ii = a1*alpha[-4]$;$= 0.393807.$

ii. **Two elements of $S$ have a common neighbour $v$**
$stab'(G,(\{A,B,C\},2)) = 1+stab'(G-v,(\{A,B,C\},2))$
$a3iii = max(2, a3f, a3h)*alpha[-1]$;$= 0.419500.$

Otherwise we must be in case $3h$.
$a3i = max(4, a3hii, a3hiii, a3ii, a3iii)$;$= 0.437037.$

$a3 = max(8, a3a, a3b, a3d, a3e, a3f, a3g, a3h, a3i)$;$= 0.688832.$
or, when we know that $S$ cannot contain two isolated vertices,
$a3x = max(8, alpha[-1]*a1x, a3b, a3d, a3e, a3f, a3g, a3h, a3i)$;$= 0.665073.$

or if there are two "two out of three" conditions with disjoint sets $S$
$a3y = max(8, a3ax, a3b, a3d, a3e, a3f, a3g, a3h, a3i);= 0.571702$.

4. **Two out of four** We use the same *evaluate* function as in the previous section. Now the structures do not have the central node $A$ but otherwise the computation is similar except that now the $default$ parameter is applicable for any graph containing $G - \overline{N}(S)$. Again in *evaluate* we ignore cases where two vertices of $S$ say $s_1$ and $s_2$ have more than 3 common neighbours since these can be handled by $stab'(G,(S,2)) = max(2 + stab(G - s_1 - s_2 - \overline{N}(s_3, s_4)), stab'(G - \overline{N}(s_1) \cap \overline{N}(s_2), (\{s_1, s_2\}, 1)))$ except for one case where there are edges $(s_1, s_3)$, $(s_1, s_4)$, $(s_2, s_3)$ and $(s_2, s_4)$ in which case $stab'(G,(S,2)) = 2 + max(stab(G - \overline{N}(s_1, s_2)), stab(G - \overline{N}(s_3, s_4)))$ suffices.

   (a) $G$ **has a vertex** $v$ **of degree** $0$
   $stab'(G,(S,2)) = stab'(G - v, (S - v, 2 - |S \cap \{v\}|))$
   $a4a = max(2, a2, a4) * alpha[-1];= 0.775711$ *** .
   or if there are two "two out of four" conditions with disjoint sets $S$
   $a4ax = a4 * alpha[-1];= 0.643809$.

   (b) $S$ **has a vertex** $v$ **of degree** $\leq 2$
   $stab'(G,(S,2)) = stab(G)$
   $a4b = max(5, c1b, c2b, a4bi, a4bii, a4biii);= 0.699629$ *** .
   The worse cases of $c2$ are dealt with by considering the number $w$ of elements of $N(v) \cap S$. The call $stab(G - \overline{N}(\overline{N}(v)))$ has the condition $(S - v, 2 - w)$ and the call $stab(G - \overline{N}(v))$ has a condition $(S - N(v) \cap S, 1)$ where this is stronger than the simple $(\overline{N}(\overline{N}(v)) - \overline{N}(v), 2)$. If $S - N(v) \cap S, 1)$ was a single vertex $v'$ with $d_{G-\overline{N}(v)}(v') = 0$, we would have used instead the much stronger $stab'(G,(S,2)) = 2 + stab(G - \overline{N}(\{v, v'\}))$ so any call $stab'(G - \overline{N}(v), (S - N(v) \cap S, 1))$ where $S - N(v) \cap S, 1) = \{v'\}$ is replaced by $1 + stab'(G - \overline{N}(\{v, v'\}), \overline{N}(\overline{N}(v)) - v', 2 - |\overline{N}(\overline{N}(v)) \cap \{v'\}|)$ with the time bound for this call given by $\alpha^{-5}$ multiplied by the appropraite constant for the $stab'$ call. This gives the following bounds where $x$ as in $c2$ is $|\overline{N}(\overline{N}(v)) - \overline{N}(v)|$, the maximum of three is over $w = 0, 1, 2$ in that order and the first summand is always that for $G - \overline{N}(v)$.

      i. $x = 3$
      $a4bi = max(3, a3 * alpha[-3] + a3 * alpha[-6], a3 * alpha[-3] + a1 * alpha[-6], max(2, a1, a3) * alpha[-5] + alpha[-6]);= 0.665073$.

      ii. $x = 4$
      $a4bii = max(3, a4 * alpha[-3] + a3 * alpha[-7], a1 * alpha[-3] + a1 * alpha[-7], alpha[-5] + alpha[-7]);= 0.699629$.

      iii. $x \geq 5$
      $a4biii = max(3, a2 * alpha[-3] + a3 * alpha[-8], a1 * alpha[-3] +$

$a1 * alpha[-8], alpha[-5] + alpha[-8]);= 0.689418.$

(c) **All vertices of $S$ have degree $\geq 3$**
$a4c = evaluate(twofrom4, 1.0);= 0.711739.$
Or for $stab'(G, (S_1, 2), (S_2, 2))$ with $|S_1| = |S_2| = 4$ and $S_1$ and $S_2$ disjoint, the condition $(S_2, 2)$ or a stronger one is still valid in any graph containing $G - \overline{N}(S_1)$
$a4cx = evaluate(twofrom4, a4);= 0.692268.$

$a4 = max(3, a4a, a4b, a4c);= 0.775711$ *** .
or if there are two "two out of four" conditions with disjoint sets $S$
$a4x = max(3, a4ax, a4b, a4cx);= 0.699629$ *** or similarly if there are a "two out of four" and a "two out of five" condition with disjoint sets $S$
$a4y = max(3, a5 * alpha[-1], a4b, a4c);= 0.745539$

5. **Two out of five**

   (a) **$G$ has a vertex of degree $\leq 4$**
   $stab'(G, (S, 2)) = stab(G, S)$
   $a5a = max(5, c9a, c1b, c2, c3, c4);= 0.898284.$

   (b) **All degrees $\geq 5$ and $d(s_1) > 7$**
   $stab'(G, (S, 2)) = max(1 + stab'(G - \overline{N}(s_1), (S - s_1, 1)), stab'(G - s_1, (S - s_1, 2)))$ $a5b = alpha[-9] + a4c * alpha[-1];= 0.777572.$

   (c) **All degrees $\geq 5$ but $\leq 6$**
   $a5c = evaluate(twofrom5, 1.0);= 0.729272.$

   $a5 = max(3, a5a, a5b, a5c);= 0.898284.$

6. **Two out of six**

   (a) **$G$ has a vertex of degree $\leq 5$**
   $stab'(G, (S, 2)) = stab(G, S)$
   $a6a = max(6, c9a, c1b, c2, c3, c4, c5);= 0.947778.$

   (b) **$S$ has a vertex $s_1$ adjacent to all other vertices of $S$**
   $s_1$ cannot be in one of the relevant independent sets
   $stab'(G, (S, 2)) = stab'(G - s_1, (S - s_1, 2))$
   $a6b = a5b * alpha[-1];= 0.645352.$

   (c) **Else**
   $stab'(G, (S, 2)) = max(1 + stab'(G - \overline{N}(s_1), (S - s_1, 1)), stab'(G - s_1, (S - s_1, 2)))$
   $a6c = alpha[-7] + max(2, a5b, a5c) * alpha[-1];= 0.916619.$

   $a6 = max(3, a6a, a6b, a6c);= 0.947778.$

7. **Exactly one out of two**
   We need consider only independent sets containing $x$ and two neighbours of $y$ or vice versa. We switch on the degrees $d(x)$ and $d(y)$, assuming w.l.o.g. that $d(x) \leq d(y)$. Except where otherwise noted we

use the simple calculation $stab'''(G,\{x,y\}) = 1 + max(stab'(G - \overline{N}(x) - y,(N(y),2)), stab'(G - \overline{N}(y) - x,(N(x),2))$.

(a) **both degrees** $< 2$
$stab''(G,x,y) = 0$.

(b) $d(x) = 0$
$stab''(G,x,y) = 1 + stab(G - \overline{N}(x) - y)$
$a7b = alpha[-2];= 0.688832$.

(c) $d(x) = 1$
$stab''(G,x,y) = 1 + stab(G - \overline{N}(x) - y)$
$a7c = alpha[-3];= 0.571702$.

(d) $x$ **and** $y$ **have a common neighbour** $v$
$stab''(G,x,y) = stab''(G - v,x,y)$
$a7d = max(16, a7c, a7d, a7e, a7f, a7g, a7h, a7i, a7j, a7k, a7l, a7m, a7n, a7o, a7p, a7q, a7r)*$
$alpha[-1];= 0.500633$.

(e) $d(x) = 2,\ d(y) = 2$
W.l.o.g. assume that $|\overline{N}(\overline{N}(x)) - \overline{N}(\{x,y\})| = b \leq |\overline{N}(\overline{N}(y)) - \overline{N}(\{x,y\})|$.

i. $b = 0$
$x + \overline{N}(y)$ dominates $y + \overline{N}(x)$.
$stab''(G,x,y) = 3 + stab(G - \overline{N}(\{x,y\}))$.
$a7ei = alpha[-6];= 0.326843$.

ii. $b = 1$
The independent sets consist of 3 elements of $\overline{N}(\{x,y\})$ together with an independent set of $G - \overline{N}(\{x,y\})$ which may contain $\overline{N}(\overline{N}(x)) - \overline{N}(\{x,y\})$ or part of $\overline{N}(\overline{N}(y)) - \overline{N}(\{x,y\})$ but not both. Let $z$ be the single element of $\overline{N}(\overline{N}(x)) - \overline{N}(\{x,y\})$ and $G'$ be the graph obtained by adding to $G - \overline{N}(\{x,y\})$ edges between $z$ and every element of $\overline{N}(\overline{N}(y)) - \overline{N}(\{x,y\})$.
$stab''(G,x,y) = 3 + stab'''(G',z)$,
$a7eii = a8 * alpha[-6];= 0.378093$.

iii. $b \geq 2$
$stab''(G,x,y) = 3 + max(stab(G - \overline{N}(x) - \overline{N}(\overline{N}(y))),$
$stab(G - \overline{N}(y) - \overline{N}(\overline{N}(x))))$
$a7eiii = 2 * alpha[-8];= 0.450280$.

$a7e = max(3, a7ei, a7eii, a7eiii);= 0.450280$

(f) $d(x) = 2,\ d(y) = 3$
$a7f = a3 * alpha[-4] + alpha[-7];= 0.598110$.

(g) $d(x) = 2,\ d(y) = 4$
$a7g = a4 * alpha[-4] + alpha[-8];= 0.593207$.

(h) $d(x) = 2,\ d(y) = 5$

i. $\overline{N}(\overline{N}(x)) - \overline{N}(x) \subset N(y)$
An independent set containing $x$ and two neighbours of $y$ is no

better than one containing $y$ and the two neighbours of $x$ so we consider only those containing three neighbours of $y$ and therefore two out of an arbitrary four of these neighbours.
$a7hi = a4 * alpha[-4] + alpha[-9]; = 0.554924.$

ii. **Otherwise** $a7hii = a5 * alpha[-5] + alpha[-9]; = 0.540607.$

$a7h = max(2, a7hi, a7hii); = 0.554924.$

(i) $d(x) = 2,\ d(y) = 6$
As in the previous case
$a7i = max(2, a5 * alpha[-4], alpha[-5]) + alpha[-10]; = 0.581310.$

(j) $d(x) = 2,\ d(y) > 6$
$a7j = alpha[-4] + alpha[-11]; = 0.603202.$

(k) $d(x) = 3,\ d(y) = 3$: We consider the neighbours of $x$, say $a,\ b,\ c$ and those of $y$, say $d,\ e,\ f$ and their degrees in the induced subgraphs $G - \overline{N}(y) - x$ and $G - \overline{N}(x) - y$ respectively.

i. **Two of** $a,\ b,\ c$ **or two of** $d,\ e,\ f$ **have degree** $0$ **in the subgraphs**
Without loss of generality suppose that the vertices concerned are $a,\ b$. We need consider only independent sets containing either $x,\ d,\ e, f$ or $y$ and two from $a,\ b,\ c$.
$stab''(G, \{x, y\}) = max(4 + stab(G - \overline{N}(x) - \overline{N}(\overline{N}(y))), stab'(G - \overline{N}(y) - x, (N(x), 2))).$
$a7ki = alpha[-8] + a3 * alpha[-5]; = 0.496407.$

ii. **else**
$a7kii = 2 * a3x * alpha[-5]; = 0.523820.$

$a7k = max(2, a7ki, a7kii); = 0.523820.$

(l) $d(x) = 3,\ d(y) = 4$
$a7l = a4 * alpha[-5] + a3 * alpha[-6]; = 0.530620.$

(m) $d(x) = 3,\ d(y) \geq 5$
$a7m = alpha[-5] + a3 * alpha[-7]; = 0.580664.$

(n) $d(x) = 4,\ d(y) = 4$
$a7n = 2 * a4 * alpha[-6]; = 0.507072.$

(o) $d(x) = 4,\ d(y) = 5$
$a7o = a5 * alpha[-6] + a4 * alpha[-7]; = 0.504023.$

(p) $d(x) = 4,\ d(y) > 5$
$a7p = alpha[-6] + a4 * alpha[-8]; = 0.501487.$

(q) $d(x) = 5,\ d(y) = 5$
$a7q = 2 * a5 * alpha[-7]; = 0.487349.$

(r) $d(x) > 4,\ d(y) > 5$
$a7r = alpha[-7] + alpha[-8]; = 0.496407.$

$a7 = max(17, a7b, a7c, a7d, a7e, a7f, a7g, a7h, a7i, a7j, a7k, a7l, a7m, a7n, a7o, a7p, a7q, a7r);$
$= 0.688832.$

8. **A vertex with added edges**
   There is a m.i.s. containing either $x$ or two neighbours of $x$.

   (a) $d(x) \leq 1$
   $stab'''(G,x) = 1 + stab(G - \overline{N}(x))$
   $a8a = alpha[-1];= 0.829959.$

   (b) $d(x) = 2$
   $stab'''(G,x) = max(1 + stab(G - \overline{N}(x)), 2 + stab(G - \overline{N}(\overline{N}(x))))$
   $a8b = 2 * alpha[-3];= 1.143404.$

   (c) $d(x) = 3$
   $stab'''(G,x) = max(1 + stab(G - \overline{N}(x)), stab'(G - x, (N(x), 2))$
   $a8c = alpha[-4] + a3 * alpha[-1];= 1.046191.$

   (d) $d(x) = 4$
   $stab'''(G,x) = max(1 + stab(G - \overline{N}(x)), stab'(G - x, (N(x), 2)))$
   $a8d = alpha[-5] + a4 * alpha[-1];= 1.037615.$

   (e) $d(x) > 4$
   $stab'''(G,x) = max(1 + stab(G - \overline{N}(x)), stab(G - x))$
   $a8e = alpha[-6] + alpha[-1];= 1.156802.$

   $a8 = max(5, a8a, a8b, a8c, a8d, a8e);= 1.156802.$

# 4 Memorisation

A useful gain in time is obtained at the cost of using exponential space by remembering the result found for any induced subgraph already analysed and avoiding doing the same calculation several times. This gain is evaluated by observing that the number of calls of the algorithm on graphs of size greater than $m$ is $O(2^{c(n-m)})$ so that if $m$ is chosen so that the number of induced subgraphs of size $\leq m$ is also $O(2^{c(n-m)})$ then so is the overall time.

## 4.1 Connected induced subgraphs

In fact it suffices to consider connected induced subgraphs and to consider graphs with degree bounded by 9 since the performance bounds we will obtain are easily proved recursively for all graphs if they hold for degree $\leq 9$ graphs. A bound on the number of small connected induced subgraphs is then obtained by considering the number of 8-ary trees of size $m$ (where a 8-ary tree is either empty or has a root node and 8 subtrees; note that the order of the subtrees is important). Relative to a given root node and an ordering of the edges at each vertex, the tree is interpreted as a spanning tree of the subgraph.

This bound overestimates the number of subgraphs for three reasons:

- Where the tree indicates that an edge is to be followed, the graph may have no edge; this is not significant since we want an upper bound on the number of connected subgraphs and adding more edges can only increase the number.

- An edge may lead to a vertex which has already been visited.
- A connected subgraph may have many spanning trees.

For the second of these three phenomena we have only an estimate of its effect on the performance of the algorithm. For the third we can show a definite improvement. We rely on the fact that the time taken by the algorithm is bounded by a polynomial times the number of graphs considered where two or more recursive calls are made. Thus, we need only count subgraphs with minimum degree at least 2 and no occurrences of dominance of the types mentioned in Section 1. In the following section we will assume that subgraphs are of this type.

### 4.2 Weighted Spanning Trees

Suppose a spanning tree of a subgraph has $l$ leaves. Since each of these vertices has degree at least 2 *qua* subgraph vertex, we can obtain other spanning trees (also with at most $l$ leaves) by attaching any leaf $v$ to another neighbour and breaking the resulting cycle just below the first ancestor of $v$ (in the original spanning tree) which is either of degree greater than 2 in the spanning tree or is its root. If we start with a spanning tree which minimises the number of leaves, no two leaves can be adjacent in the subgraph and this process of obtaining new spanning trees can be done without interference for any subset of the leaves and results in a different spanning tree in each case.

Hence if each 8-ary tree is given a weight equal to $2^{-l}$ where $l$ is its number of leaves, the sum of the weights for all spanning trees of a subgraph with minimum number $m$ of leaves over all its spanning trees, is at least $2^m 2^{-m}$ i.e. at least 1.

The number of 8-ary trees with size $m$ and $m_i$ nodes of outdegree $i$ ($0 \leq i \leq 8$) is $\left(\begin{array}{c} m \\ m_0, m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8 \end{array}\right) \prod_{i=0}^{8} \left(\begin{array}{c} 8 \\ i \end{array}\right)^{m_i} / m$ provided the $m_i$ satisfy $\sum_{i=0}^{8} m_i = m$ and $\sum_{i=0}^{8} i m_i = m - 1$. If we fix the proportion of tree nodes with out-degree $i$ to $x_i + O(1/m)$ and let $m$ tend to infinity, the sum of the weights of trees of size $m$ is to within a polynomial factor $(2^{-x_0} \prod_{i=0}^{8} (\left(\begin{array}{c} 8 \\ i \end{array}\right) / x_i)^{x_i})^m$. A simple calculation shows that subject to the constraints $\sum_{i=0}^{8} x_i = 1$ and $\sum_{i=0}^{8} i x_i = 1$ the constant is maximum at about 13.536 when the $x_i$ are about
(0.26339, 0.51713, 0.18024, 0.03490, 0.00405, 0.00028, 0.00001, 0.00000). Hence summed over all possible $m_i$ the sum is bounded by $13.536^m \times poly(m)$. This can be compared with the unweighted sum (the number of trees) which is about $17.65138^m$.

This raises the level at which the memorisation becomes effective (number of subgraphs possible is less than the number of recursive calls generated) to about $0.066n$ thereby reducing the constant $c$ to just under 0.251.

For graphs of degree greater than 9, we simply choose any vertex $A$ of degree greater than 9 and call $stab(G) = max(stab(G - A), 1 + stab(G - \overline{N}(A)))$.

Since this gives recursive calls on graphs of sizes $n - 1$ and $\leq n - 11$ and $\alpha'^{-1} + \alpha'^{-11} < 1$, this gives run time of $O(\alpha'^n)$ for all graphs.

## References


[1] Chen, J. Kanj, I. A. and Jia, W., Vertex cover: further observation and further improvement, *25th International workshop on graph-theoretic concepts in computer science*, Springer LNCS.

[2] Jian, T., An $O(2^{0.304n})$ algorithm for solving maximum independent set problem, *IEEE Trans. Comput.* **C-35** (1986), 847-851.

[3] Robson, J. M., Algorithms for maximum independent sets, *J. Algorithms* **7** (1986), 425-440.

[4] Tarjan, R. E. and Trojanowski, A. E., Finding a maximum independent set, *SIAM J. Comput.* **6** (1977), 537-546.